Lingbing Tang[1,2],
Pin Peng[3],
Changqing Luo[1, 2]

1 – Mobile E-business Collaborative Innovation Center of Hunan Province, Hunan University of Commerce, Changsha, Hunan, China
2 – Key Laboratory of Hunan Province for Mobile Business Intelligence, Changsha, Hunan, China
3 – School of Finance and Banking, Hunan University of Commerce, Changsha, Hunan, China

FINANCIAL STATEMENT FRAUD DETECTION THROUGH MULTIPLE INSTANCE LEARNING

Лінбін Тан[1,2] ,
Пінь Пен[3],
Чанцин Ло[1,2]

1 – Центр колаборативних інновацій у сфері мобільного мережевого бізнесу провінції Хунань, Хунаньський університет комерції, м. Чанша, провінція Хунань, Китай
2 – Провідна Лабораторія в галузі мобільної бізнес-інформаї та аналітики провінції Хунань, м. Чанша, провінція Хунань, Китай
3 – Факультет фінансів та банківської справи Хунаньського університету комерції, м. Чанша, провінція Хунань, Китай

ВИЯВЛЕННЯ ФАЛЬСИФІКАЦІЙ ФІНАНСОВОЇ ЗВІТНОСТІ ЧЕРЕЗ БАГАТОВАРІАНТНЕ НАВЧАННЯ

Purpose. Financial statement fraud detection (FSFD) based on machine learning is a very important problem for avoiding financial risk and maintaining an orderly market. The purpose of this research was to develop a multiple instance learning model that is capable of detecting and predicting the risk of fraudulent financial reporting.

Methodology. Each pair was composed of a singe-instance learning algorithm and its corresponding multiple instance learning algorithm, which were trained using a data set of 484 fraud companies as well as 902 normal companies with forming 4158 instances from Item 8 of the U.S. Securities and Exchange Commission (SEC) Form 10-K.

Findings. Empirical study shows that MIBoost, miGraph and CKNN are superior compared to AdaBoostM1, SVM and KNN correspondingly in accuracy, F1 score and area under receiver operating characteristics curve (AUC), which prove that multiple instance learning algorithms can fit FSFD better, especially under class-imbalance and few training data.

Originality. When a detecting label which corresponds to temporally local Financial Statement is attached collectively to groups of Financial Statements for one company without presenting the data to which Financial Statement this label is assigned, it is a multiple instance problem. The research presents a multiple instance learning model for FSFD originally.

Practical value. We have also considered the fact that some auditors are dissatisfied with the single label learning algorithms because there are many instances in one company without label. Our model is more reasonable and accurate.

Keywords: *financial statement, fraud detection, machine learning, multiple instance learning miboosting, miGraph CKNN*

Introduction. Increasing accounting fraud among public companies in the past decade has focused public attention on the corporate financial reporting process. To maintain public confidence in the reliability of financial reporting as a means to assess a company's future prospects, the SEC issued eleven financial reporting releases and ten staff accounting bulletins during the same time period. Among them, Financial Statements about U.S. public companies for the past three years are published in Item 8 of SEC Form 10-K. Though the SEC neither writes the 10-K nor vouches for its accuracy, the SEC sets the disclosure requirements and the SEC staff reviews 10-K to monitor and enhance companies' compliance with the requirements. In addition, laws and regulations prohibit companies from making materially false or misleading statements in 10-K. Likewise, companies are prohibited from omitting material information that is needed to make the disclosure not misleading. Obviously, the 10-K is so normal and authoritative that it can be used in detecting accounting fraud.

However, there is a significant challenge. That is, for each company, the single label with fraud or non-fraud gained by 10-k is attached to groups of Financial Statements in several years without making clear to which Financial Statement this label is assigned. It means that though the accounting fraud behaviour is temporally related and presents financial statements of a company within a certain period time, the identity and precise time of the label is remaining a mystery. Distinctly, the label is assigned to a group of financial statements, but this does not mean that the label applies to every financial statement in the group. The problem is that we only know whether a company is fraud or not while not knowing which financial statement responses to the category label, which is similar to a drug activity prediction. The main difficulty of that problem is that each molecule may have many alternative low-energy shapes. However, biochemists only know whether a molecule is qualified to make a drug or not, without knowing which of its alternative low-energy shapes responses to the qualification. Distinctly, if FSFD is a multiple instance problem, yet using a single instance learning method, the prediction performance may be poor.

Therefore, a good solution to this problem inherent in the FSFD task may select the computational intelligent method in the frame work of multiple instance learning like what has been done in drug activity prediction. This study makes comparisons regarding the performance of three pairs of machine learning algorithms in detecting accounting fraud, which is composed of a singe instance learning algorithm and its corresponding multiple instance learning algorithm. It discloses that the underlying nature of the FSFD problem matches well with multiple instance learning (MIL). Among them, the performance of MIBoost is superior to the state-of-the-art FSFD methods. The rest of the paper is organized as follows. Section 2 briefly reviews some related works. Section 3 provides an insight into the research methodology used. Section 4 reports on experimental results. Section 5 concludes.

Recent research. The accounting audit is an important monitoring mechanism which can help reduce information asymmetry and protect the interests of the principals by providing reasonable assurance that financial statements are free from material misstatements. But FSFD is a difficult task when using a common audit procedure since there is a shortage of knowledge concerning the characteristics of fraud. Therefore, prior research on accounting fraud has generally focused on gaining field knowledge as "red flags" and combining these indicators with quantitative models for assessing the potential for accounting fraud. Compared with the model-driven quantitative method, the data-driven machine learning method is a powerful data analysis tool for FSFD [1, 2], because it can adapt well to a new situation regarding variance of fraud motivations and methods.

Johan Perols compared the performance of six machine learning and popular statistical method in FSFD under different ratios of fraud companies to non-fraud companies and assumptions of misclassification costs. The results showed that support vector machines (SVM) performed well relative to ANN, stacking, C4.5 and bagging [3]. Salama and Omar proved that the proposed back propagation based artificial neural networks model can be used in the discovery of manipulation and fraud prediction in the account balances by comparing the predicted values and the actual values [4]. Lin, C.C. et al. examined all aspects of fraud triangle using the data mining techniques which include Logistic Regression, Decision Trees (CART), and Artificial Neural Networks (ANNs) and employ the available and public information on proxy variables to evaluate such attributes as pressure/incentive, opportunity, and attitude/rationalization. Empirically, the ANNs were not only of the highest accuracy, but also of the lowest type II error among them [5].

Though the conclusions concerning the performance of machine learning methods used in FSFD disagree with the above-mentioned, there is common ground that they are all constructed as accounting fraud detectors under the conventional supervised learning framework, in which one instance is associated with one label without considering the input ambiguity of accounting fraud data like 10-k. But in multiple instance learning, the training data is a set of labeled bags, and each bag contains several instances. A bag is labeled negative if all the instances in it are negative. On the other hand, if a bag contains at least one positive instance, it will be labeled positive. Clearly, this formulation of multiple instance learning is helpful to handle input ambiguity of data. There are many multiple instance learning algorithms which have been proposed, such as diverse density [6], CitationkNN [7], miSVM [8], miGraph [9] and MIBoosting [10] and they have been applied to a wide spectrum of applications ranging from content-based image retrieval and web index page recommendation to robot control and event prediction. Therefore, this study tries to introduce three dominant multiple instance learning, which are MIBoost, miGraph and CKNN into accounting fraud field to cope with input ambiguity and enhance detecting performance for property of data.

Research methodology. ***Data.*** The government can delegate enforcement powers concerning management fraud to the SEC, which provides a measure of consistency to eliminate difficulties in dealing with different procedures and rules defining accounting fraud. Therefore, companies involved in accounting fraud may be examined as samples in this study according to the SEC's Accounting Series Releases (ASR's), Litigation Releases (Lit) and the Accounting & Auditing Series Releases (AAER's).

There are some principles which are as follows. On the one hand, companies are selected as fraud companies by meeting three conditions at the same time, which are violating section 10(b) and 10b-5 of securities act of 1934, violating the anti-fraud provisions and falsifying the accounting records. On the other hand, companies are excluded due to financial industry, lack of data, no mentioning of a fiscal year, only concerning violations of quarterly reporting, and a shortage of matching companies. In addition, each fraud company is matched with a non-fraud company of a similar size on the basis of the industry and time period to control for external factors, since companies in the same industry are subject to accounting and reporting requirement in the similar business environment.

After defining the principles, the SEC dockets are searched to gain the fraud companies from 1999 to 2009, including Litigation releases from LR-16014 to LR-21357 and AAER's from AAER-1190 to AAER-3093. Non-fraud companies are randomly drawn from COMPUSTAT companies that are in the same industry (same four-digit SIC code) as a fraud company. And then, the DNUM classification in COMPUSTAT with the companies' individual 10-Ks and Moody's industry summaries to detect any noticeable discrepancies are checked. All summaries agree with the DNUM classification. At last, the dataset in this study includes 484 fraud companies as well as 902 normal companies to form 4158 instances. Since one bag is constructed for one company with three instances, which is an annual report of the company, 1386 bags are generated.

Variables. This study identifies 26 financial statement ratios/variables commonly used in prior studies, which seem to measure the following five aspects of a company:

1. Financial Condition. Poor financial condition may be a motivation for improving the appearance of the company's financial position, gaining as many resources as possible before termination, or reducing the threat of loss of employment. Hence, Altman's Z (Z-SCORE) is utilized as a measurement of a company's financial condition and calculated based on information from the year prior to the year of fraud occurrence.

2. Financial Performance. The expectation to maintain or improve past levels of profitability, regardless of what those levels were like, may be a motivation for accounting fraud, especially if not met by actual performance. Hence, financial performance is measured using return on assets (ROA), which is calculated as net income before extraordinary items in the year prior to the occurrence of the fraud divided by total assets at the end of that year. The return on equity (ROE), return on sale (ROS) and retained earnings/total assets (RETA) are measured regarding the financial performance, too.

3. Debt Structure. A high debt structure may be a motivation for manipulating the financial statements to shift the risk from equity owners and managers to debt owners. It means that a high debt ratio may increase the probability of accounting fraud. Hence, the logarithm of Total Debt (LOGDEBT), the Debt to Equity (DEBTEQ) ratio and the Total Debt to Total Assets (TDTA) ratio are used to measure the levels of debt corresponding to the probability of accounting fraud.

4. Receivable/Inventory. Subjective judgment involved in estimating uncollectible accounts and obsolete inventory may be a motivation for accounting fraud. Hence, the ratio Account Receivable/Sales (RECSAL), the ratio Accounts Receivable/Accounts Receivable for two successive years (RETREND), the ratios Inventory/Sales (INVSAL) and Inventory to Total Assets (INVTA) are used to detect these tactics.

5. Consistent Growth. Growth slowdown or reverses may be a motivation for accounting fraud so as to maintain the appearance of consistent growth. Especially, sustained growth occurs in combination with changes in the company structure and such changes may lead to uncertainty in roles and responsibilities. As a growth measure, the Sales Growth (SALGRTH) ratio is used.

In this study, some additional financial indexes are examined in FSFD. These variables are: net profitability/sale (NPSAL), the ratio of plant property&equipment (net fixed assets) to total assets (NFATA), sales to total assets (SALTA), Current Assets/Current Liabilities (CACL), Net Income/Fixed Assets (NIFA), Cash/Total Assets (CASHTA), Quick Assets/Current Liabilities (QACL), Earnings Before Interest and Taxes (EBIT) and Long Term Debt/Total Assets (LTDTA), the ratio Sales minus Gross Margin (COSAL), the ratio Gross Profit/Total Assets (GPTA), Logarithm of Total Assets (LTA) and Working Capital (WCAP).

In total, we compiled 26 financial variables. And then two methods were used to analyse how much each variable influences the induction. The former tests whether the differences between the two classes were significant for each variable. If the difference was significant with low p-value, the variable was considered informative. The latter is ReliefF method. The larger the value of the average ReliefF score was, the more important influence of the variable in the induction was. Table 1 depicts the means, standard deviations, t-values, p-values and average ReliefF score for each variable. As can be seen in Table 1, ten variables presented low p-values ($p \leq 0.05$). These variables were chosen to participate in the input vector, while the remaining variables were discarded. As for the latter, average ReliefF scores were ranked descendingly and the ten first variables were only chosen. All the selected variables for two methods were underlined in Table 1.

Methods. FSFD can be regarded as a typical classification problem. Hence, considering the classification and multiple instance problems, three pairs of methods are employed in this research study for their powerful capabilities. These methods are MIBoost vs. AdaBoostM1, miGraph vs. SVM, and KNN vs. CKNN.

Multiple instance boost. The standard way to approach the multiple instance learning problem is to assume that there exists one or several "key" instances in a bag that trigger the bag labels. However, the assumption of MIBoosting algorithm is very simple and intuitive, which is to assume that all instances contribute equally and independently to a bag label. Naturally, the process of predicting the label of a bag is generated in two stages. The first stage determines class probabilities in a bag for each individual instance, and the second stage combines these estimates to assign a class label to the bag. Boosting is an approach to machine learning based on the idea of creating a highly accurate predictor by combining many weak learners – that is, have accuracy only slightly better than random guess. In other words, boosting constructs an ensemble of weak classifiers. Actually, boosting is a family of algorithms, among which the AdaBoost is the most influential ensemble one. And MIBoosting is a multiple instance algorithm by upgrading AdaBoost.M1 algorithm to MI problems, while the weak learner is a standard single-instance learner (e.g. C4.5 decision tree algorithm) in the following. The pseudo code for MIBoosting algorithm is shown as Algorithm 1.

Here, N is the number of bags, and there are n_i instances in the i^{th} ($i = 1, 2, \ldots, N$) bag. x_{ij} denotes that it is the j^{th} ($j = 1, 2, \ldots, n_i$) instance in the i^{th} bag. We assume that the label of a bag is either 1 or −1, rather than 1 or 0. Let us explain the details. Two important prob-

Table 1

Statistic, P-values and Average ReliefF score of input variables

Variables	Mean Fraud	SD Fraud	mean non-Fraud	SD non-Fraud	T-test	P-value	Relief Score
Z-SCORE	–63.8142	860.074	–57.6281	1045.5786	14.0318	0	–0.00493
LOGDEBT	0.7153	3.215	0.0091	4.1641	0.2047	0.8378	–0.00671
DEBTEQ	–4.1068	178.2105	–5.6258	272.3411	6.072	0	0.00016
TDTA	–39.3106	642.8996	–51.5364	717.5676	0.2164	0.8287	–0.00054
SALGRTH	102.5099	1972.4517	201.7082	1426.5064	0.561	0.5748	0.00323
RECSAL	–0.0391	1.0332	–0.004	0.3	1.6934	0.0905	0.00112
RETREND	–4.1594	324.9175	0.1074	158.38	1.2649	0.2061	0.00539
INVSAL	0.2215	2.1085	0.1654	1.9961	0.4713	0.6375	–0.00037
INVTA	–41.19	641.7244	–51.6175	717.5617	0.8335	0.4047	1.00E–05
COSAL	1518.6704	9589.5252	2960.0956	10375.7884	0.479	0.632	0.00459
GPTA	–41.7293	641.9525	–51.4851	717.58	4.4888	0	1.00E–05
RETA	–79.5145	834.0883	–94.8305	983.3087	0.448	0.6541	–0.00627
ROS	–4.3863	43.9764	–1.2118	15.002	0.5296	0.5964	–0.00026
ROE	0.7065	26.4354	–7.5759	271.8895	2.6686	0.0077	–1.00E–05
ROA	–45.949	644.0105	–54.1109	720.8862	1.5708	0.1163	0.00094
LTA	2.1372	1.4052	2.2921	1.5114	0.3734	0.7088	0.00377
WCAP	–63.3326	2042.5007	94.7911	1603.1453	3.2995	0.001	0.00286
NFATA	–41.0031	641.7365	–51.4243	717.5756	2.5574	0.0106	0
SALTA	–40.0551	641.8028	–50.4366	717.6699	0.4787	0.6322	1.00E–05
CACL	1.7409	2.6975	4.4555	52.5429	0.4768	0.6335	3.00E–05
NIFA	–15.7146	153.7119	13.086	617.2975	2.6809	0.0074	–0.0003
CASHTA	–41.1955	641.724	–51.579	717.5645	2.2979	0.0216	0
QACL	1.0538	4.6552	3.935	52.5533	0.477	0.6334	6.00E–05
EBIT	37.3668	548.6384	295.9068	1207.9415	2.8313	0.0047	0.00766
LTDTA	2.0122	36.8017	0.2015	0.2569	9.4625	0	–0.00621
ACCRUALA	–40.8455	641.8392	–51.7562	717.5601	1.8749	0.061	7.00E–05

lems of the standard AdaBoost are how to determine the proper weights of c_m's and how to generate the instance-level model h_m's. Likewise, the key problems of MIBoosting algorithm are similar. We regard the sign E as the sample average instead of the population expectation. We are looking for a classifier $F(b)$ that minimizes the exponential loss $E_B E_{Y|B}[e^{-yF(b)}]$. In each iteration of MIBoosting algorithm, we search for the best $f(b)$ to add to the bag-level combined classifier $F(b)$. Due to the assumption in the beginning of MIBoosting algorithm, we expand $f(b)$ into $f(b) = \sum h(x_j/n)$, where $h_j \in \{-1, 1\}$ is the prediction result of the weak learner $h(.)$ for the j^{th} instance in b. We want to generate a weak learner $h(.)$ that maximizes

$$E_w[yh(x_b)/n]=\sum_{i=1}^{N}\sum_{j=1}^{n_i}\left[\frac{1}{n_i}W_i y_i h(x_{ij})\right]. \qquad (1)$$

It is obvious that when $h(x_{ij}) = y_i$ this function can get the maximum. Actually, we can use any weak single-instance learner to generate the model $h(.)$ by assigning the bag-level label and the initial weight W_i/n_i. Thus, we have got $f(b)$, now we consider the proper weights of c_m's. To do this, we can only optimize the loss after the combination

$$\begin{aligned} \mathrm{loss}_{\exp} &= E_B E_{Y|B}[e^{-yF(b)+c(-yf(b))}] = \\ &= \sum_i W_i \exp\left[c_m\left(-\frac{y\sum_j h(x_{ij})}{n_i}\right)\right] = \\ &= \sum_i W_i \exp[(2e_i-1)c_m]. \end{aligned} \qquad (2)$$

Where $e_i = \sum_j 1_{h_m(x_{ij}\neq y_i)/n_i}$, which is computed in Step 4. Note that this function has no global optimum when all $e_i < 0.5$. So if it happens, MIBoosting algorithm will go directly to the end (Step 10). By using numeric optimization, we can get the optimal c_m's in Step 6. Finally, MIBoosting algorithm updates the bag-level

Algorithm 1. MIBoosting Algorithm

1: Initialize weight of each bag to $W_i = 1/N$, $i = 1, 2, \dots, N$.

2: for $m = 1$ to M do

3: Set $W_{ij} \leftarrow W_i/n_i$, assign the bag's class label to each of its instances, and build an instance-level model

4: $h_m(x_{ij}) \in \{-1, 1\}$. Within the i^{th} bag (with n_i instances), compute the error r rate by counting the number of misclassified instances $e_i \in [0, 1]$ within that bag, i. e. $e_i = \sum_j 1_{(h_m(x_{ij \neq y_i}))} \big/ n_i$.

5: if $e_i < 0.5$ for i's, go to Step 10.

6: Compute $c_m = \arg\min \sum_i W_i \exp[(2e_i - 1)c_m]$ using numeric optimization.

7: if $(c_m \leq 0)$, go to Step 10.

8: Set $W_i \leftarrow W_i \exp[(2e_i - 1)c_m]$ and renormalize so that $\sum_i W_i = 1$.

9: end for

10: return $\operatorname{sign}(\sum_j \sum_m c_m h_m(x_j))$.

weight. The more misclassified instances occur in a bag, the greater weight the bag will have in the next iteration. It is analogous to the updating weight process of the standard AdaBoost algorithm at the instance-level.

Multiple instance graph. Almost all multiple instance learning algorithms treat instances in the bags as independently and identically distributed. The instances in a bag, however, are rarely independent in real tasks. There are two simple yet effective methods, i. e. miGraph and MIGraph, to solve the problem of multiple instance learning by treating instances as non-i.i.d.samples. Their basic idea is to regard each bag as an entity to be processed as a whole, and regard instances as inter-correlated components of the entity. miGraph is one of the two methods we mentioned above, which implicitly constructs graphs by deriving affinity matrices and defines an efficient graph kernel considering the clique information. The bag here is denoted by X_i. We can calculate the distances between pairwise instances by using Gaussian distance and derive an affinity matrix W_i by comparing the distances with a threshold δ which is given by the average distance in the bag. The key of miGraph, the kernel k_g, is defined by two given bags X_i and X_j which contain n_i and n_j instances respectively as follows

$$k_g(X_i, X_j) = \frac{\sum_{a=1}^{n_i} \sum_{b=1}^{n_j} W_{ia} W_{jb} k(x_{ia}, x_{jb})}{\sum_{a=1}^{n_i} W_{ia} \sum_{b=1}^{n_j} W_{jb}}, \tag{3}$$

$$\text{where } W_{ia=1} \Big/ \sum_{u=1}^{n_i} w_{au}^{i},\ W_{jb} = 1 \Big/ \sum_{v=1}^{n_j} w_{bv}^{j}$$

$$\text{and } k(x_{ia}, x_{jb}) = \exp\left(-\gamma \left\| x_{ia}, x_{jb} \right\|^2\right).$$

If the distance between the instances x_{ia} and x_{iu} is smaller, then W_i's element at the a-th row and u-th column is set to 1, and 0 otherwise. Thus, we can measure the similarity between the two bags by calculating the kernel k_g. Due to the lower computational complexity of miGraph's kernel compared to MIGraph's kernel, miGraph algorithm will be a better choice for FSFD.

Citation KNN. There are two variants of the K-nearest neighbour algorithm, Bayesian-kNN and Citation-kNN, solving the multiple instance learning problems. Here, we just review Citation-kNN algorithm which has better performance than Bayesian-kNN algorithm. In order to use the key idea of K-nearest neighbour algorithm, it must transform the distance between pairwise instances to the distance between pairwise bags. The minimum Hausdorff distance was used as the bag-level distance metric in Citation-kNN algorithm. The distance between pairwise bags is defined like this

$$F1 - \text{score} = 2 \cdot \text{precision} \cdot \text{recall}/(\text{precision} + \text{recall});$$

$$\operatorname{Dist}(A, B) = \underset{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}{MIN}(\operatorname{Dist}(a_i, b_j)) = \min_{a \in A} \min_{b \in B} \|a - b\|, \tag{4}$$

where A and B are two different bags, a_i $(1 \leq i \leq m)$ and b_j $(1 \leq j \leq n)$ are the instances from each bag. Therefore, the problem of measuring the distance between bags is, in fact, the problem of measuring the distance between the different feature vector sets. Note that when it predicts the label of a new bag, the Citation-kNN algorithm considers not only the bags as the nearest neighbors of the new bag, but also the bags that count the new bag as their neighbours which is analogous to the conception of "citation" in scientific literature. Although the Citation-kNN algorithm has better performance while predicting the labels of bags, it is unable to predict the labels of instances unlike the Diverse Density algorithm. However, the Citaion-kNN algorithm must save the whole training data set in memory in order to measure the distances during the test. Obviously, it will cost almost no training time, but its storage overhead and testing time overhead are very large.

Evaluation metrics. FSFD is a binary classification problem, in which the outcomes are labeled either as positive (P) or negative (N) corresponding to fraud or non-fraud. There are four possible outcomes from a binary classifier. If the outcome from a prediction is P and the actual value is also P, then it is called a true positive (TP); however, if the actual value is N then it is said to be a false positive (FP). Conversely, a true negative (TN) has occurred when both the prediction outcome and the actual value are N, and false negative (FN) is when the prediction outcome is N while the actual value is P. And then, the accuracy, F1 score and the area under the ROC (receiver operating characteristics) curve (AUC) can be defined as follows based on the above definitions

$$\text{Acurracy} = (TP + TN)/(P + N), \tag{5}$$

where precision $= TP/(TP + FP)$ and recall $= TP/(TP + FN)$. Accuracy is selected for its being a basic metric of classification. Considering there may be classification imbalance problem in data, the F1 score is selected for its being a harmonic means of the precision and recall, too.

In the signal detection theory, a ROC is a graphical plot which illustrates the performance of a binary classifier system as its discrimination threshold is varied. It is created by plotting the fraction of true positives out of the positives ($TPR = TP/(TP + FN)$) vs. the fraction of false positives out of the negatives ($FPR = FP/(FP + TN)$), at various threshold settings. TPR is also known as recall, and FPR is one minus the specificity or true negative rate. When using normalized units, AUC is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one. Considering that fraud instance is more important than non-fraud instance, the AUC is selected.

The analysis of experiments and results. In this section, we prove that FSFD is a multiple instance learning problem. Data set I and data set II are constructed by feature selection for the original data set (Section 3.1) according to P-values and ReliefF scores respectively (Section 3.2). For data set I, we randomly sample $i/10$ bags to create the training set while the remaining $(1 - i/10)$ bags are used for testing, where i is from 9 to 1. Like this, we can yield 9 partitions denoted by {I-1, ..., I-9} and {II-1, ..., II-9} for data set II, too.

To make a fair comparison of multiple instance learning algorithms (Section 3.3) with evaluation metrics (Section 3.4), we suppose that the label of each instance in bag is the same as the label of the bag in single instance learning. All algorithms are set to the best parameters by 5-fold cross validation on training sets. Specifically, for AdaBoostM1, the base classifier is set to Decision Stump, the percentage of weight mass to base training and the number of iterations are fixed to 100 and 50; For MIBoost, the base classifier is set to Naive Bayes, the maximum number of boost iterations is set to 50; For LibSVM, the parameter c and γ are set to 120 and 0.8; For migraph, the parameter c and γ are set to 80 and 1.1, the threshold is set to 0.2; For KNN, the number of neighbours is set to 4; For CKNN, the number of references and citers are set to 5 and 1, respectively.

The training/test partition is randomly generated 20 times, and the average performance is recorded. Table 2 shows the accuracy (with standard deviations) of the various methods. The best performance (paired t-tests at 95 % significance level) and its comparable results are bolded. It can be seen that multiple instance learning method is significantly better than single instance learning method correspondingly on partitions I-1. That multiple instance learning method is used to replace the single instance learning method in FSFD correspondingly, such as MIboost vs. AdaboostM1, migraph vs. svm and CKNN vs. KNN, which lead to the performance enhanced by 5 %, 2 % and 3 % respectively.

To study the influence of the amount of training data, we conduct experiments using the same setting as I-1 from I-2 to I-9. The average accuracy of partitions in Table 2 and Fig. 1, *a—c* both show that as the variation of the amount of training data, multiple instance learning methods are consistently better than single instance learning methods. The MIboost and migraph, two multiple instance learning methods, achieve highly competitive performance. In particular, MIboost has great advantage over other methods, is more obvious and is less sensitive to the variation of the amount of training data. It means that it can work well even though there are few training data, which is a universal phenomenon in FSFD.

For Fig. 1, *a—c*, X-axis is the subset of dataset I from I-1 to I-9, Y-axis is the value of Evaluation Metrics such as Accurancy, F1 score and AUC of Multiple instance learning and single instance learning algorithms. For Fig. 1, *d—f*, X-axis is the subset of dataset II from II-1 to II-9, Y-axis is the value of Evaluation Metrics

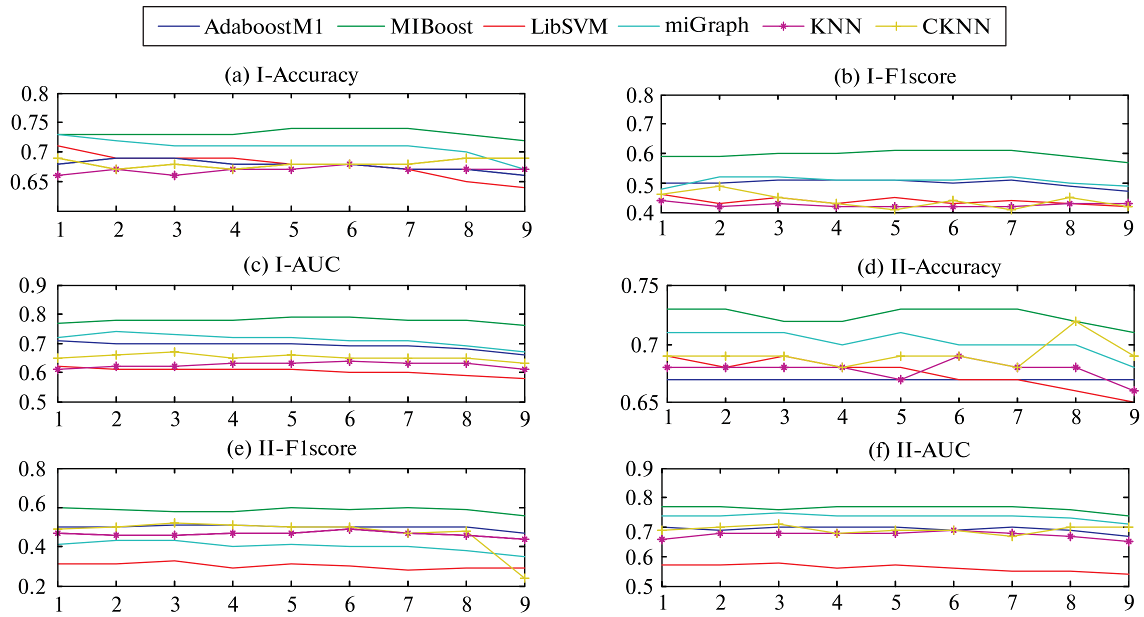

Fig. 1. Influence of amount of training data on metrics

such as Accurancy, F1 score and AUC of Multiple instance learning and single instance learning algorithms

To further investigate the classification results, we conduct paired *t*-tests at 95 % significance level and summarize the win/tie/loss counts of MIBoost versus other methods in Table 2. Paired *t*-tests at 95 % significance level denote that it achieves 9 wins, 0 tie and 0 loss when compared to three single instance learning methods (AdaboostM1, SVM and KNN) and a multiple instance learning method (CKNN), and 5 wins, 4 ties and 0 loss when compared to a comparable multiple instance learning method (miGraph).

Considering that the number of fraud samples is less than the number of non-fraud samples in data set and the fraud samples are more important than non-fraud samples since it is the goal of detection, we select F1 score and AUC to evaluate the performance of algorithms after accuracy. Table 3 and Table 4 indicate that in either F1 score or AUC, the multiple instance learning method is significantly better than the single instance learning method correspondingly on all partitions. At the same time, MIboost obtains the best performance all along as well.

As mentioned in section 3.2, there are two main feature selection methods in FSFD. We want to know whether the experiment conclusions about data set I constructed by P-values will change when using ReliefF scores, another feature selection method.

Table 2

Accuracy on data set I (mean ± std.). The best performance (paired *t*-tests at 95 % significance level) and its comparable results are bolded. The last line shows the win/tie/loss counts of MIBoost versus other methods (the bigger the value is, the better the performance is)

I-Accuracy	AdaBoostM1	MIBoost	LibSVM	miGraph	KNN	CKNN
I-1	0.68 ± 0.03	0.73 ± 0.04	0.71 ± 0.03	0.73 ± 0.03	0.66 ± 0.02	0.69 ± 0.03
I-2	0.69 ± 0.02	0.73 ± 0.02	0.69 ± 0.02	0.72 ± 0.02	0.67 ± 0.02	0.67 ± 0.02
I-3	0.69 ± 0.02	0.73 ± 0.02	0.69 ± 0.02	0.71 ± 0.02	0.66 ± 0.01	0.68 ± 0.02
I-4	0.68 ± 0.02	0.73 ± 0.02	0.69 ± 0.01	0.71 ± 0.02	0.67 ± 0.01	0.67 ± 0.02
I-5	0.68 ± 0.02	0.74 ± 0.02	0.68 ± 0.01	0.71 ± 0.01	0.67 ± 0.01	0.68 ± 0.01
I-6	0.68 ± 0.02	0.74 ± 0.01	0.68 ± 0.01	0.71 ± 0.02	0.68 ± 0.02	0.68 ± 0.01
I-7	0.67 ± 0.02	0.74 ± 0.01	0.67 ± 0.02	0.71 ± 0.01	0.67 ± 0.01	0.68 ± 0.01
I-8	0.67 ± 0.02	0.73 ± 0.01	0.65 ± 0.01	0.70 ± 0.01	0.67 ± 0.01	0.69 ± 0.02
I-9	0.66 ± 0.02	0.72 ± 0.01	0.64 ± 0.02	0.67 ± 0.01	0.67 ± 0.02	0.69 ± 0.01
Average	0.68	0.73	0.68	0.71	0.67	0.68
MIBoost: W/T/L	9/0/0		9/0/0	5/4/0	9/0/0	9/0/0

Table 3

F1score on data set I (mean ± std.). The best performance (paired *t*-tests at 95 % significance level) and its comparable results are bolded. The last line shows the win/tie/loss counts of MIBoost versus other methods (the bigger the value is, the better the performance is)

I-F1score	AdaBoostM1	MIBoost	LibSVM	miGraph	KNN	CKNN
I-1	0.50 ± 0.05	0.59 ± 0.05	0.46 ± 0.06	0.48 ± 0.06	0.44 ± 0.08	0.46 ± 0.08
I-2	0.50 ± 0.04	0.59 ± 0.04	0.43 ± 0.05	0.52 ± 0.05	0.42 ± 0.04	0.49 ± 0.05
I-3	0.51 ± 0.04	0.60 ± 0.03	0.45 ± 0.03	0.52 ± 0.04	0.43 ± 0.03	0.45 ± 0.05
I-4	0.51 ± 0.05	0.60 ± 0.02	0.43 ± 0.04	0.51 ± 0.03	0.42 ± 0.03	0.43 ± 0.03
I-5	0.51 ± 0.03	0.61 ± 0.02	0.45 ± 0.02	0.51 ± 0.03	0.42 ± 0.03	0.41 ± 0.03
I-6	0.50 ± 0.02	0.61 ± 0.02	0.43 ± 0.03	0.51 ± 0.03	0.42 ± 0.03	0.44 ± 0.03
I-7	0.51 ± 0.04	0.61 ± 0.02	0.44 ± 0.02	0.52 ± 0.02	0.42 ± 0.02	0.41 ± 0.04
I-8	0.49 ± 0.04	0.59 ± 0.02	0.43 ± 0.03	0.50 ± 0.03	0.43 ± 0.03	0.45 ± 0.06
I-9	0.47 ± 0.04	0.57 ± 0.03	0.42 ± 0.04	0.49 ± 0.03	0.43 ± 0.03	0.42 ± 0.01
Average	0.50	0.60	0.44	0.51	0.43	0.44
MIBoost: W/T/L	9/0/0		9/0/0	9/0/0	9/0/0	9/0/0

Therefore, extra experiments for data set II are conducted with similar setting. The same conclusion can be drawn according to Fig.1, *d–f*, Table 5, Table 6 and Table 7, that the multiple instance learning method is notably better than the single instance learning method correspondingly on all partitions. MIboost is still the best methods in all metrics and on all partitions.

Conclusions. In this paper, we disclose that the essence of the FSFD when every company has several time sequential Financial Statements to analysis is a typical multiple instance learning problem. Compared with traditional single instance learning methods which have advanced classification and prediction capabilities to facilitate auditors in accomplishing the task of management fraud detection, multiple instance learning methods have better performance and properties. It is proven by the experiment results that multiple instance learning has consistent superiority not only in class-imbalance, but also with a small number of training data. In addition, this significant superiority has been kept under two main feature selection methods. It is clear that a good solution to the problem inherent in the FSFD may also illustrate a promising remedy for other financial problems with similar underlying difficulties. The use of the proposed methodological framework which is the main contribute in this study, could be of assistance to auditors, both internal and external, to taxation and other state authorities, individual and institutional investors,

Table 4

AUC on data set I (mean ± std.). The best performance (paired *t*-tests at 95 % significance level) and its comparable results are bolded. The last line shows the win/tie/loss counts of MIBoost versus other methods (the bigger the value is, the better the performance is)

I-AUC	AdaBoostM1	MIBoost	LibSVM	miGraph	KNN	CKNN
I-1	0.71 ± 0.06	0.77 ± 0.04	0.62 ± 0.03	0.72 ± 0.04	0.61 ± 0.04	0.65 ± 0.05
I-2	0.70 ± 0.03	0.78 ± 0.03	0.61 ± 0.03	0.74 ± 0.03	0.62 ± 0.03	0.66 ± 0.04
I-3	0.70 ± 0.02	0.78 ± 0.02	0.61 ± 0.02	0.73 ± 0.02	0.62 ± 0.03	0.67 ± 0.02
I-4	0.70 ± 0.02	0.78 ± 0.02	0.61 ± 0.02	0.72 ± 0.02	0.63 ± 0.02	0.65 ± 0.03
I-5	0.70 ± 0.02	0.79 ± 0.02	0.61 ± 0.01	0.72 ± 0.02	0.63 ± 0.02	0.66 ± 0.02
I-6	0.69 ± 0.02	0.79 ± 0.01	0.60 ± 0.01	0.71 ± 0.02	0.64 ± 0.02	0.65 ± 0.03
I-7	0.69 ± 0.02	0.78 ± 0.01	0.60 ± 0.01	0.71 ± 0.02	0.63 ± 0.02	0.65 ± 0.02
I-8	0.68 ± 0.02	0.78 ± 0.01	0.59 ± 0.01	0.69 ± 0.02	0.63 ± 0.02	0.65 ± 0.02
I-9	0.66 ± 0.03	0.76 ± 0.02	0.58 ± 0.02	0.67 ± 0.02	0.61 ± 0.03	0.63 ± 0.02
Average	0.69	0.78	0.60	0.71	0.62	0.65
MIBoost: W/T/L	9/0/0		9/0/0	9/0/0	9/0/0	9/0/0

Table 5

Accuracy on data set II (mean ± std.). The best performance (paired *t*-tests at 95 % significance level) and its comparable results are bolded. The last line shows the win/tie/loss counts of MIBoost versus other methods (the bigger the value is, the better the performance is)

II-Accuracy	AdaBoostM1	MIBoost	LibSVM	miGraph	KNN	CKNN
II-1	0.67 ± 0.04	0.73 ± 0.03	0.69 ± 0.02	0.71 ± 0.03	0.68 ± 0.03	0.69 ± 0.03
II-2	0.67 ± 0.03	0.73 ± 0.03	0.68 ± 0.02	0.71 ± 0.02	0.68 ± 0.02	0.69 ± 0.03
II-3	0.67 ± 0.02	0.72 ± 0.02	0.69 ± 0.01	0.71 ± 0.01	0.68 ± 0.02	0.69 ± 0.02
II-4	0.67 ± 0.02	0.72 ± 0.02	0.68 ± 0.01	0.7 ± 0.01	0.68 ± 0.01	0.68 ± 0.02
II-5	0.67 ± 0.02	0.73 ± 0.02	0.68 ± 0.01	0.71 ± 0.01	0.67 ± 0.01	0.69 ± 0.02
II-6	0.67 ± 0.01	0.73 ± 0.02	0.67 ± 0.01	0.70 ± 0.01	0.69 ± 0.01	0.69 ± 0.03
II-7	0.67 ± 0.01	0.73 ± 0.01	0.67 ± 0.02	0.70 ± 0.01	0.68 ± 0.01	0.68 ± 0.05
II-8	0.67 ± 0.01	0.72 ± 0.02	0.66 ± 0.01	0.70 ± 0.02	0.68 ± 0.02	0.72 ± 0.03
II-9	0.67 ± 0.02	0.71 ± 0.02	0.65 ± 0.02	0.68 ± 0.02	0.66 ± 0.02	0.69 ± 0.03
Average	0.67	0.72	0.67	0.70	0.68	0.69
MIBoost: W/T/L	9/0/0		9/0/0	3/6/0	9/0/0	9/0/0

Table 6

F1score on data set II (mean ± std.). The best performance (paired *t*-tests at 95 % significance level) and its comparable results are bolded. The last line shows the win/tie/loss counts of MIBoost versus other methods (the bigger the value is, the better the performance is).

II-F1score	AdaBoostM1	MIBoost	LibSVM	miGraph	KNN	CKNN
II-1	0.50 ± 0.06	0.60 ± 0.06	0.31 ± 0.07	0.41 ± 0.07	0.47 ± 0.05	0.49 ± 0.07
II-2	0.50 ± 0.05	0.59 ± 0.04	0.31 ± 0.06	0.43 ± 0.05	0.46 ± 0.05	0.50 ± 0.06
II-3	0.51 ± 0.03	0.58 ± 0.03	0.33 ± 0.03	0.43 ± 0.03	0.46 ± 0.03	0.52 ± 0.05
II-4	0.51 ± 0.03	0.58 ± 0.03	0.29 ± 0.04	0.40 ± 0.03	0.47 ± 0.03	0.51 ± 0.05
II-5	0.50 ± 0.03	0.60 ± 0.02	0.31 ± 0.04	0.41 ± 0.03	0.47 ± 0.03	0.50 ± 0.05
II-6	0.50 ± 0.02	0.59 ± 0.02	0.30 ± 0.03	0.40 ± 0.03	0.49 ± 0.03	0.50 ± 0.07
II-7	0.50 ± 0.04	0.60 ± 0.02	0.28 ± 0.07	0.40 ± 0.04	0.47 ± 0.03	0.47 ± 0.05
II-8	0.50 ± 0.03	0.59 ± 0.03	0.29 ± 0.06	0.38 ± 0.05	0.46 ± 0.04	0.48 ± 0.11
II-9	0.47 ± 0.08	0.56 ± 0.03	0.29 ± 0.06	0.35 ± 0.06	0.44 ± 0.04	0.24 ± 0.16
Average	0.50	0.59	0.30	0.40	0.47	0.47
MIBoost: W/T/L	9/0/0		9/0/0	9/0/0	9/0/0	9/0/0

Table 7

AUC on data set I I (mean ± std.). The best performance (paired *t*-tests at 95 % significance level) and its comparable results are bolded. The last line shows the win/tie/loss counts of MIBoost versus other methods (the bigger the value is, the better the performance is)

II-AUC	AdaBoostM1	MIBoost	LibSVM	miGraph	KNN	CKNN
II-1	0.70 ± 0.04	0.77 ± 0.04	0.57 ± 0.02	0.74 ± 0.05	0.66 ± 0.04	0.69 ± 0.05
II-2	0.69 ± 0.04	0.77 ± 0.02	0.57 ± 0.02	0.74 ± 0.03	0.68 ± 0.03	0.70 ± 0.03
II-3	0.70 ± 0.02	0.76 ± 0.02	0.58 ± 0.01	0.75 ± 0.02	0.68 ± 0.02	0.71 ± 0.03
II-4	0.70 ± 0.02	0.77 ± 0.02	0.56 ± 0.01	0.74 ± 0.02	0.68 ± 0.02	0.68 ± 0.03
II-5	0.70 ± 0.02	0.77 ± 0.02	0.57 ± 0.01	0.74 ± 0.01	0.68 ± 0.02	0.69 ± 0.03
II-6	0.69 ± 0.02	0.77 ± 0.01	0.56 ± 0.01	0.74 ± 0.01	0.69 ± 0.02	0.69 ± 0.04
II-7	0.70 ± 0.02	0.77 ± 0.01	0.55 ± 0.02	0.74 ± 0.02	0.68 ± 0.02	0.67 ± 0.05
II-8	0.69 ± 0.02	0.76 ± 0.02	0.55 ± 0.02	0.73 ± 0.02	0.67 ± 0.03	0.70 ± 0.05
II-9	0.67 ± 0.04	0.74 ± 0.02	0.54 ± 0.02	0.71 ± 0.03	0.65 ± 0.03	0.70 ± 0.05
Average	0.69	0.76	0.56	0.74	0.67	0.69
MIBoost: W/T/L	9/0/0		9/0/0	8/1/0	9/0/0	9/0/0

stock exchanges, law firms, economic analysts, credit scoring agencies and to the banking system.

Bag generators as the preprocessing step of multiple instance learning problems are more important than the selection of multiple instance learning algorithms in some sense. Therefore, future research will replicate this study by using quarterly financial statements. Using quarterly data may increase the amount of instances in bags, which is beneficial for analyzing data structure deeply to construct complex bags. It hopes to develop a more powerful analytical tool for FSFD by multiple instance learning.

Acknowledgements. This work was supported by the National Social Science Fund Project (No. 12GBL028); The humanities and social science Fund project of the Ministry of Education (No. 11YJAZH083); Science and Technology Plan Project of Hunan Province (No. 2014GK3014); Social Science Fund Project of Hunan Province (No. 14YBA232); Science Research Project of The Education Department of Hunan Province (No. 15C0757); Situation and Decision Counseling Project of Hunan Province (No. 2015BZZ103).

References/Список літератури

1. Sharma, A. and Panigrahi, P. K., 2012. A Review of Financial Accounting fraud detection based on Data Mining Techniques. *International Journal of Computer Applications*, Vol. 39, No. 1, pp. 37–47.

2. Song, X. P., Hu, Z. H., Du, J. G. and Seng, Z. H., 2014. Application of machine learning methods to risk assessment of financial statement fraud: evidence from China. *Journal of Forecasting*, Vol. 33, No. 8, pp. 611–626.

3. Perols, J., 2011. FSFD: An Analysis of Statistical and Machine Learning Algorithms. *Auditing: A Journal of Practice and Theory*, Vol. 30, No. 2, pp. 19–50.

4. Salama, A.S. and Omar, A.A., 2014. A Back Propagation Artificial Neural Network based Model for Detecting and Predicting Fraudulent Financial Reporting. *International Journal of Computer Applications*, Vol. 106, No. 2, pp. 1–8.
5. Lin, C.C., Chiu, A.A., Huang, S.Y. and Yen, D.C., 2015. Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments. *Knowledge-Based Systems*, Vol. 89, pp. 459–470.
6. Hong, R.C., Wang, M., Gao, Y., Tao, D.C., Li, X.L. and Wu, X.D., 2014. "Image annotation by multiple-instance learning with discriminative feature mapping and selection", *IEEE Transactions on Cybernetics*, Vol. 44, No. 5, pp. 669–680.
7. Jiang, L.X., Cai, Z.H., Wang, D.H. and Zhang, H., 2014. Bayesian citation-KNN with distance weighting. *International Journal of Machine Learning and Cybernetics*, Vol. 5, No. 2, pp. 193–199.
8. Zhang, Q., Tian, Y.J. and Liu, D.L., 2013. Nonparallel support vector machines for multiple-instance learning. *Procedia Computer Science*, Vol. 17, pp. 1063–1072.
9. Nguyen, D.T., Nguyen, C.D., Hargraves, R., Kurgan, L.A. and Cios, K.J., 2013, mi-DS: Multiple-instance learning algorithm. *IEEE Transactions on Cybernetics*, Vol. 43, No. 1, pp. 143–154.
10. Ali, K. and Saenko, K., 2014. Confidence-rated multiple instance boosting for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2433–2440.

Мета. Виявлення випадків фальсифікації фінансової звітності (FSFD) на основі машинного навчання є дуже важливою проблемою для зниження фінансового ризику та підтримки впорядкованого ринку. Мета даного дослідження полягала в розробці моделі багатоваріантного навчання, що здатна виявляти й передбачати ризик фальсифікації при складанні фінансової звітності.

Методика. Кожна пара складалася з алгоритму одноваріантного навчання та відповідного алгоритму багатоваріантного навчання, що були підготовлені з використанням набору даних 484 шахрайських компаній, а також 902 нормальних компаній з формуванням 4158 варіантів з пункту 8 Форм 10-K Комісії з цінних паперів і бірж США (SEC).

Результати. Емпіричні дослідження показують, що MIBoost, miGraph і CKNN перевершують, алгоритми AdaBoostM1, SVM і KNN, відповідно, у точності, оцінці F1 і площі під кривою робочих характеристик приймача (AUC), що доводить той факт, що алгоритм багатоваріантного навчання може відповідати FSFD краще, особливо при дисбалансі класів і нечисленних повчальних даних.

Наукова новизна. Коли мітка, що виявляє, відповідна за часом локальній фінансовій звітності, додається колективно до груп фінансової звітності однієї компанії, не враховуючи, що ця мітка надається якій-небудь окремій фінансовий звітності, це багатоваріантна проблема. Дослідження представляє собою розробку оригінальної моделі багатоваріантного навчання FSFD.

Практична значимість. У роботі врахований той факт, що деякі аудитори невдоволені алгоритмами навчання на основі одиночних міток, тому що існує багато варіантів в одній компанії без міток. Запропонована модель є більш обґрунтованою та точною.

Ключові слова: *фінансова звітність, виявлення випадків шахрайства, машинне навчання, багатоваріантне навчання miboosting, miGraph CKNN*

Цель. Выявление случаев фальсификации финансовой отчетности (FSFD) на основе машинного обучения является очень важной проблемой для снижения финансового риска и поддержания упорядоченного рынка. Цель данного исследования заключалась в разработке модели многовариантного обучения, которая способна обнаруживать и предсказывать риск фальсификации при составлении финансовой отчетности.

Методика. Каждая пара состояла из алгоритма одновариантного обучения и соответствующего алгоритма многовариантного обучения, которые были подготовлены с использованием набора данных 484 мошеннических компаний, а также 902 нормальных компаний с формированием 4158 вариантов из пункта 8 Формы 10-K Комиссии по ценным бумагам и биржам США (SEC).

Результаты. Эмпирические исследования показывают, что MIBoost, miGraph и CKNN превосходят алгоритмы AdaBoostM1, SVM и KNN, соответственно, в точности, оценке F1 и площади под кривой рабочих характеристик приемника (AUC), что доказывает тот факт, что алгоритм многовариантного обучения может соответствовать FSFD лучше, особенно при дисбалансе классов и немногочисленных обучающих данных.

Научная новизна. Когда обнаруживающая метка, соответствующая по времени локальной финансовой отчетности, прилагается коллективно к группам финансовой отчетности одной компании, не учитывая, что эта метка присваивается какой-то отдельной финансовой отчетности, это многовариантная проблема. Исследование представляет собой разработку оригинальной модели многовариантного обучения FSFD.

Практическая значимость. В работе учтен тот факт, что некоторые аудиторы недовольны алгоритмами обучения на основе одиночных меток, потому что существует много вариантов в одной компании без меток. Предложенная модель является более обоснованной и точной.

Ключевые слова: *финансовая отчетность, выявление случаев мошенничества, машинное обучение, многовариантное обучение miboosting, miGraph CKNN*

Рекомендовано до публікації докт. техн. наук В.В. Гнатушенком. Дата надходження рукопису 18.06.15.