A Model for Detecting Accounting Frauds by using Machine Learning

Minh Nguyen Hoang
Faculty of Information Technology,
Ho Chi Minh City University of
Education, Ho Chi Minh City,
Vietnam
minhnh.khmt302@pg.hcmue.edu.vn

Hoang To Loan Nguyen
School of Accounting, University
of Economics Ho Chi Minh City,
Ho Chi Minh City, Vietnam
loannht@ueh.edu.vn

Hung Nguyen Viet
Faculty of Information Technology,
Ho Chi Minh City University of
Education, Ho Chi Minh City,
Vietnam
hungnv@hcmue.edu.vn

Abstract

This paper aims to develop a machine learning model that enables to predict signs of financial statement frauds by combining the domain knowledge of machine learning and accounting. Inputs of this model is a published dataset of financial statements, and outputs involve the conclusions whether the predicted financial statements indicate the signs of financial statement frauds or not. Currently, XGBoost is recognized as one of the most popular classification methods with fast performance, flexibility, and scalability. However, its default properties are not suitable for fraudulent detecting of imbalanced datasets. To overcome this drawback, this research introduces a new machine learning model based on XGBoost technique, called f(raud)-XGBoost. The proposed model not only inherits XGBoost advantages but also enables it to detect financial statement frauds. We apply the Area Under the Receiver Operating Characteristics Curve and NDCG@k to perform the evaluation process. The experimental results show that the new model performs slightly better than three existing models including logistic regression model that is based on financial ratios, Support-vector-machine model, and RUSBoost model.

1. Introduction

The major function of financial statements is to provide information about an entity's assets, liabilities, equity, income, and expenses that is useful to financial statement users in assessing the prospects of future cash inflows to the entity and in assessing management's stewardship of the entity's resources [1]. However, the financial statements are not always presented fairly and appropriately. Sometimes, for objective factors such as mistakenly entering information into accounting system, or possibly using fraudulent techniques, e.g., misrepresentations of revenues, expenses, inputting inaccurate information in financial statements. Dishonest in financial statements can conduct negative consequences for business and stakeholders, adversely affecting the integrity of financial statements, economic development, causing economic damages to the companies and their stakeholders. Unfortunately, financial statement frauds are difficult to detect because the frequency of frauds is quite low, less than 1% per year. Moreover, even if financial statement frauds are detected, serious damages have usually already been done [2]. For example, the Enron scandal in 2000 and WorldCom in 2002 in United State led to bankruptcy of both companies. Given these incidents, it has become important to be able to detect fraudulent behaviors prior to their occurrence.

The objective of this research is applying a machine learning model to develop a prediction method for frauds by readily available the financial statement data from publicly trade U.S firms. To solve the imbalanced problem, this paper proposes a machine learning algorithm called f-XGBoost. This algorithm is based-on XGBoost which is flexible, powerful, and fast by using CPU threads or GPU core [3]. The benchmark financial data is used for several reasons such as to compare our results with existing models that suggested by Cecchini et al., Dechow et al. and Bao et al. [4], and to develop a low-cost model that can be apply to any publicly traded firms. There two suitable metrics, Area Under Receiver Operating Characteristics Curve and NDCG@k with k is top 1% of the observations are used.

In the next section, the literature review of the research is presented. Then, the section 3 of this paper describes the detail about dataset that is used in the

This research is funded by University of Economics Ho Chi Minh City, Vietnam. The authors are grateful to University of Economics Ho Chi Minh City for the funding of the research

URI: https://hdl.handle.net/10125/79525
978-0-9981331-5-7

research. Because proposed model bases on ensemble learning and gradient tree boosting, so that a brief review ensemble learning and f-XGBoost method will be shown in section 4. In section 5, two metrics, AUC and NDCG@k are explained in details. Section 6 explains the experimental results and shows the comparisons of our results with the state-of-the-art models' [4]. And last section presents conclusion and future works.

The main contribution of this research is providing a new technical method by using machine learning model to detect fraud in financial statements.

2. Related works

In 1997, Green and Choi et al. [5] introduced fraud detection model that was using Neural Network technique to predict fraudulent financial statements from 1982 to 1990. It inputs were five ratio variables: allowances for doubtful accounts/net sales, receivables, net sales/account receivables, gross margin/net sales, account receivables/total assets, and three raw accounting data are net sales, account receivables, allowances for doubtful accounts. Their deep learning model was a back propagation network, which had 3 layers, 8 input nodes, 4 hidden nodes and 1 node output, the learning rate and momentum were both set to 0.1 and epochs were limited to 10,000, its' activate function was Sigmoid logistic. The output was a float number that determines whether a set of financial statements includes signs of frauds or not. If it is greater than the threshold, that means the financial statements are fraudulent and the threshold was set to 0.5. The accuracy of the model was about 74.03% based on their dataset. The limitation of this research was the dataset. It is too small, only contained 46 fraudulent financial statements and 49 non-fraudulent financial statements.

Dechow et al. provided a new technical method based on logistic regression [6] in 2009. The inputs of their model were 5 types of financial variables, which were accruals quality related variables, performance variables, non-financial variables, off-balance-sheet variables, and market-related incentives. Output of the model was F-score (fraud-score), and then compare the predicted F-score with threshold, this threshold usually set at 1.0. which financial statement has F-score greater than threshold would be consider as fraud.

Mark Cecchini et al. introduced Support Vector Machine Model with Financial Kernel in 2010 [7]. This model migrated financial raw data into financial ratios. This research dataset contained 122 fraudulent financial statements from AAERs in period 1999 to 2006. After dropping some items that had more than 25% missing data, their dataset contains only 23 variables that were the input of SVM model. Their research correctly classified 80% fraudulent cases and 90.6% non-fraudulent cases. Value of metric AUC is 0.878 in their dataset. In the experimental in Bao et al. dataset, value of AUC is 0.626. Before training model, they had some pre-processing steps that were changing some 0 value to 0.0001 to avoid dividing by zero exception and removing firms with marge number of missing values.

Chen et al.'s work [8] suggested a new technical to detect Taiwan's misstated firms during period of 2002-2013 by utilizing multiple data mining methods, e.g., Decision tree, Bayesian network, Support Vector machine and artificial neural networks.

Almost previous studies have a drawback that they were testing model within-sample and often emphasizing the causal inference. So, Bao et al. [4] introduced a detection model of fraudulent financial statements by using RUSBoost algorithm. The training dataset had 28 raw data items and 14 ratio-items, which were chosen based on Cecchini et al. and Dechow et al. research [4]. Before training model, the authors changed some misstated firms in both training and testing years into non-misstated firms in training set, due to affect the flexibility of model. Their model detected 16 fraud cases correctly in the testing period from 2003 to 2008.

Recently, Bertomeu et al. [9] have proposed a machine learning model by using GRBT tree method to detect financial misstatements. Bertomeu et al.'s work [9] showed that the machine learning methods not only enabled to detect fraudulent patterns presented in ongoing accounting misstatements, but also had a comparison with other models, such as RUSBoost and Random Forest. Bertomeu et al. also examined one-year and two-year gaps between training and testing periods.

3. Sample dataset

3.1. Sample period

Sample dataset that has been used in this research was published by Bao et al. in their GitHub repository [4]. This dataset contains publicly listed U.S firms from period 1990 to 2014. But in the implementation section, this paper mainly uses the period 1991 to 2008 to perform the training and predicting. We choose the period because the global financial crisis occurred, and U.S Securities and Exchange Commission (SEC) agency started to detect financial frauds around that time. After 2008, SEC turned its focus to Ponzi-like scheme. Besides, another reason for us to choose the period is we wish to compare our results with the models of Yang Bao, Cecchini and Dechow, which are described in the above sections [4].

The dataset contained 146,045 financial reports of publicly trade companies in U.S from 1990 to 2014, including 964 fraudulent financial statements. All raw accounting values were gotten from COMPUSTAT database, which was updated until April 2017.

3.2. Fraud sample

There are many sources to get fraud sample, e.g., University of California-Berkeley Center for Financial reporting and Management (CFRM), the Government Accountability Office's (GAO) earnings restatement, Audit Analytics' (AA) earnings restatement, and the Stanford Securities Class Action Clearinghouse (SCAC). The accounting fraud sample, used in this study, came from the SEC's AAERs was provided by CFRM [4]. Because Karpoff et al. [10] showed that CFRM dataset is the best for identifying all cases of accounting frauds, and prior works [6], [7] also used AAERs data to benchmark their models.

By the time Bao et al. obtained dataset, CFRM covered period from May 1982 to September 2016. And some fraudulent financial statements were dropped due to missing data (all fraudulent financial statements are required to have no missing data), they had to hand-collect some fraudulent observations from SEC website to enrich the dataset. They collected data up to December 2018. But the latest version of the dataset was tabulated fraud observations up to 2014 because SEC needed time to finish their investigations of alleged fraud cases [10]. Figure 1 presents distribution of fraudulent financial statements over 1990 to 2014 in latest version of dataset. It also shows that the percentage of fraudulent firms before 1997 were less than 0.5%. then from 1997 to 2005, a number of fraudulent firms increased, make percentage of fraud became about 1.3%, and in 2008 decreased to about 0.5%. In 2009, fraudulent firms increase to 0.6% then decrease until 2014.

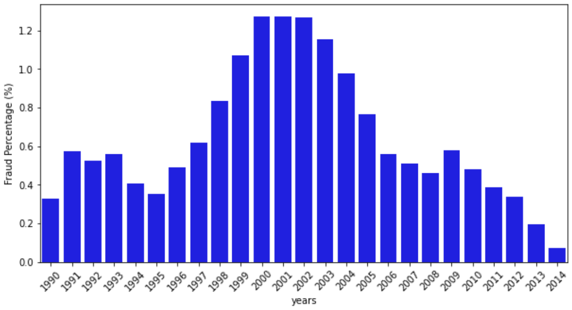

Figure 1. Distribution of fraudulent firms by years over 1990 to 2014

3.3. Sample variables

The list of raw financial variables was selected based on Cecchini et al. [7] and Dechow et al. [6]. Firstly, they were constructed based on table 3 of Cecchini et al. [7], then selected data variables by removing the variables which have more than 25% missing values within sample period 1991-2008 [4], for the reason that the large number of missing values could lead to the impact of model's performance. After that, the sample dataset retained 24 raw financial data variables. To construct list of ratio variables, Bao et al. [4] added some variables described in column "Calculation" of table 3 of Dechow et al. [6] and obtained four more variables. The details can be seen in table 2 of [4]. The latest sample dataset contains 28 raw financial variables described in Table 1.

Table 1. List of 28 raw financial data [4]

Variables	Meaning
act	Current assets, total
ap	Account payable
at	Assets, total
ceq	Common/ordinary equity, total
che	Cash and short-term investments
cogs	Cost of goods sold
csho	Common shares outstanding
dlc	Debt in current liabilities, total
dltis	Long-term debt issuance
dltt	Long-term debt, total
dp	Depreciation and amortization
ib	Income before extraordinary items
invt	Inventories, total
ivao	Investment and advances, other
ivst	Short-term investments, total
lct	Current liabilities, total
lt	Liabilities, total
ni	Net income (loss)
ppegt	Property, plant and equipment, total
pstk	Preferred/preference stock (capital), total
re	Retained earnings
rect	Receivables, total
sale	Sales/turnover (net)
sstk	Sale of common and preferred stock
txp	Income taxes payable
txt	Income taxes, total
xint	Interest and related expense
prcc_f	Price close, annual, fiscal

After collecting 28 raw variables, Bao et al. started to construct list of ratios variables based on table 3 of Dechow et al. [6]. The Table 3 of Dechow et al. [6] suggested five types of variables: "accruals quality related", "Performance", "Nonfinancial", "Off-balance-sheet", and "Market-related incentives". Bao et al. [4] calculated all variables under "accruals quality related" type, except for last four discretionary accrual measures, because Dechow et al. [6] did not use these variables in their subsequent models neither [4]. Five variables of "performance" type were also included in this dataset, except for "*deferred tax expense"* because the variable, "*deferred tax expense"*, needed "*income taxes, deferred"* of raw data to calculate, but it was dropped due to more than 25% its values missing in sample period. *"Actual issuance"* and *"book-to-market"* variables under "market-related incentives" were kept because raw financial variables for those ratio variables were available in COMPUSTAT. Furthermore, *"depreciation index"* was constructed based on formula that Beneish provided [11], and *"retained earnings over total assets"* and *"Earnings before interest and taxes*" from Summers and Sweeney [12]. The latest list of ratio variables is described in Table 2.

Table 2. List of 14 ratio-items in dataset [4]

Variables	Meaning
dch_wc	WC accruals
ch_rsst	RSST accruals
dch_rec	Change in receivables
dch_inv	Change in inventory
soft_assets	Percentage of soft assets
ch_cs	Change in cash sales
ch_cm	Change in cash margin
ch_roa	Change in return on assets
ch_fcf	Change in free cash flows
issue	Actual issuance
bm	Book to market
dpi	Depreciation index
reoa	retained earnings
EBIT	Earnings before interest and taxes

3.4. Serial fraud

Serial fraud is a fraud that spans more than one year. This dataset has about 50% serial fraud cases. Almost of prior works treated serial cases by considering frauds of each year as a company-year. Unlike some single learner models, ensemble learning models are flexible and powerful. Serial fraud may lead to training period and testing period which contain the same fraudulent firms, and overstate the performance of models [4]. So that, Bao et al. dealt with this concern in a different way, recoding fraudulent financial statements in training set, which were spanned from training set to testing set, to non-fraudulent financial statements. Although this approach helped authors avoid the association with serial fraud, it gave measuring problems during training period.

In order to compare the experimental results, the performed procedure for dealing with serial frauds is similar to Bao et al.'s, such as recoding all fraudulent financial statements that spanned from training to testing period during the training step.

4. Proposed method

Nowadays, ensemble learning is one of the state-of-the-art approaches of machine learning and represents as one of the major research trends in machine learning [13]. Ensemble learning has been widely used to solve various real-world problems, especially in finance. For examples, it was used to forecast financial time series [14], to predict financial bankruptcies [15], to forecast financial distresses [16], and so on. One of the crucial reasons to apply the ensemble method is to overcome the problems caused by imbalanced data.

The concept of ensemble learning method is to train multiple sub-models, then combines their results in order to improve the generalizable ability and robustness. Previous studies by Zhou [17] showed that the ensemble method usually performs better than others. In this research, we suggest a scalable and flexible ensemble learning method called f-XGBoost based on decision-tree boosting technique XGBoost, which is widely used in data science to archive state-of-the-art results. Additionally, this method is able to overcome many challenges in machine learning [3]. This is a supervised learning, usually used to predict variable $\hat{y}_i$ with m observed features. Furthermore, XGBoost also provides the insights on cache-aware accessible patterns, data compressions, and the sharding of data in order to build a tree boosting. By combining the mentioned insights, XGBoost uses less resources than other methods. Finally, the technique is also outstanding at handling data with Sparsity-aware Split Finding, therefore, it allows us to handle missing data without pre-processing steps.

4.1. XGBoost

Giving a dataset with has n samples and each sample has m features $D = \{(x_i, y_i)\}(|D| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R})$ and label variable y, a tree-based ensemble model with k additive function has predict output formular:

$$\hat{y} = \sum_{k=1}^{K} f_k(x_i) \qquad (1)$$

Where K is number of trees, $f_k \in F$, and F is the space of Classification and Regression Trees. Each $f(x) = w_{q(x)}$ is an independent tree with structure q and leaf weights w. Unlike other decision-tree methods, each regression tree contains a continuous score on each of the leaves, w_i represents score on i^{th} leaf.

To learn a set of functions used in the model, following regularized objective function is minimized:

$$\mathcal{O} = \sum_{i} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k) \qquad (2)$$

Eq. (2) has two parts. The first part is loss function, which is used for measuring the differences between the predicted values and grow-truth values. The second part is the regularization, which is added to control the complexity of the model, if we set it to 0 then the objective function becomes traditional methods. The Ω is defined as follow:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{t=1}^{T} w^2. \qquad (3)$$

The Eq. (2) includes regularization parameters and cannot be optimized by using traditional methods in Euclidian space because it uses functions as parameters. Instead of this, the model is trained in an additive manner. Formally, let $\hat{y}_i^{(t)}$ be the prediction of i^{th} instance at the t^{th} iteration, the tree f_t will be added to minimize the objective functions, which will improve the model, as below:

$$\mathcal{O}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}^{(t-1)} + f(x_i)\right) + \Omega(f_t) \qquad (4)$$

The above objective function Eq. (**4**) can be optimized by applying the second order in Taylor series approximation:

$$\mathcal{O}^{(t)} \cong \sum_{i=1}^{n} \left[l(y_i, \hat{y}^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i^2(x_i) \right] + \Omega(f_t) \qquad (5)$$

Where $g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ are first and second derivatives of loss function respectively. The constants can be removed to simplify the objective function at t^{th} iteration:

$$\mathcal{O}^{(t)} = \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) \qquad (6)$$

The Eq. (6) is used for calculating the loss between the prediction values and ground truth. Usually, if the values of objective function are lower, the model will perform better.

4.2. f-XGBoost

After we understand how XGBoost works, we continue to examine f-XGBoost. F-XGBoost is XGBoost when we already identified its parameters. In particular, f-XGBoost uses "binary:logistic" as objective function, which uses binary cross-entropy as loss function, defined as follows [18]:

$$l(y_i, \hat{y}_i) = -y_i \log(\hat{y}_i) - (1 - y_i)\log(1 - \hat{y}_i) \qquad (7)$$

Then, XGBoost calculates the first and the second order gradients as follows:

$$\begin{aligned} g_i &= \sigma(\hat{y}_i) - y_i \\ h_i &= \max\left(\sigma(\hat{y}_i)\left(1 - \sigma(\hat{y}_i)\right), \varepsilon\right) \end{aligned} \qquad (8)$$

Where $\sigma(\hat{y}_i) = \frac{1}{1+e^{-\hat{y}_i}}$ denotes as sigmoid function of $\hat{y}_i$, and $\varepsilon = 10^{-16}$. ε term is added to the Eq. (**8**) in order to ensure that the predicted value is higher than ε.

And value of λ is set to 1, γ is set to 0 by default, so that Eq. (3) becomes:

$$\Omega(\mathrm{f}) = \frac{1}{2} \sum_{t=1}^{T} w^2 \qquad (9)$$

Since the major problem of the dataset is the data imbalance, the number of non-fraudulent financial statements are greater than fraudulent financial statements. XGBoost requires one more parameter called scale_pos_weight, this is scale weight of fraudulent financial statements in training dataset. This parameter reduces loss function values by adding weight to the gradient (first order derivate) and hessian (second order derivate). Usually, it would be the ratio of number of the non-fraudulent financial statements over the fraudulent financial statements. During the research, we set the value of scale_pos_weight is 250, and the observed results showed better.

To avoid overfitting, maximum iteration is set to 5000, training dataset also involved in validation dataset. During the training, the metric was use in training evaluation is Area Under the Receiver Operating Characteristics Curve (AUC), and early stopping round is to 10. This means if value of AUC does not increase in 10 iteration rounds, model would stop training.

Learning rate of XGBoost takes a role in training part. The learning rate adds weights by factor η after each step of tree boosting. This technique is used to avoid the overfitting issue. The value of this parameter is between 0 and 1. It is recommended that this parameter should less than or equal 0.1 [19]. The proposed model uses 0.05 as learning rate value, because it would maximize the NDCG@k metric, which is described in section 5.2.

The output of this model is probability of frauds and non-frauds, this research only focuses on fraudulent probability. If a set of financial statements has predicted value is greater than or equal 0.5, it is considered as fraudulent, otherwise it is non-fraudulent.

Table 3. Averaged of performance metrics over the test period 2003-2008 with 28 raw data items.

Metric / Method	AUC	NDCG@k	Sensitivity	Precision
SVM-FK	0.626	0.020	2.53%	1.92%
Logit	0.690	0.006	0.73%	0.85%
RUSBoost	0.725	0.049	4.88%	4.48%
XGBoost	0.689	0.047	3.56%	3.36%
f-XGBoost	0.693	0.054	5.00%	4.22%

5. Evaluation metrics.

There are several metrics to evaluate the performance of this classification model. A standard approach is to use the Accuracy metric, but it is not suitable for imbalanced dataset. For example, if our model predicts all financial statements are non-fraudulent, then value of this metric is about 98%, which shows that the model has a high performance. Another method is to use k-fold validation because the fraud data has time property, and performing this validation is inappropriate [4]. Particularly, k-fold cross validation is a procedure that splits the training dataset into k folds (or groups), then takes a group for testing and remains groups are used for training, repeats steps k times. It would issue testing year occurs before the training period. For example, if the period is used for training and testing is 1991-2003, the performing cross-validation would take 1991 for testing, and 1992-2003 for training in the first iteration, inappropriately. An alternative method to measure the performance of this classification method is Area Under Receiver Operating Characteristic (ROC) Curve (AUC). Furthermore, the fraudulent prediction task can be thought as a ranking problem. Specifically, we can limit the evaluation to only a small number of financial statements with the highest predicted probability of fraud [4], so the metric is used in this one is NDCG@k. This metric is widely used for evaluating ranking algorithms such as search engine and recommendation algorithms [20].

5.1. Area Under a Receiver Operator Characteristic Curve.

Receiver Operator Characteristic (ROC) curve is described as a two-dimensional depiction of classifier's performance that combined true positive rate and false positive rate in a graph. To measure the performance of a classifier, a common method is to calculate the Area Under the ROC Curve (AUC). AUC is a portion of the area of the unit square, and its value will fall within range between 0 and 1.0. Because random guessing produces the diagonal line between (0,0) and (1,1) which has area of 0.5, no realistic classifier should have an AUC less than 0.5 [21]. The AUC is equivalent to probability that a randomly chosen positive instance (i.e., a true fraud) will be ranked higher by a classifier than a randomly chosen negative instance (nonfraud) [4].

This metric is used instead of Balanced Accuracy (BAC) metric. Balanced Accuracy is widely used to perform the evaluation, which is defined as the average $Sensitivity = \frac{TP}{TP+FN}$ and $Precision = \frac{TP}{TP+FP}$ where TP is number of observations that is correctly classified as fraudulent, FP is number of observations misclassified as fraudulent. FN are fraudulent observations that misclassified as non-fraudulent. But David and Zakolyukina [22] pointed out that this metric has two limitations. Firstly, BAC is based on specific predicted fraud probability threshold of a given classifier. A different threshold will result give different BAC value. Therefore, if auditor has no knowledge about the cost of misclassifying false positive and the false negative, they could not determine the optimal threshold. Secondly, BAC depends on Sensitivity, which is sensitive to the relative frequency of positive and negative instances in the sample (i.e., imbalanced data).

5.2. NDCG@k

Normalized discounted cumulative gain at position k (NDCG@k) is a theory in evaluation of search engines result. It is the normalization of discounted cumulative gain at position k (DCG@k), which is defined as following formula:

$$DGC@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)}$$

Where rel_i equals 1 if financial statement at i^{th} is considered as fraud, and 0 otherwise. k is the number of financial statements in test period that have the higher probability of fraudulent financial statements. Value of k is set 1% of test firms-year because the average frequency of accounting frauds detected by SEC's AAERs are typically less than 1% in a year. On the other hand, due to imbalanced dataset and the avoiding of investigating cost of false positive, value of k is set to 1% of test financial statements. For example, Cecchini et al. [7] reported that SVM FK correctly

Table 4. Averaged performance metrics over the test period 2003 - 2008 by combine multiple input types.

Input Variable	Method	Metric			
		AUC	NDCG@k	Sensitivity	Precision
28 raw financial data	RUSBoost	0.725	0.049	4.88%	4.48%
	f-XGBoost	0.693	0.054	5.00%	4.22%
14 financial ratios	RUSBoost	0.659	0.017	2.03%	1.69%
	f-XGBoost	0.607	0.030	3.26%	2.27%
28 raw financial data + 14 financial ratios	RUSBoost	0.696	0.035	3.19%	2.54%
	f-XGBoost	0.672	0.035	3.10%	3.08%

classified 80% fraudulent financial statements, and 90.6% non-fraudulent. However, Bao et al. [4] reported that SVM-FK resulted too many false positive within test period 2003-2008 in their dataset, specially SVM-FK mislabeled 2,881 non-fraudulent observations as frauds [4]. Obviously, auditors have to pay a high cost if they want to investigate all predicted fraud firms.

DCG@k relies on two keys: First one is a fraudulent observation has higher probability than a non-fraudulent observation, and second one is a fraudulent observation has a higher score if it is ranked higher in ranking list. This means that a higher ranked observation will be weighted more highly by position discount, that is denoted by $\log_2(i+1)$.

$NDCG@k$ is $DCG@k$ normalized by the ideal $DCG@k$, which is denoted as below:

$$NDCG@k = \frac{DCG@k}{iDCG@k}$$

Where $iDCG@k$ is $DCG@k$ value when all true frauds are ranked at the top of ranking list. So, the values of NDCG@k are bounded between 0 and 1 and a higher value represents model has better ranking performance.

To illustrate the benefit of $NDCG@k$, this paper also presents the performance of top 1% of Sensitivity:

$$Sensitivity = \frac{TP}{TP + FN}$$

where TP represents as a number of cases that are correctly predicted as fraudulent, and FN is a number of cases that are fraudulent firms but misclassification as non-fraudulent, sum of TP and FN is the observations that have highest predicted fraudulent probability. This metric shows how well the model correctly identifies fraudulent financial statements. And Precision:

$$Precision = \frac{TP}{TP + FP}$$

where FP represents non-fraudulent firms that are misclassified as fraudulent. Sum of TP and FP is number of observations that are classified as fraudulent financial statements in the top 1% of firms. This metric shows the ratio of correctly predicted fraudulent observations to the total predicted fraudulent observations, and the higher precision the lower number of non-fraudulent observations are misclassified as frauds.

6. Empirical result

6.1. Processing

After built-up the proposed model, this research continues with selecting data for training and testing the model. To have an objective comparison with Bao et al. work [4], this paper uses financial statements within 2003-2008 for testing, and training period contains financial statements from 1991 to test year with two years gap. For example, if test year is 2003, training period would be 1991-2001. To ensure the reliability, it is required that training period should be higher than 10 years [4]. Moreover, Bao et al. assumed that SEC would take about 24 months for the disclosures of fraudulent firms.

The determining training and testing period are suggested before training and testing the proposed model. Table 3 displays the model performance with other models. Specifically, the average AUC of the proposed models is 0.693, that is lower than RUSBoost model by Bao et al., but higher than the logistic of Dechow et al. (0.690), and Support Vector Machine with financial kernel of Cecchini et al. (0.626), which are already described in [4]. Furthermore, to evaluation our model, NDCG@k is used. In comparison with previous ones, our model gives an average value is 0.054, that is higher than the average value of Bao et al.'s model for top 1% of predicted fraudulent firms in test period 2003-2008. Particularly, XGBoost cannot detect any fraudulent cases, f-XGBoost model correctly predicted 23 fraudulent cases, while model of Bao et al. identified total 16 fraudulent cases, 9 fraudulent cases

Table 5. Averaged performance metrics over test period 2003-2008 ignore serial fraud.

Metric Method	AUC	NDCG@k	Sensitivity	Precision
SVM-FK	0.661	0.025	2.90%	2.24%
Logit	0.708	0.002	0.24%	0.28%
RUSBoost	0.801	0.158	13.56%	10.74%
XGBoost	0.700	0.028	2.79%	2.52%
f-XGBoost	0.777	0.089	8.12%	6.49%

Table 6. Averaged performance metric over test period 2003-2005 with 28 raw financial data items

Metric Method	AUC	NDCG@k	Sensitivity	Precision
SVM-FK	0.637	0.024	2.28%	2.53%
Logit	0.685	0.012	1.45%	1.69%
RUSBoost	0.753	0.085	7.64%	7.83%
f-XGBoost	0.691	0.079	6.59%	6.71%

Table 7. Averaged performance metric over test period 2003-2011 with 28 raw financial data items

Metric Method	AUC	NDCG@k	Sensitivity	Precision
SVM-FK	0.647	0.025	3.07%	1.98%
Logit	0.702	0.012	1.87%	1.19%
RUSBoost	0.710	0.040	4.40%	3.60%
f-XGBoost	0.678	0.040	3.69%	3.02%

Table 8. Averaged performance metric over test period 2003-2014 with 28 raw financial data items

Metric Method	AUC	NDCG@k	Sensitivity	Precision
SVM-FK	0.628	0.019	2.30%	1.48%
Logit	0.709	0.011	1.84%	1.04%
RUSBoost	0.717	0.030	3.30%	2.70%
f-XGBoost	0.678	0.030	2.77%	2.26%

for Dechow et al. model, and 7 fraudulent cases for Cecchini et al. model.

6.2. Combine raw data items and ratio-items.

In additionally, the experimental results not only examine on 28 raw data items alone, but also use 14 ratio-items and combine both 28 raw data items with 14 ratio-items. Table 4 reports the performance statistic of the results. By using 14 ratio-items, value of AUC for f-XGBoost model is 0.607, that is lower than RUSBoost of Bao et al.'s model., but the value of NDCG@k is 0.030, while Sensitivity of our model is 3.26% and Precision is 2.27%, higher than Bao et al.'s model. If we combine 28 raw data items with 14 ratio-items, the proposed model gives 0.632 for AUC and 0.035 for NDCG@k, while Sensitivity is 3.10% and Precision is 3.08%, slightly outperforms Bao et al.'s model. This experiment shows that the results of both ratio-items and combinations of raw data items and ratio-items do not outperform the one based on 28 raw data items alone.

6.3. Serial fraud

As mentioned, financial frauds that span multiple year may impact the flexible and robustness of the proposed model, this research recodes all of fraudulent observations in training period as non-fraudulent if they span both the training and testing periods to prevent the overstating performance of the model. However, most of previous studies did not show the impact of these cases. Therefore, the research is also ignoring serial fraud for the test period 2003-2008. Table 5 shows that the performance slightly improves in comparison with the performance of the same model in Table 3. When ignoring serial fraud, the proposed method does not perform as well as RUSBoost model, but it works slightly better than SVM-FK and Logit models. Conversely, the proposed model performs faster than RUSBoost model because it uses GPU power.

6.4. Alternative test periods.

To show the robustness of the proposed model, the experiments are continually examined following periods as alternative test samples: 2003-2005, 2003-2011 and 2003-2014. The results are reported in Table 6 for test period year 2003-2005, Table 7 for period 2003-2011, and Table 8 for period 2003-2014. We could easily see that the longer of the test period, the less reliable of the model performance. Because it is reasonable to assume that the undetected frauds grow over time [4]. For the comparison with state-of-the-art models, the first period our model performs not so well when compare with Bao et al.'s model, but higher than Dechow et al.'s model and Cecchini et al.'s model. The reason could be the predicted probability of False Positive is higher than the predicted probability of True Positive. It leads to some true fraudulent firms stay outside of top 1%. When we do not use cut-off 1%, the proposed model correctly predicts 15 fraudulent cases in period 2003-2005. And the second period, 2003-2011, the model predicts 24 fraudulent cases without cut-off 1%. For the last period, the results correctly predict 28 cases without cut-off 1%.

7. Conclusion

This paper provides a machine learning method that enables to detect accounting frauds based on a dataset of publicly traded U.S. firms over the period 1991-2008. The period 2003-2008 are used as the out-of-sample test period and the years from 1991 to test year as the training period. The other periods, 2003-2005, 2003-2011, 2003-2014 are used for alternative testing. We required a gap 24 months between last year of training period and testing year because Dyck et al. [2] proved that it would take about 24 months for fraudulent financial statements to be disclosed.

In the comparison with existing studies, the available dataset of financial statements is used as input of the model. F-XGBoost model is implemented by using the XGBoost algorithm, the ensemble learning method, and a state-of-the-art paradigm. To evaluate the performance of proposed model, we used two metrics AUC and NDCG@k with k is top 1%.

The used research dataset was provided by Bao et al. [4]. It contains 28 raw data items and 14 ratio-items based on Cecchini et al. and Dechow et al. The model is mainly used raw financial statements while doing experiments rather than ratio-items. Because there is a finding that the proposed model worked on raw data items better than ratio-items. In particular, value of NDCG@k at top 1% is 0.054, and it predicts about 24 fraudulent cases within period 2003-2008.

In future, the further research could enhance the proposed model by trying with some other dataset such as like non-financial items to improve the model's accuracy.

8. References

[1] L. Shkulipa, "Conceptual Framework for Financial Reporting," *SSRN Electronic Journal,* p. 8, 2018.

[2] A. Dyck, A. Morse and L. Zingales, "Who Blows the Whistle on Corporate Fraud," *The Journal of Finance,* vol. LXV, no. 6, pp. 2213-2253, 2010.

[3] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 2016.

[4] Y. Bao, B. Ke and Y. J. Yu, "Detecting Accounting Fraud in Publicly Traded U.S. Firms Using a Machine Learning Approach," *Journal of Accounting Research,* pp. 199-235, 2020.

[5] B. P. Green and J. H. Choi, "Assessing the Risk of Management Fraud Through Neural Network Technology," *Auditing A Journal of Practice & Theory* , vol. 16, no. 1, 1997.

[6] P. M. Dechow, W. Ge, C. R. Larson and R. G. Sloan, "Predicting Material Accounting Misstatements," *Contemporary Accounting Research,* vol. 28, no. 1, pp. 17-82, 2011.

[7] M. Cecchini, H. Aytug, G. J. Koehler and P. Pathak, "Detecting Management Fraud in Public Companies," *MANAGEMENT SCIENCE,* vol. 56, no. 7, pp. 1146-1160, 2010.

[8] S. Chen, "Detection of fraudulent financial statements using the hybrid data mining approach," in *SpringerPlus 5,* SpringerOpen, 2016, pp. 1-16.

[9] B. Jeremy, C. Edwige, F. Eric and P. Wenqiang, "Using machine learning to detect misstatements," *Review of Accounting Studies,* 2020.

[10] J. M. Karpoff, A. Koester, D. S. Lee and G. S. Martin, "Proxies and Databases in Financial Misconduct Research," *Accounting Review,* no. 92, 2017.

[11] M. D. Beneish, "The detection of earnings manipulation," *Financial Analysts Journal,* vol. 55, no. 5, pp. 24-36, 1999.

[12] S. L. Summers and J. T. Sweeney, "Fraudulently misstated financial statements and insider trading: An empirical analysis," *Accounting Review,* pp. 131-146, 1998.

[13] M. Re and G. Valentini, Ensemble Methods: A review, ResearchGate, 2012.

[14] S. Sun, Y. Wei and S. Wang, "AdaBoost-LSTM Ensemble Learning for Financial Time Series Forecasting," *Computational Science – ICCS 2018,* vol. 10862, pp. 590-597, 2018.

[15] H. Faris, R. Abukhurma, W. Almanaseer, M. Saadeh and A. M. Mora, "Improving financial bankruptcy

prediction in a highly imbalanced class distribution using oversampling and ensemble learning: a case from the Spanish market," *Progress in Artificial Intelligence*, vol. 9, p. 31–53, 2019.

[16] H. Choi, H. Son and C. Kim, "Predicting Financial Distress of Contractors in the Construction," *Expert Systems With Applications*, vol. 110, pp. 1-10, 2018.

[17] Z.-H. Zhou, Ensemble Methods: Foundations and Algorithms, Boca Raton: CRC Press, 2012.

[18] E. Moshe, "The loss function and evaluation metric of XGBoost," 29 November 2018. [Online]. Available: https://stackoverflow.com/a/53535742. [Accessed 2 September 2021].

[19] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367-378, 2002.

[20] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422-446, 2002 .

[21] T. Fawcett, "An introduction to ROC analysis," in *Pattern Recognition Letters 27*, Palo Alto, 2006.

[22] L. F. David and A. A. Zakolyukina, "Detecting Deceptive Discussions in," *Journal of Accounting Research*, vol. 50, no. 2, pp. 495-540, 2012.