JMLR: Workshop and Conference Proceedings 45:173–188, 2015 ACML 2015

Detecting Accounting Frauds in Publicly Traded U.S. Firms: A Machine Learning Approach

Bin Li BINLI.WHU@WHU.EDU.CN
Economics and Management School, Wuhan University, Wuhan, P.R. China 430072

Julia Yu JULIAYU@NTU.EDU.SG
Division of Accounting, Nanyang Business School, Nanyang Technological University, Singapore 639798

Jie Zhang ZHANGJ@NTU.EDU.SG
School of Computer Engineering, Nanyang Technological University, Singapore 639798

Bin Ke BIZK@NUS.EDU.SG
Division of Accounting, Business School, National University of Singapore, Singapore 119245

Editor: Geoffrey Holmes and Tie-Yan Liu

Abstract

This paper studies how machine learning techniques can facilitate the detection of accounting fraud in publicly traded US firms. Existing studies often mimic human experts and employ the financial or nonfinancial ratios as the features for their systems. We depart from these studies by adopting raw accounting variables, which are directly available from a firm's financial statement and thereby can be easily applied to new firms at low cost. Further, we collected the most complete fraud dataset of US publicly traded firms and labeled the fraud and non-fraud firm-years. One key issue of the dataset is that the data is extremely imbalanced, in which the fraud firm-years are often less than one percent. Without re-sampling the data, we further propose to tackle the imbalance issue by adopting the techniques of imbalanced learning. In particular, we employ the linear and nonlinear Biased Penalty Support Vector Machine and the Ensemble Methods, both of which have been proved to successfully handle the imbalance issue in the machine learning literatures. We finally evaluate our approach by conducting extensive empirical studies. Empirical results show that the proposed schema can achieve much better performance, in terms of balanced accuracy, than the state of the art. Besides the performance, our approaches can also compute very fast, which further supports their practical deployment.

Keywords: Fraud Detection, Ensemble Methods, Machine Learning Application

1. Introduction

Machine learning has been applied in computational finance (Li and Hoi, 2014), among which one of the most interesting and challenging ones is the detection of accounting frauds in publicly traded firms (Ravisankar et al., 2011; Abbasi et al., 2012). Accounting frauds committed by insiders (i.e., managers and controlling shareholders) of publicly listed firms are a worldwide problem. If not detected and prevented on a timely basis, frauds can cause significant harms to stakeholders of the firms directly linked to the frauds (e.g., Enron and WorldCom). Frauds can also negatively affect non-fraudulent firms indirectly

because the latter often have to compete with fraudulent firms for investors' scarce capital or customers. Furthermore, due to information asymmetry between corporate insiders and outside investors, the financial market's uncertainty about the existence of frauds may hinder the normal functioning of a country's financial markets and economic growth.

Unfortunately, accounting frauds are rare and therefore difficult to detect. For example, the frequency of detected accounting frauds among publicly traded U.S. firms is typically less than one percent. Even if a fraud is detected, it is usually too late and the real damage is already done by the time of the fraud's disclosure. Hence, an important research question in academic research is **to develop effective methods to detect corporate accounting frauds on a timely basis so that the extent of damages from such frauds can be minimized**.

The objective of this study is to develop an effective approach based on machine learning techniques and a sample of publicly traded U.S. firms over the period 1991-2005. While there are useful non-financial indicators of frauds (e.g., an executive's personal behavior), we use only publicly available financial data as inputs for two reasons. First, fraud detection models based on publicly available financial data can be readily applied to any publicly traded firm at low costs. Second, most prior accounting fraud research also relies on publicly available financial data. Hence, the performance of our methods can be directly compared with the performance of traditional fraud detection methods.

Two recent literatures have applied machine learning techniques to the fraud detection task. Dechow et al. (2011) developed a modified logit method working on the features of the financial ratios, while Cecchini et al. (2010) developed a Support Vector Machines with Financial Kernel (SVM-FK) working on the features of expanded ratios. Both suffer from severe drawbacks, which may hinder their real employment. Financial ratios or non-financial ratios are derived from raw accounting variables, the process of which may lose some key information for the fraud detection task and thus decrease an algorithm's detection accuracy. Thus, in this paper, we propose to employ raw accounting variables as our features, which keep firms' key information and are directly available from public firms' financial statements. Besides the features, we also propose to detect the fraud using the nonlinear Support Vector Machines (Burges, 1998) and the Ensemble methods (Liu and Zhou, 2013), which have been proved to realise better performance for imbalanced scenarios, which is much needed for the fraud detection task. Lastly, existing experiments in their studies often have crucial pitfalls, such as hindsight matching of non-fraud samples and hindsight selection of parameters. We therefore address each of them, and evaluate our proposed features and approaches in a practical and applicable way.

In all, the contributions of this paper can be summarized as follows:

1. We are the first to employ raw accounting variables as features for fraud detection,and we have collected a new and most complete dataset for the fraud detection research;

2. We comprehensively studied the task using the latest machine learning techniques, including the nonlinear Support Vector Machines and the Ensemble methods;

3. Experimental results show that our proposed frameworks, including features and algorithms, perform much better than the state of the art in terms of *balanced accuracy*.

The rest of the paper is organized as follows. Section II reviews the related studies on the fraud detection problem and gives an overview of our proposed novel machine learning system. Section III studies the features in existing studies and proposes to use raw accounting variables. Section IV presents the machine learning methods used in our studies. Section V conducts an exhaustive set of experiments to show the effectiveness of the proposed machine learning approaches. Section VI concludes the current work and proposes future directions.

2. Background and the Proposed System

We view the fraud detection problem as a binary classification task. In the fraud detection task, each firm-year is represented by (x_i, y_i), $i = 1, \ldots, n$, where n denotes the number of training firm-years, x_i is feature vector representing the firm-year and y_i labels the firm-years with x_i as fraud (+1) and non-fraud (-1). The basic idea of a classifier is to fit a discriminant function, which returns +1 or -1 and thus separates the fraud from non-fraud. When a new firm-year comes, it is fed into the discriminate function, which then outputs the label of the firm-year. If the predicted label is the same the the true label of the firm-year, then the prediction is correct.

2.1. Literature Review

There have been plenty of research on fraud detection. A proper model for fraud detection consists of the features and the algorithms. Green and Choi (1997) employed *Neural Networks* (NN) techniques to solve the task. Using five financial ratios chosen by experts, their sample consists of 46 frauds and 49 nonfrauds. Moreover, one typical drawback of NN is its poor interpretability and thus nobody can explain what the detection rules are. Summers and Sweeney (1998) proposed a *Cascaded Logit* model, and used financial ratios and variables for insider trading as the features. The dataset used in their study consists of 51 non-fraudulent firms and 51 fraudulent firms. Using signals for insider trading may prevent the schema from real application, because such information is not publicly available. Beneish (1999) proposed a *Probit* model and a *Weight Exogenous Sampling Maximum Likelihood* (WESML) model. Featuring with eight quantitative financial ratios, the dataset consists of 50 frauds and 1758 non-frauds. Bell and Carcello (2000) developed a detection model based on *Logistic Regression* (LR). They used some subjective risk factors, such as weak internal control environment, as the features. It requires additional inside information and subjective judgement, which hinder its practical deployment.

Cecchini et al. (2010) proposed to tackle the challenge using *linear Support Vector Machine with financial kernel* (SVM-FK). Its key contribution is to propose a financial kernel, which expands the raw accounting variables to their ratios and growth in a systematic way, and their features are composed of various generated financial and non-financial ratios. Their dataset consists of 122 fraudulent firms and 6427 non-fraudulent firms. The authors solve the imbalance issue by leveraging an inner mechanism of SVM, or tuning the penalty of misclassifying a fraud firm-year. Dechow et al. (2011) collected a new collection of fraud dataset of financial ratios. Their dataset consists of 494 fraudulent firms and

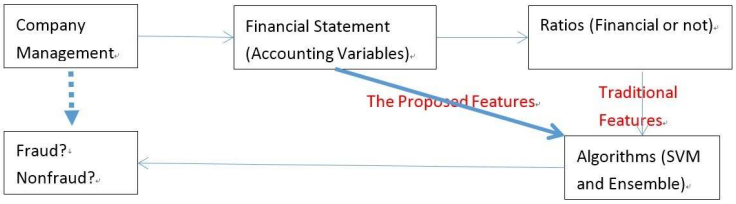

Figure 1: Illustration of the proposed fraud detection system.

Table 1: Summary of the features in the literature

Existing Work	Types of Features	#Fraud: #Non-Fraud
Green and Choi (1997)	5 Financial ratios	46:49
Summers and Sweeney (1998)	Financial ratios & insider information	51:51
Beneish (1999)	8 Financial ratios	50: 1758
Bell and Carcello (2000)	Subjective risk factors	77:305
Cecchini et al. (2010)	Expanded financial and nonfinanaicla ratios	122:6427
Dechow et al. (2011)	11 Financial ratios	494:132967

132,967 nonfraudulent firms[1]. They also address the imbalance issue by improving the *Logit model with a F(raud)-score*, which equals the predicted probability of fraud divided by the unconditional probability of the fraud in the training data. The authors further compare the F-score with a threshold, or 1 in default.

2.2. Overview of the Proposed Machine Learning System

Figure 1 illustrates an overview of the proposed system. Because company's management team will not admit that they are committing frauds, so we have to detect the frauds alternatively. As a proxy, financial statements are used to evaluate the management team, and our basic principle is that if the management team is conducting frauds, the financial statement will differ from the normal forms. The auditing industry always tries to detect a fraud using the financial ratios, so the traditional approaches in Section 2.1 often transform the raw accounting variables from the financial statement into certain financial ratios (such as Debt-to-Asset ratio, etc). Then the traditional approaches feed these ratios into a classifier or algorithm such that the machine learning algorithms could obtain a model for the detection. When new firms come, they will be converted into ratios in a similar way and feeded into the trained model. A final "Fraud" or "Nonfraud" will be outputted from the trained model.

Clearly, two components play key roles in the whole system, i.e., features fed into the algorithms and the algorithms to be used. We thus reshape the whole fraud detection system as follows. First, rather than the traditional (financial or non-financial) ratios, we propose to use raw accounting variables from the financial statement. We believe that raw accounting variables can proxy the management team better than the ratios, and if used correctly, they will realise higher detection power than the converted ratios. Second, we propose to use

1. Depending on the settings, another two datasets contain 449 fraudulent and 122,366 non-fraudulent firms, and 354 fraudulent and 88,032 non-fraudulent firms, respectively.

nonlinear SVM and Ensemble methods to detect the frauds, which significantly outperform traditional approaches.

3. Data and Features

For a typical machine learning application, one key component is the features (Domingos, 2012). This section summarizes the features in existing studies and proposes to use a new feature set of raw accounting variables.

3.1. Summary of Existing Features

Table 1 summarizes the features used in the existing studies. It is clear that the extant accounting fraud detection research typically relies on accounting experts to identify potential determinants of frauds, and most fraud determinants identified by accounting experts are financial ratios (see Dechow et al. (2011)). E.g., ratio analysis based on the DuPont model is an integral part of the traditional accounting valuation analysis. Therefore, it is natural to design the algorithms by mimicking the behaviors of human audits and adopt the financial ratios as features for these algorithms. The resulting decision rules, if extractable, are often straightforward to human experts and thus easy to interpret.

However, accounting experts' conversion of raw accounting data into a limited number of financial ratios results in a loss of useful information for the purposes of fraud prediction, which may weaken the detection power of the algorithms. We thus depart from existing studies by using a new set of features.

3.2. New Features: Raw Accounting Variables

To avoid the information loss in converting the raw accounting variables to ratios, we propose to use raw accounting data directly from firms' financial statements. Using raw accounting variables provides the following benefits. First, the raw accounting variables directly come from the financial statement and contains the most complete information we could explore. Without conversion, this set of features will incur no information loss. Second, the raw accounting variables are widely available, even a user without accounting expertise can get these information. Third, the approach can be easily applied to new firms at a low cost.

While there are a lot of raw accounting data items from a company's financial statements, we limit our empirical analyses to only 24 raw data items so that we can compare the performance between our fraud detection models and the traditional fraud detection methods whose fraud determinants are also derived from the same or a similar set of raw accounting data. Specifically, we follow Cecchini et al. (2010), one of the most recent and comprehensive studies on fraud detection, in the selection of the raw accounting data items. We obtained a final list of 24 raw accounting items.

Another important element is the label. In our study, we label the fraudulent cases as "1", which is the main focus of our study, and the non-fraudulent cases as "0". Finally, each of our records is indexed by firm and year. If the firm on a year is announced to be fraud, then we label it as "1" (fraud).

Figure 1 illustrates the two approaches for the fraud detection task. Traditional approaches usually convert the accounting variables to various (financial or non-financial)

Table 2: Summary of the datasets. The years are used to split training and test samples.

Dataset Name	Years	# of all firm-years(%)	# of frauds (%)	# of non-frauds (%)	# of Features
USRaw_0paired	91-05	90172 (100%)	632 (0.7%)	89540 (99.3%)	24
	91-02	72456 (100%)	514 (0.71%)	71942 (99.29%)	24
	03-05	17716 (100%)	118 (0.67%)	17598 (99.33%)	24
USRaw_1paired	91-05	77605 (100%)	548 (0.62%)	77057 (99.38%)	48
	91-02	61662 (100%)	438 (0.71%)	61224 (99.29%)	48
	03-05	15943 (100%)	110 (0.67%)	15833 (99.33%)	48
USRatio	91-05	88875 (100%)	693 (0.78%)	88182 (99.22%)	11
	91-02	70879 (100%)	568 (0.80%)	70311 (99.20%)	11
	03-05	17996 (100%)	125 (0.69%)	17871 (99.31%)	11

ratios and then feed to various classifiers. Our proposed approach, on contrary, directly feeds the accounting variables to the classifiers. To the best of our knowledge, we are the first to employ raw accounting variables as the features for the fraud detection task.

3.3. Dataset

Our initial accounting fraud sample comes from the SEC's Accounting and Auditing Enforcement Releases (AAERs) [2] over the period between May 17th, 1982 and September 1st, 2010. The AAERs cover accounting frauds that occurred during the fiscal years 1971-2008. However, we excluded the years before 1991 because there is a significant shift in U.S. firms' fraud behavior. Besides, the fraud percentage is abnormally low in the last three years 2006-2008, largely due to the fact that our sample of AAERs ends in 2010 but it typically takes five years from the fraud occurrence and the AAER publication date. Hence, we exclude the last three years from our sample. Finally, we refined our dataset ranging from 1991 to 2005.

We therefore collected a list of 24 raw accounting data items of all US public listed firms during the year 1991 to 2005 from Compustat. Following AAER, we thus manually label the fraudulent firm-years as "1" and all other firm-years as "0". Each record is indexed by the firm-year. E.g., Enron in 2000 is fraud. The final dataset is named "USRaw_0paired", which represents the complete dataset. To feed SVM-FK, we follow Cecchini et al. (2010) and further pair the consecutive years in the dataset and obtained "USRaw_1paired". To fairly show the advantage of our use of raw accounting variables, we followed Dechow et al. (2011) and collected another dataset of financial ratios named "USRatio". Table 2 summarizes the three datasets for the task.

Our datasets represent the most likely scenario in real business for the following three salient. First, our datasets is one of the **largest** fraud detection datasets[3]. Second, different from existing studies that manually match several non-fraudulent firms to one fraudulent firm and thus not directly applicable in real life, we are the first dataset without the matching. Third, as shown in Table 2, without any manipulation, our dataset are the most **imbalanced** dataset ever, i.e., there is only 0.7% fraudulent firm-years in all the samples.

2. More details can be found on its website: https://www.sec.gov/divisions/enforce/friactions.shtml.
3. We could cover longer year range (1979-2008), but accounting experts remind us that the years before 1991 could be biases, as stated above.

4. Machine Learning Methods

4.1. Analysis of Fraud Detection

Although frauds lead to serious consequences to investors, regulators, etc., and affect market confidence, one intrinsic property of the fraud detection task is that fraud firm-years are only a minority. For example, reflected in our collected data, the total number of fraud cases in USRaw_0paired is approximately 0.7% of all firm-years. We view the task as a binary classification task, in which a classifier separates frauds from non-frauds. However, as one common assumption of standard classification methods is that all classes are balanced (He and Garcia, 2009), which means similar number of either class, such a skewed dataset poses challenges to fraud detection. Therefore, due to the extremely skewed data, we further formulate the fraud detection task as an imbalanced binary classification task (Liu and Zhou, 2013). Various machine learning methods have been proposed to tackle such challenge, mainly including the data level pre-process techniques, such as under-sampling the majority class or over-sampling the minority, and the algorithm level techniques. For a comprehensive survey, please refer to He and Garcia (2009).

A subsequent problem with such skewed (or imbalanced) data is the performance metric used to gauge a fraud detector. The most common performance metric for classification, e.g., accuracy, is biased on the skewed datasets. For example, one naïve strategy that classifies all firm-years as non-fraud could obtain an extremely high accuracy of 99.33%. However, the strategy is useless to us, as our main concern is to accurately detect fraud firm-years without misclassifying too much non-fraud firm-years. That is, we care about both detection accuracy within the fraud firm-years and the detection accuracy within the non-fraud firm-years. To properly gauge the performance of a detector, we adopt *balanced accuracy* (BAC) (He and Ma, 2013), which is an average of the two detection accuracies within frauds and non-fraud, as our main performance metric, or the average of *sensitivity* and *specificity*, respectively.

4.2. Summary of Existing Methods

This section will review the existing methods in literature. To address the task properly, the methods have to address the imbalanced issue. Existing approaches usually solve the task via two methods, i.e., the data-level or the algorithmic-level. The data-level method, including "undersampling" and "oversampling", is to adjust the data so as the data is balanced and thus can be fed into common classification algorithms. While the studies in the machine learning community often randomly (under/over)-sample the data, the related studies in the accounting community usually adopt the undersampling by manually matching certain properties of the non-frauds to that of frauds, such the same industry or the same firm size, etc. Among existing studies, Green and Choi (1997) and Summers and Sweeney (1998) adopted this approach, by manually selecting a dataset of balanced frauds and non-frauds.

On the other hand, the algorithmic-level method is to modify the classification algorithms such that they can handle the imbalanced data. Without touching the data, this approach has become more popular in recent studies. Based on Logit model, Dechow et al. (2011) proposed a f-score to handle the imbalanced issue embedded in the fraud detection task. While the authors train the model as usual, they define a f-score for all test instances,

which equals to the predicted probability of fraud divided by the unconditional probability of fraud in the training data. Then the firm-year is predicted to be "fraud" if its f-score is greater than 1, and "non-fraud" if the score is less or equal to 1. By comparing with the unconditional probability of fraud, the model could address the imbalanced issue. On the other hand, Cecchini et al. (2010) addressed the imbalanced issue by adopting Biased-Penalty Linear Support Vector Machines (BP-SVM). By tuning a parameter reflecting the cost of misclassify a fraud firm year, its decision boundary could be shift towards the non-fraud firm-years, which thus can address the imbalance issue.

Now let us focus on the two recent studies, i.e., Dechow et al. (2011)'s modified logit and Cecchini et al. (2010)'s SVM-FK. For the modified logit (Dechow et al., 2011), the model is easy to interpret. In essence, it reflects a linear relationship in the 11 specified financial ratios, and its output denotes the probability of fraud. On the other hand, the choices of financial ratios are ad-hoc, depending on the human expert's domain knowledge. When choosing ratios, human expert may incorporate useful information of fraud detection. However, the chosen ratios, as a product of the accounting variables, may lose certain beneficial information. Such information may be important for the task, and ignoring them will lower the detection performance. Moreover, given the financial ratios, nonlinear relationship may exist within the data, which means the fraud may be not linearly separable. In this case, a linear relationship of these financial ratios may not be ideal for fraud detection.

For SVM-FK (Cecchini et al., 2010), there are mainly two drawbacks within their approaches. First, given the observed imbalanced level, the fraud detection task is challenging. To ease the difficulty, existing approaches often matched of fraudulent and non-fraudulent firms, such as one-to-one match (Green and Choi, 1997; Summers and Sweeney, 1998), and SIC match (Cecchini et al., 2010). While one-to-one match is self-explaining of its matching process, the latter matches four-digit SIC codes and year, which allows many non-fraud firms to each fraud firm. Matching the training data is fine, however, matching the test data is unrealistic or must be done in hindsight. Before AAER's announcement, nobody knows the fraud firms, and thus it is impossible to exclude SIC with only non-fraud firms ex-ante. From this aspect, the out of sample tests (Cecchini et al., 2010) do not reflect the fraud detection in reality. During their matching on the test data, they implicitly assume that the fraud firm-years are already known before their test, which is actually not, otherwise no detection is required. Another drawback of SVM-FK is that they choose an optimal parameter via test performance. Cecchini et al. (2010) set one parameter, or $C^{+1} : C^{-1}$. While the authors manually select an appropriate value that achieves the best test performance, they suffer from hindsight bias. However, manually choosing a cost ratio may not be a good solution, as the cost of frauds is determined by the market (or economy), although it is hard to determine the exact value for a specific firm. And in reality, nobody can tune any parameter with future unseen test data.

4.3. Support Vector Machines

4.3.1. Motivations

After choosing the features, we try to build a classification model using Support Vector Machines (SVM) (Cortes and Vapnik, 1995), which is one state of the art classification technique in machine learning. We choose SVM for the following reasons. SVM has been

proved to be efficient in various fields, including pattern recognition (Burges, 1998), and finance (Tay and Cao, 2001), etc. It usually has a good generalization performance (Burges, 1998), which ensures less probability of over-fitting and its classification accuracy in future unseen data. SVM formulation is convex (Its formulation is a standard quadratic programming, which is convex), thus has a unique global solution, but not local solutions in some other techniques, such as Neural Networks; SVM variants can handle the imbalance issue (Veropoulos et al., 1999), which is one intrinsic property of fraud datasets. There are many publicly available SVM packages[4]. As we will adopt in the next section, SVM can be easily extended to solve nonlinear classification problems using a kernel (Burges, 1998).

Although SVM has been proven to be an effective classifier, SVM has some limitations. The biggest limitation of SVM lies in the choice of the kernel (Burges, 1998). It is not an easy task to incorporate domain knowledge; for example, Cecchini et al. designed a financial kernel. However, later we show that it is not necessary. We have several general kernels, such as Polynomial kernel and RBF kernel, but they are not guaranteed to perform well on a specific application. A second limitation is its training speed (Its training time complexity (Chang and Lin, 2011) is around $O(n^2)$ to $O(n^3)$, depending on kernel and solver for the quadratic programming.), which is not computationally scalable to large number of training instances. The third limitation is its poor interpretability in the nonlinear cases (Martin-Barragan et al., 2014), which thus is hard to extract knowledge from SVM models. But our polynomial kernel is interpretable. Fourth, it has several parameters, which cost a lot of effort to tune.

4.3.2. Support Vector Machine for Fraud Detection

The basic SVM tends to learn a binary linear decision function, denoted as $f(x) = \mathbf{w}^T\mathbf{x}+b$, where $\mathbf{x}$ denotes the input vector and $\mathbf{w}$ denotes coefficients to be fitted, by minimizing classification errors and maximizing the geometric margin of decision boundary to a set of points (or so-called support vectors). According to its idea, we can write the primal form of Biased Penalty SVM (BP-SVM) (Veropoulos et al., 1999; Bach et al., 2006):

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \ & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C^{-1}\left(\frac{C^{+1}}{C^{-1}}\sum_{y_i=+1}\xi_i + \sum_{y_i=-1}\xi_i\right) \\ \text{s.t. } & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi \geq 0, i = 1, \ldots, n \end{aligned}$$

where $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ denotes the inverse of geometric margin (equivalently, maximizing the geometric margin) and the latter part denotes the penalty of classification errors (equivalently, minimizing the error). Note that C^+ denotes the penalty of misclassifying positive (or fraud) as negative (or nonfraud), and C^- denotes the penalty of misclassifying negative as positive. By varying both penalties, the formulation can handle the imbalanced issue.

Without loss of generality, the model has two parameters, $C = C^-$ and $p = C_+/C_-$. While C denotes the common factors for the penalty, p refers to the ratio of FN penalty (misclassifying the fraud) and FP penalty (misclassifying the non-fraud). As we often lack

4. refer to http://www.support-vector-machines.org/SVM_soft.html for a list; in our empirical analysis, we use LIBSVM (Chang and Lin, 2011).

the knowledge to decide the ratio of fraud detection error and non-fraud detection error, one common practice in machine learning (Morik et al., 1999) is to set the ratio such that the costs of misclassification all points are equals, or $p = \frac{\#\ of\ nonfraud\ firm-years}{\#\ of\ fraud\ firm-years}$.

4.3.3. Comparison with Cecchini et al. (2010)'s Financial Kernel

In our approach, SVM with polynomial kernel adopts raw accounting variables, and expands the features into

$$\Phi(\mathbf{x}) = (x_n^2, \ldots, x_1^2, \sqrt{2}x_n x_{n-1}, \ldots, \sqrt{2}x_n x_1, \ldots, , \sqrt{2}x_n, \ldots, \sqrt{2}x_1, 1),$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, and n denotes the number of features. It is clearly that the polynomial kernel expands the raw accounting variables into various products of two raw variables.

On the other hand, SVM with financial kernel (Cecchini et al., 2010) pairs two consecutive years, i.e., $\mathbf{x} = (x_{11}, x_{12}, \ldots, x_{1n}, x_{21}, x_{22}, \ldots, x_{2n})$, where the first indexes denotes the year and the second indexes indicate the feature. The financial kernel expands the paired variables into

$$FK(\mathbf{x}) = (\frac{x_{1i}}{x_{1j}}, \frac{x_{1j}}{x_{1i}}, \frac{x_{2i}}{x_{2j}}, \frac{x_{2j}}{x_{2i}}, \frac{x_{1i}x_{2j}}{x_{1j}x_{2i}}, \frac{x_{1j}x_{2i}}{x_{1i}x_{2j}}),$$

where $i, j = 1, \ldots, n, i < j$. Thus, the financial kernel expands the raw accounting variables into their divisions and the growth. In accounting, some divisions of raw accounting variables mean the financial ratios, and the later two terms denote the growth of the ratios. By trying all kinds of divisions, the expanded features contain certain meaningful ratios.

Comparing the polynomial kernel and the financial kernel, we can see their different behaviors. However, as we will see in the experiments, using raw accounting variables itself could realise satisfying performance. It seems that either financial kernel or expansion is not necessary, as they provide only marginal improvement.

4.4. Ensemble Methods

4.4.1. Motivations

Limiting the data to the same 24 raw accounting data items, we next examine whether it is possible to further improve the out-of-sample fraud prediction performance by using more advanced machine learning methods besides SVM. Specifically, we use ensemble learning, one of the main paradigms in machine learning, because ensemble methods have achieved great success in many real-word applications in recent years (Zhou, 2012, see pages 17-19 for a review of applications of ensemble methods). Different from conventional machine learning methods (e.g., SVM methods) which usually generate one single estimator, ensemble method combines the predictions of a set of base estimators (e.g., decision trees) in order to improve the generalizability or robustness over any single estimator. Previous studies (Zhou, 2012) show that ensembles can usually outperform any single base estimator. However, due to the class imbalance issue, conventional ensemble methods usually need to be combined with a sampling technique in order to balance the class distribution of training data by either adding examples to the minority class (over-sampling) or removing examples from the majority class (under-sampling) (Liu and Zhou, 2013).

4.4.2. Ensemble Methods for Fraud Detection

As a mature machine learning paradigm, ensemble method uses a set of classifiers (so-called base learners, which can belong to the same class, as used in this article, or different classes.) to make predictions. The philosophy is "Group of people can often make better decisions than individuals. Note that in traditional scenarios, such as the previous Logit and SVM studies, we only employ one classifier.

As empirical comparison of multiple ensemble methods on imbalanced data (Liu and Zhou, 2013) shows that hybrid approaches have the best performance, we borrow the idea from two hybrid ensemble approaches, or EasyEnsemble and BalancedCascade (Liu and Zhou, 2013). The system contains $m1$ Adaboost classifiers; Each Adaboost contains $m2$ decision trees. We randomly under-sample $m1$ balanced sub-datasets and on each sub-dataset the system trains an Adaboost (Freund and Schapire, 1997) (weak learners of the system) classifier containing $m2$ decision trees (weak learners of Adaboost). The final outputted labels are a combination of the $m1$ Adaboost classifiers. As ensemble methods contain multiple classifiers, tuning all classifiers' parameters becomes extremely difficult. Different from tuning cost parameters in SVM approach (or algorithmic approach for imbalanced learning), we adopt another technique to solve the challenge, or sampling techniques (He and Garcia, 2009). Its basic idea is, as traditional (unadjusted) machine learning algorithms require balanced training data, we randomly sample from the original training dataset, such that the sampled datasets are balanced. It is possible to over-sample the minority cases or under-sample the majority cases. Due to the large number of training firm-years and we require multiple base learners (each requires training individually, which costs significant time), we use under-sampling, which decreases the number of training samples in each dataset and thus is fast to train. Thus, the sub-datasets contain a balanced dataset of both fraud and nonfraud (1:1). In this way, any traditional machine learning algorithm can be used without any adaption.

5. Experiments

To evaluate the effectiveness of the proposed approaches, in this section we implemented and compared with the two latest methodologies.

5.1. Settings

5.1.1. Out-of-Sample Performance

Most existing fraud detection papers in accounting only evaluate the in-sample performance of their models. However, because the in-sample prediction error is a very optimistic estimate of performance in an out-of-sample data set, it is necessary to assess the out-of-sample performance. A common approach to evaluating the out-of-sample performance of a classification model is to perform an n-fold cross validation (Hastie et al., 2003) as follows: (1) split the data into n roughly equal samples (folds), where n is typically set at 10; (2) estimate the fraud detection model using only $n-1$ folds, leaving one fold out for out-of-sample performance evaluation; and (3) the out-of-sample performance of the model is evaluated using the left-out fold. Our fraud data are *inter-temporal* in nature and hence randomly splitting the data as suggested above is less appropriate because such a cross-validation

would destroy the inter-temporal nature. Therefore, we follow (Cecchini et al., 2010; Dechow et al., 2011) by using the last three years 2003-2005 as the out-of-sample test period and 1991-1999 to train and 2000-2002 to validate the fraud detection models. Note that different splitting years will not affect the conclusion of our study.

5.1.2. Performance Metric

As discussed in Section IV-A, we adopt the "Balanced Accuracy" (BAC) to measure the performance of a fraud detection model. Moreover, we also demonstrate the "Sensitivity" and "Specificity", which represent the detection accuracy of the frauds and the non-frauds, respectively.

Note that there are also other types of performance metrics, such as g-mean, which is the geometric mean of sensitivity and specificity, and Area Under Curves (AUC), etc. But using a different measure will not change the conclusion of our paper (please refer to the journal version of this paper for more detail).

5.1.3. Parameters

For all approaches, we choose their parameters using the grid search with validation. For BP-SVM, with our data, we set the value as $p = \frac{71942}{514} = 139.96$, where 71942 refers to the number of nonfraud firm-years in the training data, and 514 refers to the number of fraud firm-years. We choose the other parameter C via 1D grid search and validation. The search range is $C \in \{2^{-4}, 2^{-3}, \ldots, 2^{10}\}$, or 15 possible choices.

Both EasyEnsemble and BalancedCascade contain two parameters, referring to the number of iterations, and the number of base learners, as one iteration generates one base learner. In all our experiments, we set $m2 = 20$, $m1 = 20$.

5.2. Existing Approaches for Comparison

In our empirical studies, we have implemented several combination of algorithms and features. In particular, we implemented SVMs and ensemble methods, including SVM with linear kernel (SVM-Linear), 2-degree Polynomial kernel (SVM-Poly2), 3-degree Polynomial kernel (SVM-Poly3), RBF kernel (SVM-RBF), EasyEnsemble, and BalancedCascade. For features, we mainly used datasets with the raw accounting variables (USRaw_0paired), and used ratio features (USRatio) and paired variables (USRaw_1paired) for comparison.

We also replicated and compared with two latest approaches, i.e., Modified Logit with financial ratios (USRatio + Logit) and SVM-FK with paired features (USRaw_1paried+SVM-FK). Note that SVM-FK has to use the 1paired data, because its financial kernel is designed to convert paired variables into ratios. We do not replicate the earlier approaches, because either their methods are too old to study or their datasets are not publicly available.

5.3. Experimental Results

5.3.1. Out-of-Sample Performance

Table 3 shows the main results of the empirical studies. The first experiment is to replicate Dechow et al. (2011) and Cecchini et al. (2010). The first row, or "Logit+USRatio", denotes the results of Dechow et al. (2011). It has a better detection of fraud than the

Table 3: Balanced accuracy achieved by various approaches. The "Sensitivity" refers to the detection accuracy of the fraud firm-years; The "Specificity" refers to the detection accuracy of the non-fraud firm-years; The "Balanced Accuracy" equals to the average of "Sensitivity" and "Specificity". "Balanced Accuracy" is the main performance metric to evaluate an approach in the fraud detection problem.

Data	**Methods**	**Performance (%)**		
		Sensitivity	**Specificity**	**Balanced Accuracy**
USRatio	Logit	64	55.99	60
USRaw_1paired	SVM-FK	65.45	74.03	69.74
USRaw_1paired'	SVM-FK	63.64	74.19	68.91
USRaw_0paired	SVM-Linear	70.34	68.96	69.65
	SVM-Poly2	71.19	72.28	71.73
	SVM-Poly3	73.73	72.83	73.28
	SVM-RBF	71.19	74.22	72.70
USRaw_1paired	SVM-Linear	65.45	71.55	68.50
	SVM-Poly2	71.82	73.73	72.78
	SVM-Poly3	68.18	74.57	71.38
	SVM-RBF	69.09	74.57	71.83
USRaw_0paired	EasyEnsemble	75.76	77.10	76.43
	BalancedCascade	72.54	82.52	77.53

detection of non-fraud (64% > 55.99%), and the balanced accuracy equals to 60%. Considering the number of non-frauds, this approach will cause too much misclassification of non-frauds. The second and third rows, or "SVM-FK"+USRaw_1paired", show the results of Cecchini et al. (2010). Note that SVM-FK matched both training and test datasets, which looks ahead the fraud information in the test data. We therefore tested two types of setting. The second row shows the setting of SVM-FK on the data without matching, and the third row shows the setting of SVM-FK on the data with matching the training data. As the fraud information is not available before the testing procedure, we do not show their performance on the matched training and test data. From the results, we can see that SVM-FK outperforms the Logit by around 9%.

Now let us focus on our proposed approaches. We first predict the frauds using the linear and nonlinear SVM with the raw accounting variables, i.e., the "USRaw_0paired" dataset. The results are surprising, with the following observations. First, SVM-Linear with raw accounting variables can realise a balanced accuracy of 69.65%, which is similar to that of SVM-FK's and is much better than that of Logit's. It seems that the financial kernel built in the SVM-FK approach is redundant, and we can achieve similar performance with concise raw accounting variables. Second, as expected, SVMs with nonlinear kernels perform better than linear SVM, outperforming by at most 4% in terms of the balanced accuracy. This result shows that the certain nonlinear relationships do exist in the fraud data, and the nonlinear SVM can exploit such relationships.

To further compare with SVM-FK, we also show the performance of the proposed SVM approaches on the 1paired dataset. The rows indexed by "USRaw_1paired"+"SVM-Linear" and the following three rows shows the results. It is clear that the financial kernel is not necessary, as SVM with linear kernel also realises similar out-of-sample performance.

Table 4: Computational Time. "Validation", "Training", and "Testing" measure the computational time costs during validation, training and test, respectively. All evaluations are based on the same machine and platform (CPU: Intel Xeon 2.53GHz, 16 cores. Memory: 14GB. OS: Window Server 2008 R2. All programs are coded and tested in Matlab 2011b.).

Data	**Methods**	**Performance (%)**		
		Validation	**Training**	**Testing**
USRatio	Logit	1.3	0.7	0
USRaw_1paired	SVM-FK	9.86E+04	1.08E+04	1980
USRaw_1paired'	SVM-FK	915	110	214
USRaw_0paired	SVM-Linear	4.64E+04	9.65E+02	102
	SVM-Poly2	7.88E+04	7.18E+02	86
	SVM-Poly3	1.01E+05	7.50E+02	90
	SVM-RBF	1.14E+05	5.89E+02	97
USRaw_0paired	Ensemble1	NA	98	175
	Ensemble2	NA	422	175

Moreover, SVMs with nonlinear kernels would outperform SVM-FK, which validates that nonlinearity does exist in the fraud data.

To improve the detection performance, we further proposed the Ensemble methods. Our next set of experiments evaluated the performance of Ensemble methods on the raw accounting variables. In particular, we evaluated EasyEnsemble and BalancedCascade, based on "USRaw_0paired" dataset. Clearly, EasyEnsemble achieved a balanced accuracy of 76.43% and BalancedCascade achieved a balanced accuracy of 77.53%, both of which outperform the SVM approach by around 4% to 5%. Moreover, the Ensemble methods outperform in both sensitivity and specificity. Comparing with our benchmarks, the ensemble methods outperform SVM-FK by more than 8%, and outperforms Logit by more than 18%.

These results not only validate the effectiveness of ensemble methods, which outperform the state of the art, but also confirm our proposal that raw accounting variables have better discriminant ability than the (manually) calculated financial or non-financial ratios.

5.3.2. Computational Time

A practical machine learning application often involve computational time efficacy. Table 4 shows the computational time for the main experiments. First, though the Logit benchmark is pretty fast, its balanced accuracy is the lowest. The SVM-FK benchmark expands all raw accounting variables to a set of ratios, and it costs much longer time than Logit. Second, our first approach, the SVM and raw accounting variables, costs similar validation time [5], but costs much less training and test time, due to less features. Third, the ensemble methods cost much less training time than the SVM approach, but cost longer time on test. Nevertheless, the result show that our models could achieve better performance, with less computational time than the performance comparable SVM-FK.

5. We cross validated our model using two dimensions (weight and cost), while we cross validated SVM-FK on one dimension (cost) following its original study.

6. Conclusion

This paper provides a novel business application of machine learning, which detects the accounting frauds in US publicly traded firms. Observing the drawbacks of existing studies, we propose to improve the detection accuracy via features and algorithms. Different from previous studies, we are the first to employ raw accounting variables as features for the task. We further improve the detection accuracy, in terms of balanced accuracy, via latest machine learning techniques. In particular, we employed the ensemble methods for the task, which realises much better balanced accuracy than the state of the art.

Acknowledgments

Part of this research is funded by a Singapore Ministry of Education Tier 2 grant (No. MOE2012-T2-1-045) and a project of the National Natural Science Foundation of China (No. 71401128).

References

A. Abbasi, C. Albrecht, A. Vance, and J. Hansen. Metafraud: a meta-learning framework for detecting financial fraud. *MIS Quarterly*, 36(4):1293–1327, 2012.

F. R. Bach, D. Heckerman, and E. Horvitz. Considering cost asymmetry in learning classifiers. *J. Mach. Learn. Res.*, 7:1713–1741, 2006.

T. B. Bell and J. V. Carcello. A decision aid for assessing the likelihood of fraudulent financial reporting. *Auditing: A Journal of Practice & Theory*, 19(1):169–184, 2000.

M. D. Beneish. The detection of earnings manipulation. *Financial Analysts Journal*, 55(5): 24–36, 1999.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–824, 1998.

C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

M. Cecchini, H. Aytug, G. J. Koehler, and P. Pathak. Detecting management fraud in public companies. *Management Science*, 56(7):1146–1160, 2010.

C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2:27:1–27:27, 2011.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

P. M. Dechow, W. Ge, C. R. Larson, and R. G. Sloan. Predicting material accounting misstatements. *Contemporary Accounting Research*, 28:17–82, 2011.

P. Domingos. A few useful things to know about machine learning. *Commun. ACM*, 55 (10):78–87, 2012.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

B. P. Green and J. H. Choi. Assessing the risk of management fraud through neural network technology. *Auditing: A Journal of Practice & Theory*, 16:14–28, 1997.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. New York : Springer, New York, 2003.

H. He and Y. Ma. *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley, 2013.

H. He and E.A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

B. Li and S.C.H. Hoi. Online Portfolio Selection: A Survey. *ACM Computing Surveys*, 36 (3):35:1–35:36, 2014.

X. Liu and Z.-H. Zhou. *Ensemble Methods for Class Imbalance Learning*, pages 61–82. John Wiley & Sons, Inc., 2013.

B. Martin-Barragan, R. Lillo, and J. Romo. Interpretable support vector machines for functional data. *European Journal of Operational Research*, 232(1):146 – 155, 2014.

K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proceedings of ICML*, pages 268–277, 1999.

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*, 50 (2):491 – 500, 2011.

S. L. Summers and J. T. Sweeney. Fraudulently misstated financial statements and insider trading: An empirical analysis. *The Accounting Review*, 73(1):131–146, 1998.

F. E. H. Tay and L. Cao. Application of support vector machines in financial time series forecasting. *Omega*, 29(4):309–317, 2001.

K. Veropoulos, C. Campbell, and N. Cristianini. Controlling the sensitivity of support vector machines. In *Proceedings of IJCAI*, 1999.

X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. McLachlan, A. Ng, B. Liu, P. Yu, Z.-H. Zhou, M. Steinbach, D. Hand, and D. Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008.

Z.H. Zhou. *Ensemble Methods: Foundations and Algorithms*. Taylor & Francis, 2012.