TEAM ID : NM2023TMID06781

PROJECT NAME : A MACHINE LEARNING FOR DETECTING FRAUD IN AUDIT DATA

A Machine Learning Approach to Detect Accounting Frauds [§]

Keywords: machine learning; logistic regressions; accounting irregularities; AAERs

JEL Classification: C44; C50; C53; M41

[§] We thank, for very their very helpful comments, Jeremy Bertomeu, Shuyu Li (discussant) and participants in the 2022 FMARC conference. All computer programs used to generate the results presented in this article are available through the GitHub repository at https://github.com/hkalager/ML_AccountingFraud [†] Corresponding author, email: pp537@bath.ac.uk

A Machine Learning Approach to Detect Accounting Frauds

Abstract

This paper introduces a new fraud detection model to the accounting literature using machine learning (ML). This model, which we refer to as LogitBoost, applies ensemble learning to logistic regressions. We show, using seven alternative measures assessing the ability to detect fraud, that our model outperforms the methods based solely on logistic regressions or other ML methods used by prior literature. Additionally, our model outperforms the others in predicting fraud beyond the current accounting period. Importantly, our method relies on a lower number of predictors than those used in prior ML research, thus minimizing concerns over multicollinearity and potential overfitting associated with machine learning methods.

1. Introduction

This paper introduces a new fraud detection model using machine learning (ML) methods, which we refer to as LogitBoost, to the accounting literature. The model incorporates both the traditional logistic regression approach and more recently introduced ML methods in accounting research. We show, using seven alternative performance measures, that our model outperforms methods based solely on logistic regressions or other ML methods used in prior literature.

Research in accounting has used a wide range of techniques to develop fraud detection models. Within this literature, the most widely used method is the logistic regression (Logit) model (Beneish, 1997, 1999; Summers & Sweeney, 1998; Dechow et al., 2011). Among the studies using logistic regressions, the specification proposed by Dechow et al. (2011) is arguably regarded the most influential one. Recent contributions, which use ML methods to detect accounting fraud, include Cecchini et al. (2010), Perols (2011), Abbasi et al. (2012), Perols et al. (2017), Dutta et al. (2017), Bao et al. (2020) and Bertomeu et al. (2021). The two studies most closely related to our paper are Cecchini et al. (2010) and Bao et al. (2020). Cecchini et al. (2010) introduce a novel kernel for support vector machine (SVM) that maps raw accounting data into a set of financial ratios. Bao et al. (2020) use random undersampling boosting (RUSBoost), which is a type of ensemble learning based on decision trees. We build on this strand of literature and propose a fraud detection model that applies ensemble learning to logistic regressions (LogitBoost).

Friedman et al. (2000) introduce LogitBoost and argue that ensemble learning algorithms (AdaBoost in particular) can be used as stagewise estimation procedures for fitting a Logit model. This enables the optimization of the binomial log-likelihood when using ensemble learning algorithms and thus leads to a superior performance when applied to settings with discrete outcomes. In other words, the method uses a binomial log-likelihood loss function changing linearly with the classification error. This approach is less sensitive to noise, bias and

outliers and it does not require a balanced number of firms. While its superiority has been shown in a number of settings such as medical science (Cai et al., 2006; Zhang & Fang, 2007), computer science (Lutz, 2006; Otero & Sánchez, 2006) and landslide prediction (Pham & Prakash, 2019), this is the first study to introduce LogitBoost in an accounting setting.

Our review of the literature on fraud detection models based on ML methods highlights three key methodological concerns in the application of ML techniques which could introduce bias and thus lead to incorrect conclusions. The research design which we use is aimed at minimizing the biases related to these issues. We expand on these concerns below and address them in our application.

First, due to the mechanics of the accounting processes, many variables used as predictors in ML models are likely to be highly correlated. Using of a high number of predictors may lead to collinearity and subsequently overfitting. The number of predictors used by prior ML literature ranges from 23 in Cecchini et al. (2010) to 116 in Dutta et al. (2017). Irrespective of the level of multicollinearity, machine learning algorithms will complete and learning will take place, unlike methods based on regressions (Bertomeu, 2020). This could, however, increase the chance of overfitting because of the co-dependence among predictors. To implement LogitBoost, we use the set of 11 financial ratios proposed by Dechow et al. (2011). As we show in Section 2.2, this set of variables minimizes multicollinearity issues when compared to other predictors used by prior literature (e.g., Bao et al., 2020; Cecchini et al., 2010).

Second, many fraud cases span over multiple accounting periods, which adds further complexity in developing a detection model. This is because some information is used for processing the training samples which is not available at the point of time in which training samples are observed. In our dataset, more than 50% of the cases span over more than one annual statement. Overlooking serial frauds, as done by some earlier studies (e.g., Cecchini et

al., 2010; Perols et al., 2017), can induce bias and lead to misleading conclusions. Bao et al. (2020) show that the performance of ML techniques, and, in particular, ensemble learning, can be inflated by up to three times when not adjusting for such "serial frauds".[1] A bias-free approach to treat serial cases is conceptualized in Bertomeu (2020), who argues that test samples should be selected before any assessments of performance are known. Drawing upon Bertomeu (2020), we adjust for serials frauds by discarding the observations for the firms with a misstatement from the testing sample, prior to any model training.

Third, machine learning methods involve splitting the data population into training, validation, and testing sub-samples. Typically, the split is done chronologically with the sample used for validation being restricted to a few pre-specified periods. Therefore, the performance of the model may be biased by the contextual events of the validation sample which are not generalizable to subsequent periods. To minimize the potential bias, Bertomeu (2020) recommends the use of k-fold cross-validation, which is widely used in computer science and has been applied recently in accounting research (Ding et al., 2020). With this approach, model performance is measured over k iterations involving a variation of sub-samples assigned for validation and the overall performance is measured as the average of all assignments. Further, splitting the sample chronologically is an imperfect solution to the panel structure of accounting data as it ignores within-firm correlations which could result in incorrectly reporting high model performance in the validation sample (Bertomeu, 2020).

In order to assess the effectiveness of our model, we use seven detection models as benchmarks. First, we present the results obtained using logistic regressions (Logit) as in Dechow et al. (2011). Second, following Cecchini et al. (2010), we use SVM with a financial

[1] This finding, although not explicitly discussed in Bao et al. (2020), can be inferred when comparing the results shown in Table 3 and Table 6. Specifically, sensitivity and precision, two performance metrics, are almost three times higher for RUSBoost when serial frauds are ignored. In contrast, this is not the case for simple logistic models predicting fraud.

kernel (SVM-FK). Third, we use the RUSBoost method, which was introduced to accounting research by Bao et al. (2020). Drawing from Perols (2011), we then use an SVM with linear

kernel (SVM) and multilayer perception (MLP), which belongs to the category of algorithms referred to as artificial neural network. Next, we consider two methods which, to our knowledge, have not been used in accounting research: stochastic gradient descent (SGD) classifier, and a combination of the alternative models which we call fused ML (FML). We note that SGD is a generalized linear classifier like SVM. To estimate SVM-FK and RUSBoost, we use, as predictors, the same variables used by Cecchini et al. (2010) and Bao et al. (2020), respectively, because we are interested in a direct comparison with these studies. To estimate all the other models, we use the 11 predictors in Dechow et al. (2011).

To compare the performance of the models, we use seven separate performance metrics. First, we use the probability that a fraud observation is correctly classified as fraud (true positive), which is known as sensitivity (or recall). Additionally, we employ the probability that a nonfraud observation is correctly classified as nonfraud (true negative), which is referred to as specificity. Third, we use the ratio of the number of the true positives to the total number of positives, which we refer to as precision. Following Bertomeu et al. (2021), we then present the F1 score, defined as the harmonic mean of sensitivity and precision. Fifth, we use the area under the receiver operating characteristic curve (AUC), following Larcker and Zakolyukina (2012). Sixth, following Bao et al. (2020), we use the normalized discounted cumulative gain at the position k (NDCG@k). Finally, following Perols (2011) and Perols et al. (2017), we employ the expected cost of misclassification (ECM). Following Bao et al. (2020), we compute all these metrics, except the AUC, focusing on the top percentile of the distribution of the firmyear observations, based on the likelihood to be classified as fraudulent.

The results show that the proposed LogitBoost performs substantially better in detecting fraud than any of the methods used in prior literature that we employ as benchmarks. This result holds across all the seven performance metrics we use. In particular, the value of sensitivity, that is the true positive rate, using our method, LogitBoost, is higher than that of the SVM-FK (Cecchini et al., 2010) and of the RUSBoost (Bao et al., 2020) methods by over 100%. Thus, our proposed methodology has higher sensitivity, i.e., effectiveness in identifying fraud firms, than the existing methods using machine learning techniques, despite using a lower number of predictors than those used in these studies. Additionally, our results indicate that some of the main findings in prior literature—in particular, RUSBoost (Bao et al., 2020) and SVM-FK (Cecchini et al., 2010) outperforming Logit—do not hold in our setting; this suggests that the methodological issues we identified in prior literature can affect the conclusions drawn from machine learning methods.

In a further set of analyses, we compare the ability of the eight fraud detection models to predict fraud cases in future accounting periods. Specifically, we investigate the likelihood that a firm which is predicted to be fraudulent by an ML method in the current accounting period is actually detected for fraud by the Securities and Exchange Commission (SEC) at any subsequent accounting period. These results can be relevant to the SEC in improving their ability to detect fraud given that it can take up to several years to conclude their investigations. The results show that LogitBoost outperforms the other methods in predicting fraud in subsequent periods. In addition, our method is the only one which is able to predict fraud, on average, more than a year ahead of SEC investigations.

Finally, we perform a set of tests to examine the effects of the three key methodological issues we identify in prior literature. Specifically, using a Monte Carlo simulation, we show that collinearity among the predictors strongly decreases the reliability of the estimates obtained from the fraud detection models. In addition, we find that ignoring serial fraud and taking a chronological approach to the validation of the hyperparameters strongly affect the estimates

of the ML methods we examine. This further confirms the importance of these issues in interpreting of implementing a fraud detection model.

Our study contributes to the small but growing literature on detecting fraud using ML methods (Cecchini et al., 2010; Dechow et al., 2011; Perols, 2011; Perols et al., 2017; Bao et al., 2020; Bertomeu et al., 2021). In general, our paper responds to the calls by Bertomeu (2020) and Krupa and Minutti-Meza (2022) for more research in accounting and ML methods, a research area which is still underdeveloped. More specifically, we provide the following two contributions to the literature.

First, we show that a machine learning method that applies ensemble learning to logistic regressions outperforms models based on logistic regressions alone (Dechow et al., 2011), ensemble learning (Bao et al., 2020), artificial neutral networks (Perols, 2011), SVM (Perols, 2011) and SVM-FK (Cecchini et al., 2010) in detecting fraud. We also show that our model outperforms the others in predicting fraud beyond the current accounting period. Importantly, our model outperforms the methods in prior literature despite using a substantially lower number of predictors (Bao et al., 2020; Cecchini et al., 2010; Dutta et al., 2017; Perols, 2011).

Second, we address a set of methodological issues in prior literature. Specifically, our research design is aimed at minimizing concerns related to multicollinearity, serial frauds and the method used to validate the ML model. We show that the conclusions in Bao et al. (2020), Cecchini et al. (2010) do not hold in our setting; in particular, we find that the SVM-FK and RUSBoost methods do not outperform logistic regressions. We also document that the set of predictors used by prior research (in particular, Cecchini et al., 2010 and Bao et al., 2020) are highly collinear. These results indicate that some of the main findings in prior fraud detection literature based on ML are sensitive to the three key methodological concerns we identified.

Our findings have important implications for auditors and regulators. Auditors could use the insights presented in this article to assess the potential fraud risk. Our results inform the regulators such as the SEC by providing a new fraud detection tool. Given the limited resources

available for a full investigation, such tool based on publicly available data could allow them to better prioritize their investigatory efforts. Recent fraud cases by firms in the US such as Kraft Heinz, Granite Construction and CHS or internationally Carillion in the UK have resulted in significant losses for shareholders. Thus, fraud prediction models could be highly relevant for investors to make informed decisions.

The remainder of the paper is structured as follows. Section 2 reviews the relevant literature and discusses the methodological concerns in prior literature. Section 3 discusses our sample while Section 4 presents our methodology. Sections 5 and 6 presents our main findings and additional analyses, respectively. Section 7 concludes.

2. Prior literature and relevant methodological issues

ML uses computational techniques to automate the discovery of patterns that may be difficult to find otherwise. Cho et al. (2020) define ML as "computer programs that automatically and iteratively learn patterns from historical data to make predictions about future data". Prior studies have applied ML to a number of accounting topics, which include the detection of accounting misreporting, audit opinions, earnings or components of earnings, as well as bankruptcy.[2] Our paper is primarily related to the prior studies which use ML techniques applied to quantitative financial statement data to detect misreporting.

The majority of prior papers in the accounting literature rely on logistic regressions to detect misreporting (Dechow et al., 1996; Beneish, 1997; 1999; Summers & Sweeney, 1998; Dechow et al., 2011). Arguably, one of the most influential papers in this stream of research is the study by Dechow et al. (2011) developing a logistic model that consists of 11 financial ratios. It is worth noting that previous studies using logistic regressions aim to explain fraud within sample and by causal inference. In contrast, our aim is to detect fraud out of sample. As

[2] See Krupa and Minutti-Meza (2022) for comments on other papers in accounting research which use ML techniques.

Bao et al. (2020, p. 201) notes, “the objective of causal inference modelling is to minimize the bias resulting from model misspecification to obtain the most accurate representation of the underlying theory. In contrast, the objective of predictive modelling seeks to minimize out-ofsample prediction error”.

More recent literature relies on more advanced, and potentially more effective methods, which are based on ML, to detect accounting irregularities. Two papers have been highly influential within the accounting literature and are most closely related to our research. Cecchini et al. (2010) develop a fraud detection model based on support vector machines (SVM) with a financial kernel (SVM-FK) that maps raw financial data into a broader set of ratios within the same year and changes in ratios across different years. They find that their method outperforms the traditional fraud detection models, including the Dechow et al. (2011)’s model. Bao et al. (2020) use a class of ensemble learning methods, RUSBoost, to develop a fraud detection model. They revisit the datasets in Cecchini et al. (2010) and Dechow et al. (2011) and argue that converting raw accounting data into a limited number of financial ratios based on potentially incomplete behavioral theories could lead to the loss of useful predictive information. Accordingly, they use raw financial data and show that their model (RUSBoost) has stronger detection ability than previously used methods.

Other contributions, which are strictly related to this study, use other ML methods. Perols (2011)’s analysis is based on decision trees, SVM, and multilayer perception approach (MLP). Perols et al. (2017) introduce three new approaches: Multi-Subset Observation Undersampling (OU), Multi-Subset Variable Undersampling (VU), VU partitioned by type of fraud (PVU). The papers by Abbasi et al. (2012) and Dutta et al. (2017) are published outside of the accounting literature. Abbasi et al. (2012) introduce a new method, based on adaptive learning, which they call MetaFraud. Dutta et al. (2017) use a set of ML methods which include

naïve Bayes, SVM, and Bayesian belief network. Bertomeu et al. (2021) find merits of using Random Forest and RUSBoost when a small proportion of total firms—those firms which are most likely to associated with fraud based on the models—are inspected.

Although an increasing number of papers apply machine learning techniques in accounting, the area is still underdeveloped, with the recent studies by Bertomeu (2020) and Krupa and Minutti-Meza (2022) calling for more research. Reflecting on these commentary studies and the application of ML techniques by prior literature detecting fraud, we identify three key methodological issues which introduce bias. First, the high number of predictors used in the vast majority of prior studies may lead to collinearity and overfitting. The number of predictors used in the literature we review ranges from 23 to 116. Specifically, the number of predictors is: 109 in Perols et al. (2017), 102 in Bertomeu et al. (2021), 42 in Perols (2011), 28 in Bao et al. (2020), 23 in Cecchini et al. (2010), 48 in Abbasi et al. (2012) (core features at quarterly frequency) and 116 in Dutta et al. (2017) (at the initial stage and 15 after removing less significant predictors). In Section 2.2, we provide evidence of strong collinearity among some of the predictors used in prior research. This co-dependence between predictors could, however, increase the chance of overfitting and thus induce bias in the performance evaluation metrics used to assess the effectiveness of the model. Importantly, we show that our set of predictors minimizes collinearity issues. Second, all prior studies, with the exception of Bao et al. (2020) and Bertomeu et al. (2021), do not make adjustments for serial fraud. Overlooking serial fraud when using ML methods can be misleading and result to a substantial overstatement of the predictive ability of the models, as empirically shown by Bao et al. (2020) (see also footnote 1). We discuss how we treat serial fraud cases in Section 4.2. Third, machine learning methods require the identification of a sub-sample to validate the hyperparameters. Bertomeu (2020) suggests the use of cross-validation as opposed to splitting the sample chronologically, that some studies have used (e.g., Bao et al., 2020), which could lead to nongeneralizable performance. We discuss this issue further in Section 4.2

3. Sample and data

We obtain financial data for the U.S. non-financial firms in Compustat for the period 19912018. We require firm-year observations to have available data to calculate the 11 financial ratios in Dechow et al. (2011)'s model which we use as predictors in our main analysis. In addition, for the purpose of comparison with prior research, we require the availability of the variables used as predictors by Cecchini et al. (2010) and Bao et al. (2020). Further, we collect information about fraud from the SEC's Accounting and Auditing Enforcement Releases (AAERs). This dataset was developed in Dechow et al. (2011) using SEC filings. The final sample consists of 96,028 firm-year observations and 702 misstatement events.[3]

3.1 AAER cases over time

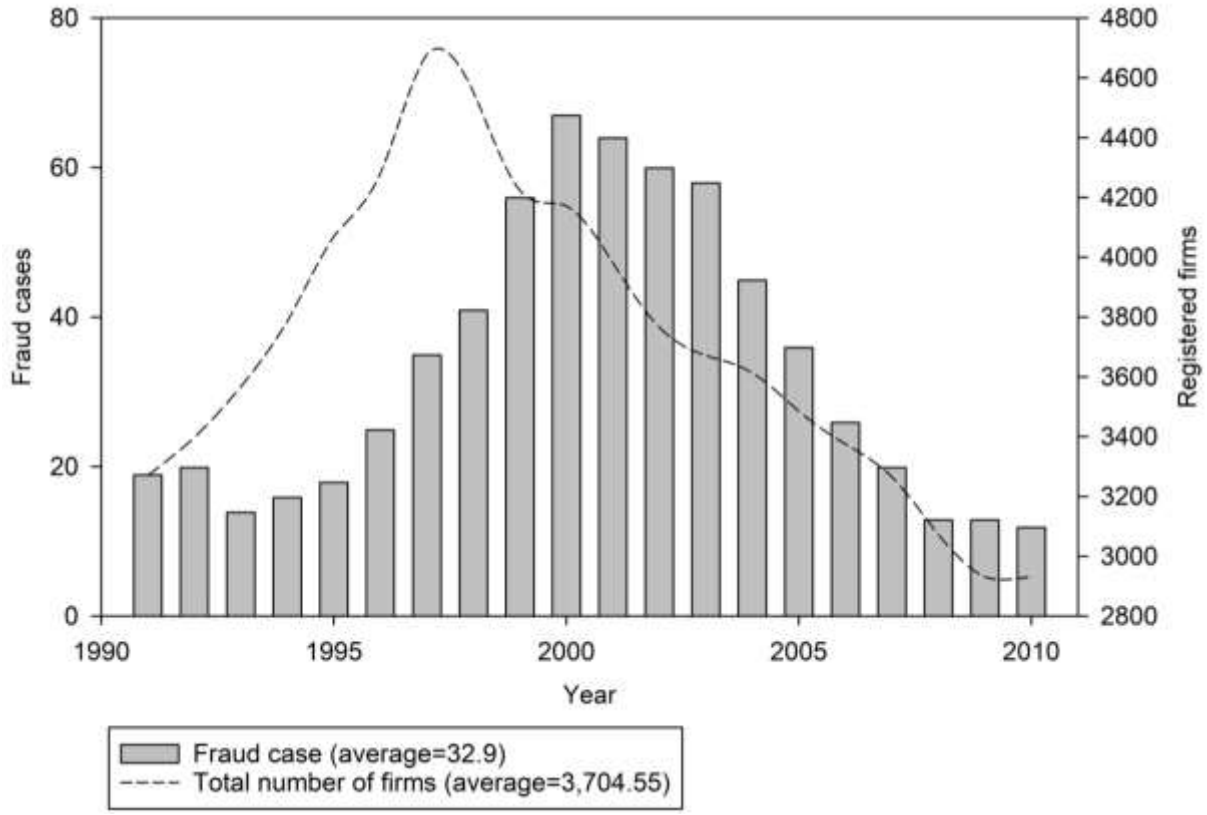
The number of AAERs over each calendar year varies with the number of random investigations and the number of active companies. In Figure 1, we present the AAER cases reported against the number of unique firms for each calendar year.

We note that, with the rise in the number of registered firms (in Compustat) in the 90s, the number of AAER cases rises. This rise in the number of registered firms and AAER cases is followed by a fall in early 2000s. We take advantage of the rise of AAERs in the 1990s and use this period as the main source of training observations, similar to Cecchini et al. (2010) and

[3] We use this dataset to optimize ML hyperparameters (by using firm-year observations in the period 1991-2000), to train and test ML models (using observations from 1991 to 2010), and to evaluate forward looking performance of ML models (using observations from 2002 to 2018). This choice is motivated by the observation that many AAER cases materialize several years after a misstatement is initially reported. Our choice of samples assumes that all misstatements for statements up to financial year 2010 are exhaustively identified and using those observations provide a reliable account of alternative ML models accuracy in detecting misstatements.

Bao et al. (2020). The proportion of the AAERs relative to the number of registered firms remains around 1% throughout the study period.

Figure 1. The number of AAER cases and unique firms over time.

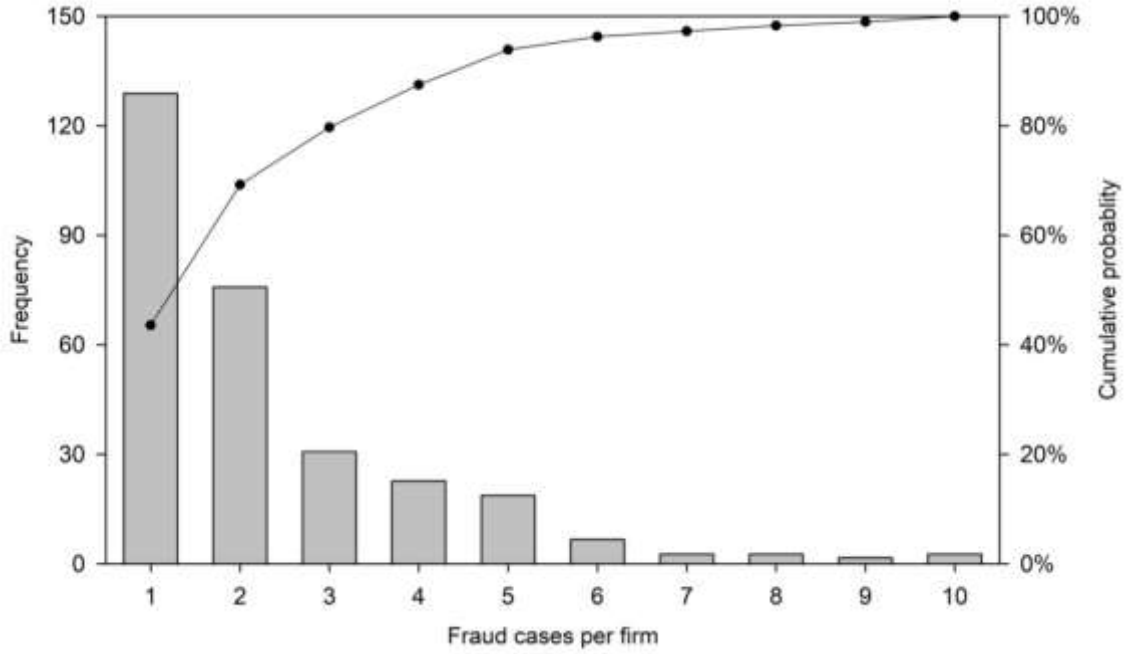


Notes: The figure presents the number of AAER cases and the number of firms in each calendar year (right) for the main study period (1990-2010).

We also note from Figure 1 that there is a lag between an increase in the number of firms and an increase in the number of AAERs. For instance, while the number of registered firms peaks in 1997, the number of AAERs peaks in 2000. This lagged behaviour can be explained through the operational difficulties of SEC investigations and considering that SEC

often tracks back into previous financial statements of those firms implicated in an AAER. The review of previous statements and the detection of more misstatements for the same company over multiple accounting periods is referred to as "serial frauds" (see among others, Bao et al., 2020 and Bertomeu et al., 2021). Figure 2 presents a histogram for the frequency of AAERs reported per firm.

Figure 2. Histogram of firms with at least one AAER case.



Notes: The figure presents, for the firms with at least one AAER in our sample, a histogram describing the number of AAER per firm.

Figure 2 shows that more than 50% of AAERs span over more than one annual accounting period and are classified as serial frauds. The serial frauds add further complexity to developing a fraud detection model since those statements were not initially classified as positive for fraud. Overlooking serial frauds, as done by some earlier studies (e.g., Cecchini et al., 2010), can be misleading. Bao et al. (2020) reports that the sensitivity of ML techniques can be up to three times when not adjusting for serial frauds. Section 4.2 discusses our methodology which aims to reduce the potential bias.

3.2 Variables used as fraud predictors

3.2.1 Variables used by prior research

Choosing the right set of explanatory variables is central to the success of ML models in pattern recognition. Previous literature on fraud detection employs a wide range of variables which could be associated with fraud, ranging from 11 in Dechow et al. (2011) to 116 in Dutta et al. (2017). Yet, there is no agreed set of variables to predict fraud cases. In our model, we use the 11 financial ratios employed by Dechow et al. (2011) to predict AAERs. We choose this set of variables because, as documented in the next subsection, they exhibit little evidence of multicollinearity. Table 1 presents the summary statistics for the financial ratios which are the main predictors used in our analysis. Appendix A presents the definition of these variables.

Table 1. Financial ratios used as fraud predictors in Dechow et al. (2011)

Variable	*Minimum*	*Maximum*	*Mean*	*Std. dev.*
WC accruals	-1.2591	0.7832	-0.0003	0.1437
RSST accruals	-2.8215	1.932	0.0148	0.3323
Change in receivables	-0.3602	0.3739	0.0122	0.0802
Change in inventory	-0.2789	0.2847	0.0067	0.0588
% soft assets	0.019	0.9715	0.5353	0.2501
Change in cash sales	-3.0525	11.0331	0.2263	0.9512
Change in cash margin	-16.6237	14.8529	-0.0519	1.8797
Change in return on assets	-2.6329	4.8188	-0.006	0.342
Change in free cash flows	-3.8077	3.7083	-0.0042	0.4637
Actual issuance	0	1	0.8876	0.3158
Book-to-market	-47.8635	10.1248	0.4289	2.2158

Notes: The table presents the variables used as AAER predictors in our main analysis. The variables are chosen following the main model in Dechow et al. (2011). All the variables are winsorized at the 1% level. See Appendix A for variable definitions. This table is based on 96,028 observations over the period 1991 to 2018.

3.2.2 Collinearity in prior research

A high level of collinearity signals strong linear dependence among the predictors. Machine learning algorithms will complete, and learning will take place, however, strong

multicollinearity among predictors would increase the chance of overfitting because of the codependence among predictors. A simple proxy for collinearity is the variance inflation factor (VIF), measured for each independent variable j as $VIF_j = \frac{1}{1-R_j^2}, j = 1,\dots, N$ where R_j^2 is the coefficient of determination of a regression with other variables and N is the total number of predictors (Gujarati, 2011). For example, a VIF of 5 indicates that 80% of the variance in the variable considered is explained by other predictors, thereby reducing the predictive power of that indicator.

Previous research has devoted limited attention to the statistical characteristics of the datasets used and, in particular, to the presence of correlation among the predictors. While the addition of an extra factor can improve the explanatory power of ML models, linear codependence between predictors increases the chance of overfitting.

Here, we focus on two of the most influential papers in prior research: Cecchini et al. (2010) and Bao et al. (2020). Cecchini et al. (2010) consider 23 raw (figures directly obtained from financial statements) variables and generate 1,518 features from the original set of inputs. Bao et al. (2020) expand the set of predictors used in Cecchini et al. (2010) to 28 variables. Table 2 lists the raw variables used in Cecchini et al. (2010) and Bao et al. (2020) along with the descriptive statistics of these variables. Next, Table 3 presents the VIF values for the variables used in Cecchini et al. (2010) and Bao et al. (2020). Further, the table shows the VIF values for the 11 ratios used in Dechow et al. (2011).

Table 3 shows that the sets of predictors used in Bao et al. (2020) and Cecchini et al. (2010) exhibit high levels of structural dependence (collinearity). Specifically, we note that the 13 out of 23 predictors in Cecchini et al. (2010) have strong linear dependence (VIF>10) with other predictors. Similarly, 18 out of 28 predictors used in Bao et al. (2020) exhibit strong dependence. Hence, many of the variables used in Cecchini et al. (2010) and in Bao et al. (2020) tend to co-move together. However, the financial ratios used in Dechow et al. (2011) show no

traces of collinearity. Based on the results in this section, we argue that, among the three sets of predictors considered, only the one using financial ratios is fit to be used for ML analysis given that co-dependence among predictors could increase the chance of overfitting.

In Section 6, we provide further evidence about the effects of collinearity among predictors.

Table 2. Raw variables used as fraud predictors in Cecchini et al. (2010) and Bao et al. (2020)

Variable	*Compustat code*	*In Bao et al. (2020)*	*In Cecchini et al. (2010)*	*Min.*	*Max*	*Mean*	*Std. dev.*
Current assets, total	ACT	Yes	No	0.05	19059.00	464.55	1414.58
Accounts payable	AP	Yes	No	0.03	4894.00	104.19	364.03
Assets, total	AT	Yes	Yes	0.18	58493.00	1487.72	4813.39
Common/ordinary equity, total	CEQ	Yes	Yes	-529.20	23046.00	525.78	1720.14
Cash and short-term investments	CHE	Yes	Yes	0.00	6369.00	116.25	400.70
Cost of goods sold	COGS	Yes	Yes	0.00	38609.00	872.38	2858.63
Common shares outstanding	CSHO	Yes	Yes	0.76	1975.05	60.42	158.12
Debt in current liabilities, total	DLC	Yes	Yes	0.00	2441.50	58.87	233.16
Long-term debt issuance	DLTIS	Yes	No	0.00	5385.04	115.89	421.63
Long-term debt, total	DLTT	Yes	Yes	0.00	15758.00	366.81	1270.97
Depreciation and amortization	DP	Yes	Yes	0.01	2482.00	62.87	198.51
Income before extraordinary items	IB	Yes	Yes	-2071.00	3651.00	56.09	291.45
Inventories, total	INVT	Yes	Yes	0.00	4966.00	116.23	375.05
Investment and advances, other	IVAO	Yes	Yes	0.00	3146.00	33.77	187.37
Short-term investments, total	IVST	Yes	Yes	0.00	1617.77	26.26	118.06
Current liabilities, total	LCT	Yes	Yes	0.17	13015.09	332.47	1100.88
Liabilities, total	LT	Yes	Yes	0.19	36773.00	932.21	3141.14
Net income(loss)	NI	Yes	Yes	-2071.90	3905.00	55.45	301.20
Property, plant and equipment, total	PPEGT	Yes	No	0.04	40255.79	970.45	3431.16
Price close, annual, fiscal	PRCC_F	Yes	Yes	0.01	125.19	14.95	17.36
Preferred/preference stock(capital), total	PSTK	Yes	Yes	0.00	390.94	5.68	32.45
Retained earnings	RE	Yes	Yes	-5116.42	17424.00	234.06	1243.26
Receivables , total	RECT	Yes	Yes	0.00	6250.00	163.27	513.11
Sales/turnover(net)	SALE	Yes	Yes	0.00	53674.00	1302.42	4130.72
Sale of common and preferred stock	SSTK	Yes	No	0.00	882.00	18.35	62.09

Income taxes, total	TXC	Yes	Yes	-93.00	1692.00	30.59	116.86
Income taxes payable	TXP	Yes	Yes	0.00	620.52	12.89	54.02
Interest and related expense, total	XINT	Yes	Yes	0.00	992.00	30.10	96.89

Notes: The table presents the raw financial data used by Cecchini et al. (2010) and Bao et al. (2020) in their fraud detection models. All the variables are winsorized at the 1% level. This table is based on 96,028 observations.

Table 3. Variance inflator factors (VIF) comparison across three main datasets

28 raw variables		*23 raw variables*		*11 ratios variables*	
Variables	*VIF*	*Variables*	*VIF*	*Variables*	*VIF*
Current assets, total	43.67	Assets, total	129.77	WC accruals	1.73
Accounts payable, trade	10.53	Common/ordinary equity, total	26.21	RSST accruals	2.21
Assets, total	133.27	Cash and short-term investments	4.85	Change in receivables	1.35
Common/ordinary equity, total	26.25	Cost of goods sold	25.85	Change in inventory	1.29
Cash and short-term investments	7.90	Common shares outstanding	3.28	% soft assets	1.02
Cost of goods sold	28.98	Debt in current liabilities, total	3.76	Change in cash sales	1.08
Common shares outstanding	3.32	Long-term debt, total	20.51	Change in cash margin	1.01
Debt in current liabilities, total	4.17	Depreciation and amortization	8.56	Change in return on assets	1.32
Long-term debt issuance	2.13	Income before extraordinary items	32.01	Change in free cash flows	1.65
Long-term debt, total	22.65	Inventories, total	4.09	Actual issuance	1.02
Depreciation and amortization	12.43	Investment and advances, other	1.79	Book-to-market	1.03
Income before extraordinary items	32.52	Short-term investments, total	2.26		
Inventories, total	6.18	Current liabilities, total	24.91		
Investment and advances, other	1.80	Liabilities, total	75.09		
Short-term investments, total	2.30	Net income(loss)	29.01		
Current liabilities, total	36.20	Preferred/preference stock(capital), total	1.21		
Liabilities, total	77.62	Retained earnings	3.91		
Net income(loss)	29.18	Receivables, total	7.05		
Property, plant and equipment, total	12.63	Sales/turnover(net)	49.45		

Preferred/preference stock(capital), total	1.23	Income taxes, total	5.35
Retained earnings	4.06	Income taxes payable	2.35
Receivables, total	11.19	Interest and related expense, total	18.54
Sales/turnover(net)	50.41	Price close, annual, fiscal	1.38
Sale of common and preferred stock	1.49		
Income taxes, total	5.38		
Income taxes payable	2.44		
Interest and related expense, total	18.77		
Price close, annual, fiscal	1.42		

Notes: The table presents the VIFs across three datasets considered in this study: the 28 raw variables dataset is used in Bao et al. (2020); the 23 raw variables dataset is used in Cecchini et al. (2010); the 11 ratios dataset is used in Dechow et al. (2011). The variables with VIF>10 are shaded in grey.

4. Methods

In Section 4.1, we discuss our model, which applies ensemble learning (similar to Bao et al., 2020) to commonly used logistic regressions (similar to Dechow et al., 2011). Additionally, we discuss the seven models that we use as benchmarks. In Section 4.2, we discuss how hyperparameters for each ML model are optimized through a cross-validation and present the optimal specifications. Finally, Section 4.3 discusses the performance metrics used to compare our model with the remaining seven.

4.1 Fraud detection models

4.1 LogitBoost

Our fraud detection model, which we refer to as LogitBoost, brings together ML methods (i.e. ensemble learning) recently introduced in the accounting literature with the commonly used logistic regressions (e.g., Dechow et al., 2011). LogitBoost was developed by Friedman et al. (2000), who argue that ensemble learning algorithms (as in Bao et al., 2020)

and, in particular, AdaBoost, can be used as stagewise estimation procedures for fitting a Logit model. This method uses a binomial log-likelihood loss function changing linearly with the classification error and it is less sensitive to noise, bias and outliers than other methods. In Appendix B, we also compare the performance of LogitBoost relative to using a decision tree, and AdaBoost in particular, as the base model for boosting; the results confirm the superiority of LogitBoost relative to AdaBoost. To estimate LogitBoost we use the 11 financial ratios in Dechow et al. (2011) as predictors because we show that this set of predictors minimizes collinearity concerns (Section 3.2).

4.2 Alternative fraud detection models

We compare the effectiveness of our model to detect accounting fraud against seven models. In the following we outline these models.

(1) *Logit* following Dechow et al. (2011). The Logit (logistic regressions) represents the most widely used method to predict accounting irregularities by prior literature. In our analysis, we follow Dechow et al. (2011) and use the same 11 financial ratios as predictors.[4]

(2) *SVM-FK* following Cecchini et al. (2010). SVM (support vector machine) refers to a category of algorithms which draw a boundary between observations of different groups. SVM methods have also been used by Perols (2011) to predict fraud and by Dutta et al. (2017) to predict restatements. The financial kernel maps raw accounting data into financial ratios. This model uses the same 23 raw variables used in Cecchini et al. (2010).

(3) *RUSBoost* following Bao et al. (2020). RUSBoost (random undersampling boosting) is a type of ensemble learning technique which belongs to the gradient boosting category. In addition to Bao et al. (2020), gradient boosting methods have been used by Dutta et al.

[4] Appendix C considers two alternative extensions of Logit, namely Rare Event Logit model of King and Zeng (2001) and a class-balanced Logit model penalizing false negatives. Neither approach present a sizable advantage over a simple Logit model.

(2017) and Bertomeu et al. (2021) to predict accounting irregularities. This model uses the same 28 raw variables used in Bao et al. (2020).

(4) *SVM with linear kernel*. We use this specification of SVM as a naïve benchmark to Cecchini et al. (2010), Perols (2011), and Bao et al. (2020). This model uses the 11 financial ratios in Dechow et al. (2011) as predictors and aims to provide empirical evidence for the adverse impact of high collinearity on the predictive power of machine learning techniques.

(5) *MLP*. The MLP (multilayer perception) approach consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not

---

linearly separable. This approach belongs to the category of algorithms referred to as artificial neural network (ANN). The ANN is a method which is designed to simulate the way the human brain analyzes and processes information. ANN methods have been used by Green and Choi (1997), and Perols et al. (2017) to predict accounting irregularities. This model uses the 11 financial ratios in Dechow et al. (2011) as predictors.

(6) *SGD*. The SGD (stochastic gradient descent) is a generalized linear classifier that can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). This approach belongs to the SVM methods. This model uses the 11 financial ratios in Dechow et al. (2011) as predictors.

(7) *FML*. We combine the alternative models (SVM, Logit, SGD, LogitBoost, and MLP) and apply a weighted average of them. We call the method Fused Machine Learning (FML) following Sevim et al. (2014) and Hassanniakalager et al. (2020) as a blended approach

considering previous techniques. We use the normalised AUC score obtained in the cross-validation stage as weights for each ML technique. The probability of a fraud based on FML is calculated as the weighted average probability of fraud across the five alternative models. This model uses the 11 financial ratios in Dechow et al. (2011) as predictors.

We note that models 6 (SGD) and 7 (FML) have thus far not been employed in prior accounting research. Finding evidence that our model outperforms all these would provide strong support to the use of *LogitBoost* that applies ensemble learning to logistic regressions to predict fraud. To estimate SVM-FK and RUSBoost, we use the same predictors used by Cecchini et al. (2010) and Bao et al. (2020), respectively, because we are interested in a direct comparison with these studies. To estimate all the other models we use, as predictors, the 11 financial ratios in Dechow et al. (2011).

4.2 Model estimation

4.2.1 Estimating the model parameters

Each of the selected ML technique requires one or more hyperparameters for developing a predictive model. A practitioner can specify these hyperparameters by preference or by following a procedure to compare a range of values for each hyperparameter. We choose the latter approach to avoid bias and consider a wide range of choices for each hyperparameter as presented in Table 4. Among the possible choices, a widely used practice to optimize the hyperparameters is cross-validation (Ding et al., 2020).

In the cross-validation process, a sample dataset is randomly split into $k-1$ training and 1 testing subsamples (folds). Each of the k folds can act as a random testing subsample to evaluate performance metrics for alternative predictive models. Accordingly, for each random trial, the training subsample has $n(k-1)/k$ observations and testing subsample has n/k observations, where n denotes the number of observations in the cross-validation sample. We

conduct a 10-fold cross-validation to identify optimal hyperparameters using firm-year statements from 1991 to 2000. We use the average area under the receiving operating curve (AUC) as the performance criteria over 10 random testing subsets over the cross-validation sample. The combination of hyperparameters to yield the highest average AUC are then considered as the optimal hyperparameters. The optimal hyperparameters for each ML technique are shown in Table 4.

Table 4. Range of hyperparameters for alternative models

Technique	*Parameter*	*Range*	*Optimized specification*
SVM-FK	Class penalty C^+/C^-	{1, 2, 5, 10,20, 50, 100, 200,500}	50
RUSBoost	Number of estimators	{10, 20, 50, 100, 200, 500, 1000}	1,000
	Learning rate	$\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$	10^{-4}
SVM	Kernel (ϕ)	$\phi_1 = x.\, x^T$, $\phi_2 = \gamma(x.\, x^T)^3$, $\phi_3 = \exp(-\gamma\|x - x^T\|^2)$	ϕ_1
	Class penalty C^+/C^-	{1, 2, 5, 10,20, 50, 100, 200,500}	100
SGD	Regularization term	$R_1(w) = \Sigma_{j=1}^{m}\|w_j\|$, $R(w) = \frac{1}{2}\sum_{j=1}^{m} w_i^2$, $R_2(w) = \frac{1}{2}\sum_{j=1}^{2}$	
	Loss function	$L_1(y, \hat{y}) = \log(1 + \exp[-y_i\, \hat{y}_i])$, $\max(0, [1 - y_i\, \hat{y}_i]^2)$, $if\ y_i\, \hat{y}_i \geq -1$ $L_2(y, \hat{y}) = \{ -4\ y_i\, \hat{y}_i,$ $otherwise$	$L_1(y, \hat{y})$
	Class penalty C^+/C^-	{1, 2, 5, 10,20, 50, 100, 200,500}	200
LogitBoost	Class penalty C^+/C^-	{1, 2, 5, 10,20, 50, 100}	1
	Number of estimators	{10, 20, 50, 100, 200, 500}	20
	Learning rate	{0.1, 0.5, 0.9, 1}	0.9
MLP	Hidden layer size	{1, 2, 5, 10}	5
	Solving algorithm	stochastic gradient descent, Adam as in Kingma and Ba (2014)	Adam
	Activation function	$f_1(x) = x$, 1	$f_2(x)$

$$f_2(x) = \overline{1 + \exp(-x)}$$

Notes: We use a ten-fold cross-validation for a sample dataset over 1991-2000 using AUC as performance metric. In the table, x is the set of predictors with elements in form of $x_{i,j}, i = 1, \dots, n;\ j = 1, \dots, m$ with n observation and m element. In the SVM setting, $\gamma = m\_^{1}\sigma_x^2$ and $\|x - x^T\|^2$ is the squared Euclidean distance between the elements of x. For the SGD hyperparameters, w_j is the coefficient for element j in $\hat{y}_i = \Sigma_j{}^{m}{}_{=1} w_j x_{i,j} + b$ with the goal to minimize $E(w, b) = {}_{n}\_^{1}\ \Sigma_i{}^{n}{}_{=1}[L(y_i,\hat{y}_i) + aR(w)]$ where $L(y_i,\hat{y}_i)$ is the loss function and $R(w)$ is the regularization term.

An alternative approach to validate the model would be to select a validation sample based on a chronological order and thus select specific years as the validation sample. However, the disadvantage of this approach is that the performance may be biased by contextual events occurring during the period selected for validation. Under such conditions, the performance may not be generalizable to other periods in which the model is used. In contrast, the cross-validation approach minimizes potential biases arising from contextual events in the validation sample. Additionally, splitting the sample based on a chronological order may introduce bias because of the panel structure of the accounting data. Specifically, this approach ignores the within-firm and between year error correlations. Thus, this method is likely to fit noise specific to the firm or time causing high model performance in the validation period (Bertomeu et al., 2020).

Next, we train our ML models based on the optimal hyperparameters and update the ML models each year. We follow Bao et al. (2020) in updating the ML models based on an expanding-window basis. For each out-of-sample study period (one calendar year) from 2001 to 2010, we use the observations with a report date from 1991 to the last year before the study period as training observations. For instance, we use firm-year reports over 1991 to 2000 as our training sample to predict the probability of an AAER based firm-year reports in year 2001. In a similar fashion we use observations over 1991 to 2001 for detecting misstatements in 2002, and 1991 to 2009 for detecting misstatements in 2010. Such setting ensures at least 10 years'

worth of observations for training to estimate the probability of leading to an AAER based on the figures reported in the financial statements.

4.2.2 Addressing serial frauds

As discussed in section 3.1, more than 50% of AAERs span over multiple periods. Bao et al. (2020) show how not addressing the serial AAERs can lead to reporting the sensitivity up to 3 times higher. To address serial frauds there are at least two approaches. Bao et al. (2020) correct for serial fraud in the following way. For each year of out-of-sample, they first identify the positive cases for misstatements and then consider these observations as non-fraud in the training dataset. While their approach tends to imitate the sequence of events during SEC investigations, it increases the risk of using data that are only available in hindsight. Their analysis of serial fraud is subject to a look-ahead-bias, i.e., some information is used for processing the training samples which is not available at the point of time in which the training samples are observed.[5] An alternative treatment of serial frauds is conceptualized in Bertomeu (2020). He recommends that test samples should be selected before any assessments of performance are known. Hence, a bias-free approach to treat serial AAERs involves solely looking at the training observations and discarding the observations for the firms with a misstatement from the testing sample, prior to any model training. This approach does not depend on the SEC investigations for the testing sample and can guide the SEC in their investigations to identify the AAER cases immediately after the report of financial statements. In our analysis, we draw from Bertomeu (2020) and address the serial AAER cases by

[5] For instance, consider the case of detecting the likelihood of an AAER occurring in 2001 by using training samples over 1991 to 2000. Bao et al. (2020) discard the training samples (observations between 1991 to 2000) in case a firm has an AAER in the test sample (reports recorded in 2001). While in retrospect the data since 1991 is available for both test and training split, the same practice is not reproducible for a forward-looking practice. Furthermore, excluding the rare cases of AAERs in the training sample adversely affects the predictive power of ML models.

discarding the firms with prior records of AAERs from the out-of-sample observations to avoid overestimating the performance of ML techniques.

4.3 Measuring the effectiveness of fraud detection models

We use seven measures to evaluate the effectiveness of fraud detection models. (1) AUC is defined as the area under the ROC curve. It can be interpreted as the probability that a randomly chosen fraud observation will be ranked higher by the model than will a randomly chosen nonfraud observation. (2) Sensitivity (also referred to as "recall") refers to the probability of a true positive (TP); i.e., the probability that a fraud observation is correctly classified as fraud. (3) Specificity refers to the probability of a true negative (TN); i.e., the probability that a nonfraud observation is correctly classified as nonfraud. (4) Precision refers to the ratio of the number of the true positives to the total number of positives: TP/(TP+FP). (5) Following Bertomeu et al. (2021), we present the F1 score, calculated as $F_1 = 2TP/(2TP + FP + TN)$. (6) Following Bao et al. (2020) and Brown et al. (2020), we use the Normalized Discounted Cumulative Gain at the position k (NDCG@k). This is calculated as $NDCG@K =$ DCG@k/(ideal DCG@k) where DCG@k is calculated as $DCG@K = \sum_{i=1}^{k}(2^{rel_i} - 1)log_2(i + 1)$ where rel_i equals 1 if the *i*th observation in the ranking list is a true fraud, and 0 otherwise, *k* represents the *k* number of firm-years in a test period that have the highest predicted probability of fraud and *ideal DCG@k* is the DCG@k value when all the true instances of fraud are ranked at the top of the ranking list. Intuitively, this measure can be interpreted as the probability of a true negative in the top k observations in a test year that have the highest predicted probability of fraud. Finally (7) we present the ECM as in Perols (2011) and Perols et al. (2017) calculated as $ECM = C^{FN} \times P_{fraud} \times n^{FN}/n^{P} + C^{FP} \times P_{non-fraud} \times n^{FP}/n^{N}$ where $C^{FN} = 30$ and $C^{FP} = 1$ are the penalty terms for an FN and an FP respectively. The

P_{fraud} and are $P_{non-fraud}$ are the prior probabilities of fraud and non-fraud within the sample population. We derive these probabilities based on our cross-validation dataset using observations recorded 1991 to 2000 where $P_{fraud} = 99.2\%$ and $P_{non-fraud} = 0.8\%$. Finally, the n^{FN} and n^{FP} are the number of FN and FPs for each forecasting model in each year. We average the ECMs for each ML model across the 10 out-of-sample testing periods (2001 to 2010). With exception to ECM, a higher score for each measure indicates higher effectiveness in predicting fraud. We compute metrics (2)-(7) focusing on the top percentile of the distribution of the firm-year observations, based on the likelihood to be classified as fraudulent.

Each performance metric has its own merits. For instance, AUC, used among others by Cecchini et al. (2010), provides insights about the statistical properties of the predictive models. Sensitivity, as suggested by among others Bao et al. (2020) and Brown et al. (2020), can provide a more realistic evaluation of the models and assist/guide the SEC practice. In imbalanced samples (as in the fraud detection where positive cases are extremely rare), *sensitivity* is the most relevant measure to evaluate the effectiveness of a fraud detection model. The F1 provides a harmonic mean of precision and sensitivity considering both effectiveness and efficiency. Similarly, the ECM assigns different penalty terms for false positive and false negative misclassifications. Thus, it can be used as a bespoke average of precision and sensitivity based on a practitioner's interest. Accordingly, we use a range of performance metrics to allow for a more holistic evaluation of the performance of the ML methods. Yet, our main focus is on the sensitivity and the F1 scores at top 1 percentile. Similar to Bao et al. (2020), we select 1% because less than 1% of all firms in a year are detected for fraud by SEC (see also Beneish, 1999).

5. Results

In Section 5.1, which presents our main analysis, we examine the ability of our model to detect fraud. Section 5.2 investigates whether the models can raise an early warning for

future periods. In this analysis, we only consider the firm-year observations without AAERs and examine whether the ML methods can predict misstatements for these firms in subsequent years. We further investigate how many years in advance an ML technique can detect a misstatement.

5.1 Fraud detection

Table 5 (Panel A) presents our main results for the period 2001-2010. To compare our results with those in Bao et al. (2020) we also replicate the analysis using the same testing period as in their study (i.e., 2003-2008) in Panel B.

Table 5. Performance of alternative ML models to detect fraud

Panel A: 2001-2010

Technique	*AUC*	*Sensitivity*	*Specificity*	*Precision*	*F1 Score*	*NDCG*	*ECM*
SVM-FK-23	54.23%	2.45%	98.99%	0.66%	1.04%	2.01%	23.74%
RUSBoost-28	62.97%	3.43%	98.99%	0.66%	1.10%	1.40%	23.53%
SVM	63.12%	2.58%	98.99%	0.89%	1.32%	1.55%	23.72%
Logit	63.79%	7.33%	99.00%	1.80%	2.89%	3.25%	22.61%
SGD	64.89%	7.33%	99.00%	1.80%	2.89%	3.57%	22.61%
LogitBoost	64.24%	**8.04%**	**99.01%**	**2.08%**	**3.30%**	**4.41%**	**22.44%**
MLP	**65.70%**	3.41%	99.00%	1.19%	1.77%	2.52%	23.53%
FML	63.96%	6.74%	99.00%	1.53%	2.49%	3.06%	22.75%

Panel B: 2003-2008

Technique	*AUC*	*Sensitivity*	*Specificity*	*Precision*	*F1 Score*	*NDCG*	*ECM*
SVM-FK-23	50.66%	0.00%	98.98%	0.00%	–	0.00%	24.32%
RUSBoost-28	60.42%	3.33%	98.99%	0.52%	0.90%	1.19%	23.55%
SVM	65.14%	4.29%	99.00%	1.49%	2.21%	2.58%	23.31%
Logit	65.96%	11.24%	99.01%	2.55%	4.15%	4.79%	21.68%

SGD	67.06%	11.24%	99.01%	2.55%	4.15%	5.26%	21.68%
LogitBoost	66.47%	**12.43%**	**99.02%**	**3.01%**	**4.85%**	**5.99%**	**21.40%**
MLP	**68.85%**	5.68%	99.01%	1.99%	2.95%	4.20%	22.99%
FML	66.05%	11.24%	99.01%	2.55%	4.15%	5.11%	21.68%

Notes: The table reports the performance of the alternative ML models over the whole sample (2001-2010) and over a reduced time sample (2003-2008). Except for AUC, the performance metrics are based on predictions at top 1 percentile level. The values in bold represent the best performance for each column.

In both panels of the table, focusing on sensitivity, LogitBoost outperforms the other methods. The fused approach (FML) and stochastic gradient descent (SGD) methods are next in the ranking. When using the F1 Score, LogitBoost is still the best performing method, followed by SGD and FML. By looking at sensitivity and the F1 score, these three methods, which have not been used by prior literature yet, substantially outperform those used by prior literature. In particular, they outperform the methods used by the two main studies we use as a reference in our analysis, Cecchini et al. (2010) (SVM with financial kernel) and Bao et al. (2020) (RUSBoost). Additionally, we draw similar conclusions when focusing on the remaining measures assessing the performance of the models. Interestingly, focusing on sensitivity, the three best performing methods outperform both SVM-FK and RUSBoost by over 100%.

Comparing the two panels, we note that the performance for the models with 11 ratios is relatively unchanged. However, the performance of SVM-FK and RUSBoost with raw variables drastically decays when expanding beyond the study period in Bao et al. (2020) given that these models have lower AUC and sensitivity compared to those using 11 financial ratios. In sum, our proposed methodology uses a lower number of predictors which exhibit a lower level of collinearity and has higher effectiveness in detecting fraud than existing methods. Overall, our model, which applies ensemble learning to logistic regressions, which is a commonly used approach, outperforms the remaining seven benchmarks in detecting fraud.

Importantly, in both panels, we note that the performance of SVM-FK and RUSBoost is lower than that of the Logit model. This is in contrast with the results obtained by prior studies which introduce these models to the accounting literature (SVM-FK: Cecchini et al., 2010; RUSBoost: Bao et al., 2020). Our setting, however, differs from these papers in two main ways. First, we draw from Bertomeu (2020) for the treatment of serial fraud cases and remove observations for the firms with a misstatement from the testing sample that span both the training and test periods, prior to any model training. In contrast, Bao et al. (2020) re-code all the fraudulent years in the training period to zero in such cases; Cecchini et al. (2010) and Perols (2011) do not consider serial frauds. Second, we use cross-validation as opposed to partitioning the sample by chronological order to validate the models (e.g. Bao et al., 2020) (see also section 4.2).

5.2 Fraud prediction

In this section, we explore whether the selected ML models can detect misstatements before the SEC does. We focus only on the firm-year observations without AAERs and investigate whether any of the firms predicted by an ML model to have a misstatement at any accounting period end up in an AAER over the following years. We consider two forwardlooking performance measures: "forward precision" and "time-to-catch". The forward precision calculates the proportion of the firms, which are in the top 1 percentile likelihood of having a misstatement and are implicated in an AAER at least one year after being identified by an ML technique. For instance, for the out-of-sample period 2001, we identify the set of firms that are predicted to have the highest chance of reporting misstatements. We then calculate the proportion of those firms which show up in AAERs on or after 2002. In a similar fashion, the time-to-catch measures how early a model can identify the future occurrence of fraud. This is calculated as the difference, in number of years, between the year of the AAER

and the year in which the model predicts an AAER (based on the top 1 percentile of likelihood). Table 6 presents the forward-looking performance metrics.

The results in Table 6 show that LogitBoost performs best in predicting future fraud than any other model with respect to the average and median of both forward precision and time-to-catch. To add granularity to our analysis, we also compare, using a t-test (MannWhitney U-test) the mean (median) forward precision of each model with that of an SVM with a linear kernel based on 11 financial ratios (denoted as SVM). We use the linear SVM as a naïve benchmark given the simple structure. We find that the LogitBoost and SGD are the only models able to have significantly higher forward precision relative to an SVM. Finally, LogitBoost is the only model which is able to predict misstatements, on average, more than a year ahead of SEC investigations.

Table 6. Performance of alternative ML models to predict fraud

Measure	*Forward precision*			*Time-to-catch*		
		Standard deviation			*Median*	*Standard deviation*
			Median	*Mean*		
SVM-FK-23	0.00%	0.00%	0.00%	0.00	0	0.00
RUSBoost-28	0.65%	0.00%	1.30%	0.75	0	1.94
SVM	0.92%	0.00%	1.41%	0.60	0	**1.20**
Logit	1.80%	0.00%	2.77%	0.70	0	1.19
SGD	2.43%*	1.35%	**3.04%**	0.75	0.5	0.93
LogitBoost	**2.72%****	**2.82%***	2.53%	**1.15**	**1***	1.14
MLP	1.23%	0.00%	2.02%	0.50	0	0.92
FML	1.5[6]%	0.00%	2.55%	0.50	0	0.92

[6] **.1 Collinearity among the predictors**

Notes: The table examines the forward-looking performance of the eight ML models. The mean forward precision measures the proportion of the firms, in the top 1 percentile in each out-of-sample year based on the probability of being classified as fraudulent by a model, that are identified as reporting misstatements at least one year after that year. The time-to-catch measures the difference, in number of years, between the year of the AAER and the year of the model prediction of an AAER (based on the top 1 percentile of likelihood). The out-of-sample study period is 2001 to 2010 and the Fraud prediction performance is measured over the period 2002 to 2018. We also perform a t-test (Mann-Whitney U-test) for the null hypothesis that the mean (median) forward precision and the mean (median) time-to-catch of a model is lower to that of the SVM with a linear kernel; * and ** indicate statistical significance for these tests at the 10% and 5% level, respectively. The values in bold are the highest in each column.

6. Additional analyses

We identify three key methodological concerns in prior literature using ML to detect accounting irregularities: collinearity among the predictors, overlooking or not appropriately dealing with serial fraud, validating the hyperparameters using a chronological approach. In this section, we empirically examine the effects of these issues on the estimates of the ML methods. The results of these tests are presented in Appendix D (Tables D1, D2 and D3). variables (as in Bao et al 2020) as predictors. We repeat this practice 1,000 times and report our findings as in Table D1.

We discuss our collinearity concerns about prior literature in Section 3.2.2. In, particular, we show that the majority of the predictors used in Cecchini et al. (2010) and Bao et al. (2020) have strong linear dependence (VIF>10) with other predictors (Tables 2 and 3).

In the absence of a direct test of overfitting for ML methods, we use a Monte Carlo simulation to show the effects of collinearity on the estimates obtained from the fraud detection models. More specifically, we run a simulation, without replacement, using the firm-year reports from 1991 to 2010, focusing on a simple non-parametric Logit model. In each replication of the simulation, we draw a random sample of observations including 90% of population as training dataset and 10% as testing dataset. For each bootstrap, we calculate our performance metrics using either the 11 ratios (as in Dechow et al., 2010) or the 28 raw

The results of the Monte Carlo simulation show that the variability of the accuracy estimates, measured by the coefficient of variation (denoted as "noise to signal" in the table), is substantially higher when using 28 variables. These findings are in line with the view that collinearity among the predictors strongly increases the instability of the estimates and decreases the reliability of the conclusions that can be obtained from the fraud detection models.

6.2 Serial fraud

In Section 3.1, we present evidence on the incidence of serial fraud in our sample; notably, more than 50% of AAERs span over more than one annual accounting period (Figure 2). In Section 4.2.2, we explain how we deal with serial fraud in our research design and we argue that the method used by Bao et al. (2020) should be improved because it suffers from a look-ahead bias.

In Table D2, we further investigate the effect of serial fraud on the estimates of the ML methods. We repeat our estimations ignoring serial fraud. The results are substantially different from our main findings. One of the most remarkable differences is that RUSBoost, when not controlling for serial fraud, is the best performing model across all the measures of effectiveness of fraud detection. These findings suggest that the treatment of serial fraud has a crucial effect on the estimates obtained from the fraud prediction methods. This further confirms the importance of controlling for serial fraud.

6.3 Validation of the hyperparameters

We describe our approach to the validation of the hyperparameters (cross-validation) in Section 4.2.1. We argue that our approach overcomes the problems with the chronological approach to validation, which is used by part of prior literature.

Next, we examine the effect of the approach used for the validation of the hyperparameters on the estimates obtained from fraud detection methods. We repeat our estimations using the

chronological approach to the validation of the hyperparameters. Specifically, we use observations from 1991 to 1999 for training and firm-year observations of 2000 as the validation sample; we then compare the AUC for each choice of hyperparameters combination. The results, which are reported in Table D3, are markedly different from our main findings. In particular, it is worthwhile to note that the accuracy of SGD substantially increases relatively to our main results. We interpret these results as evidence that the validation approach to the hyperparameters strongly affects the conclusions that can be obtained by implementing the ML methods of fraud detection.

7. Conclusion

In this paper, we introduce a new fraud detection model to the accounting literature. The model combines the traditional logistic regression approach and more recently introduced machine learning (ML) methods in accounting research.

Our review of prior literature on ML methods used to detect fraud highlights three key methodological concerns. First, the high number of predictors used in the vast majority of prior studies may lead to collinearity and overfitting. Second, most prior studies do not make adjustments for serial fraud; overlooking serial fraud can lead to a substantial overstatement of the predictive ability of the models. Third, splitting the sample chronologically with the sample used for validation being restricted to a few pre-specified periods reduces the generalizability of the performance and ignores the within-correlation structure of the data. Our research design is aimed at minimizing the potential biases related to these concerns.

We compare the performance of our model with five models previously used in this strand of the literature and two which have not been previously used. First, we present the results obtained using logistic regressions (Logit) as in Dechow et al. (2011). Second, following Cecchini et al. (2010), we use SVM with a financial kernel (SVM-FK). Third, we employ the

RUSBoost method, which was introduced to the accounting research by Bao et al. (2020). Drawing from Perols (2011), we then use an SVM with linear kernel (SVM) and multilayer perception (MLP), which belongs to the category of algorithms referred to as artificial neural network. Next, we consider two methods which, to our knowledge, have not been used in accounting research: stochastic gradient descent (SGD) classifier, and a combination of the alternative models which we call fused ML (FML). We note that SGD is a generalized linear classifier like SVM.

Based on seven alternative measures assessing the ability to detect fraud, our results show that our method outperforms the models we use as benchmarks. Importantly, our method uses a smaller number of predictors than those used by prior literature (Bao et al., 2020; Cecchini et al., 2010; Perols et al., 2017) and has a substantially higher fraud detection ability. Our results also indicate that some of the main findings in prior ML literature on fraud are sensitive to the three key methodological concerns we identified. Additionally, we show that our model outperforms the other methods in predicting fraud in future periods; this result is particularly relevant because it can take several years for the SEC to conclude their investigations on potential fraud.

References

Abbasi A. A., Albrecht, C., Vance, A. & Hansen, J. (2012). A Meta-Learning Framework for Detecting Financial Fraud. *MIS Quarterly*, *36*(4), 1293-1327. https://doi.org/10.2307/41703508

Bao, Y., Ke, B., Li, B., Yu, Y.J. & Zhang, J. (2020). Detecting accounting fraud in publicly traded US firms using a machine learning approach. *Journal of Accounting Research*, *58*(1), 199-235. https://doi.org/10.1111/1475-679X.12292

Beneish, M. D. (1997). Detecting GAAP violation: Implications for assessing earnings management among firms with extreme financial performance. *Journal of accounting and public policy*, *16*(3), 271-309. https://doi.org/10.1016/S0278-4254(97)00023-9

Beneish, M. D. (1999). The detection of earnings manipulation. *Financial Analysts Journal*, *55*(5), 24-36. https://doi.org/10.2469/faj.v55.n5.2296

Bertomeu, J. (2020). Machine learning improves accounting: discussion, implementation and research opportunities. *Review of Accounting Studies*, *25*(3), 1135-1155. https://doi.org/10.1007/s11142-020-09554-9

Bertomeu, J., Cheynel, E., Floyd, E. & Pan, W. (2021). Using machine learning to detect misstatements. *Review of Accounting Studies*, *26*(2), 468-519. https://doi.org/10.1007/s11142-020-09563-8

Brown, N.C., Crowley, R.M. & Elliott, W.B. (2020). What are you saying? Using topic to detect financial misreporting. *Journal of Accounting Research*, *58*(1), 237-291. https://doi.org/10.1111/1475-679X.12294

Cai, Y. D., Feng, K. Y., Lu, W. C., & Chou, K. C. (2006). Using LogitBoost classifier to predict protein structural classes. *Journal of Theoretical Biology*, *238*(1), 172-176. https://doi.org/10.1016/j.jtbi.2005.05.034

Cecchini, M., Aytug, H., Koehler, G.J. & Pathak, P. (2010). Detecting management fraud in public companies. *Management Science*, *56*(7),1146-1160. https://doi.org/10.1287/mnsc.1100.1174

Cho, S., Vasarhelyi, M. A., Sun, T., & Zhang, C. (2020). Learning from machine learning in accounting and assurance. *Journal of Emerging Technologies in Accounting*, *17*(1), 1-10. https://doi.org/10.2308/jeta-10718

Dechow, P. M., Ge, W., Larson, C.R. & Sloan, R.G. (2011). Predicting material accounting misstatements. *Contemporary Accounting Research*, *28*(1), 17-82. https://doi.org/10.1111/j.1911-3846.2010.01041.x

Dechow, P. M., Sloan, R. G., & Sweeney, A. P. (1996). Causes and consequences of earnings manipulation: An analysis of firms subject to enforcement actions by the SEC. *Contemporary accounting research*, 13(1), 1-36. https://doi.org/10.1111/j.1911-3846.1996.tb00489.x

Ding, K., Lev, B., Peng, X., Sun, T., & Vasarhelyi, M. A. (2020). Machine learning improves accounting estimates: Evidence from insurance payments. *Review of Accounting Studies*, 25(3), 1098-1134. https://doi.org/10.1007/s11142-020-09546-9

Dutta, I., Dutta, S. & Raahemi, B. (2017). Detecting financial restatements using data mining techniques. *Expert Systems with Applications*, *90*, 374-393. https://doi.org/10.1016/j.eswa.2017.08.030

Friedman, J., Hastie, T. & Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The Annals of Statistics*, *28*(2), 337-407. https://doi.org/10.1214/aos/1016218223

Green, B. P., & Choi, J. H. (1997). Assessing the risk of management fraud through neural network technology. *Auditing: A Journal of Practice & Theory*, *16*(1), 14-28. https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=9708303198&site=ehostlive (Accessed: 20 July 2022).

Gujarati, D. N. (2011). *Econometrics by example* (Vol. 1). New York: Palgrave Macmillan.

Hassanniakalager, A., Sermpinis, G., Stasinakis, C. & Verousis, T. (2020). A conditional fuzzy inference approach in forecasting. *European Journal of Operational Research*, *283*(1), 196-216. https://doi.org/10.1016/j.ejor.2019.11.006

King, G. & Zeng, L., 2001. Logistic regression in rare events data. *Political Analysis*, *9*(2), 137-163. https://doi.org/10.1093/oxfordjournals.pan.a004868

Kingma, D.P. & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980

Krupa, J., & Minutti-Meza, M. (2022). Regression and Machine Learning Methods to Predict Discrete Outcomes in Accounting Research. *Journal of Financial Reporting, Forthcoming*. https://doi.org/10.2308/JFR-2021-010

Larcker, D.F. & Zakolyukina, A.A. (2012). Detecting deceptive discussions in conference calls. *Journal of Accounting Research*, *50*(2), 495-540. https://doi.org/10.1111/j.1475-679X.2012.00450.x

Lutz, R. W. (2006). Logitboost with trees applied to the wcci 2006 performance prediction challenge datasets. *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 1657-1660. https://ieeexplore.ieee.org/document/1716306

Otero, J., & Sánchez, L. (2006). Induction of descriptive fuzzy classifiers with the Logitboost algorithm. *Soft Computing*, *10*(9), 825-835. https://doi.org/10.1007/s00500-0050011-0

Perols, J.,L. (2011). Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory*, *30*(2), 19-50. https://doi.org/10.2308/ajpt-50009

Perols, J.L., Bowen, R.M., Zimmermann, C. & Samba, B. (2017). Finding needles in a haystack: Using data analytics to improve fraud prediction. *The Accounting Review*, *92*(2), 221-245. https://doi.org/10.2308/accr-51562

Pham, B. T., & Prakash, I. (2019). Evaluation and comparison of LogitBoost Ensemble, Fisher's Linear Discriminant Analysis, logistic regression and support vector machines methods for landslide susceptibility mapping. *Geocarto International*, *34*(3), 316-333. https://doi.org/10.1080/10106049.2017.1404141

Sevim, C., Oztekin, A., Bali, O., Gumus, S. & Guresen, E. (2014). Developing an early warning system to predict currency crises. *European Journal of Operational Research*, *237*(3), 1095-1104. https://doi.org/10.1016/j.ejor.2014.02.047

Summers, S. L., & Sweeney, J. T. (1998). Fraudulently misstated financial statements and insider trading: An empirical analysis. *Accounting Review, 73(1),* 131-146. https://www.jstor.org/stable/248345

Zhang, G., & Fang, B. (2007). LogitBoost classifier for discriminating thermophilic and mesophilic proteins. *Journal of Biotechnology*, *127*(3), 417-424. https://doi.org/10.1016/j.jbiotec.2006.07.020

Appendix A: Definitions of the financial ratios used in Dechow et al. (2011) as fraud predictors

Variable	*Definition*	*Compustat codes*
WC accruals	Working capital accruals scaled by average total assets. Working capital accruals are defined as the change in current assets less the change in cash and short-term investments less the sum of the change of current liabilities, less the change of short-term debt less the change of tax payable.	ACT, CHE, LCT, DLC, TXP, AT
RSST accruals	Sum of the change in working capital accruals, the change in net long term operating assets and the change in net financial assets scaled by average assets. Following Richardson, Sloan, Soliman, and Tuna (2005), net long term operating assets is defined as the difference between the sum of change in total assets less current assets and investments and advances and the sum of total liabilities less current liabilities and long term debt. Net financial assets is defined as the difference between the sum of short term investments and investments and advances less the sum of the long-term debt, short-term debt and preferred stock.	ACT, CHE, LCT, DLC, TXP, AT, IVAO, LT, DLTT, IVST, PSTK
Change in receivables	Change in receivables scaled by average total assets	RECT, AT
Change in inventory	Change in inventory scaled by average total assets	INVT, AT
% soft assets	Percentage of soft assets, defined as total assets less PPE and cash as the proportion of total assets.	AT, PPENT, CHE
Change in cash sales	Change in cash sales defined as the difference between sales and change in receivables.	SALE, RECT
Change in cash margin	Change in cash margin defined as 1 less ratio of costs of good soles less the change in inventory plus the change in accounts receivables to cash sales.	COGS, INVT, AP, SALE, RECT
Change in return on assets	Change in return on assets	Δ(IB/average AT)
Change in free cash flows	Change in free cash flows defined as the difference between income before extra items and RSST accruals, scaled by average assets.	IB, AT
Actual issuance	Indicator variable that equals to one if a firm issues either stock or debt during the year, and zero otherwise.	SSTK, DLTIS
Book-to-market	Book to market ratio	SEQ, PRCC_F, CSHO)

Appendix

B: Additional evidence on the effectiveness of LogitBoost

In this appendix, we provide further empirical evidence on the merits of applying ensemble learning to logistic regressions (LogitBoost) relative to ensemble learning alone (AdaBoost). Specifically, Table B.1 compares LogitBoost with AdaBoost with a decision tree as the base model using a number of performance metrics. For both specifications, we perform a 10-fold cross-validation to validate the model and optimize the hyperparameters. The optimal set of hyperparameters for LogitBoost is presented in Table 4. The optimal setting for the AdaBoost is $C^+/C^- = 50$, learning rate of 0.1 and 500 estimators. The results in Table B.1 highlight a major improvement, across all performance metrics, over the testing sample, by using LogitBoost relative to AdaBoost.

Table B.1 Performance of alternative AdaBoost extensions over 2001-2010.

Metric	*AdaBoost*	*LogitBoost*
AUC	54.63%	**64.24%**
Sensitivity @ 1 percentile	2.41%	**8.04%**
Specificity @ 1 percentile	98.99%	**99.01%**
Precision @ 1 percentile	0.84%	**2.08%**
F1 Score @ 1 percentile	1.25%	**3.30%**
NDCG @ 1 percentile	1.88%	**4.09%**
ECM @ 1 percentile	23.76%	**22.44%**

Notes: The table compares the cross-validation and out-of-sample performance for a traditional AdaBoost model against the setting used in this study. The traditional model is defined as an AdaBoost setting with a decision tree as the base model (column "AdaBoost") compared to the proposed setting of using a Logit as a base model (column "LogitBoost"). The out-of-sample results are averages over 2001 to 2010. The values in bold are the best performance for each row.

Appendix

C: Extensions of Logit models

In this appendix, we compare the performance of a simple Logit model relative to two extensions of Logit that account for imbalanced samples. King and Zeng (2001) argue the coefficients in a Logit model are biased when samples are finite and positive cases are scarce. To address this bias, they propose the Rare Event Logit (RE-Logit) approach that we consider as the first alternative. The second extension is a Class-balanced Logit (C-Logit) model as a Generalized Linear Model where the logistic loss function is modified to penalize false negatives. Following Table 4, we consider a class-weight penalty term C^+/C^- in range {1,2,5, 10,20,50,100,200,500}. We perform a 10-fold cross-validation to optimize the class-weight hyperparameter in C-Logit. The optimal setting for the C-Logit is $C^+/C^- = 20$. Table C.1 compares the three extensions of Logit. The results in Table C.1 present no sign of a major improvement in favor of either RE-Logit or C-Logit.

Table C.1 Performance of alternative AdaBoost extensions over 2001-2010.

Metric	*Logit*	*RE-Logit*	*C-Logit*
AUC	63.79%	**66.15%**	63.93%
Sensitivity @ 1 percentile	**7.33%**	7.33%	4.00%
Specificity @ 1 percentile	**99.00%**	99.00%	99.00%
Precision @ 1 percentile	**1.80%**	1.80%	1.47%
F1 Score @ 1 percentile	**2.89%**	2.89%	2.14%
NDCG @ 1 percentile	3.25%	**3.31%**	2.24%
ECM @ 1 percentile	22.61%	**22.61%**	23.39%

Appendix

Notes: The table compares the cross-validation and out-of-sample performance for a Rare Event Logit model and a Class-balanced Logit against the setting used in main text. The benchmark model is a simple Logit model compared to the RE-Logit and C-Logit. The out-of-sample results are averages over 2001 to 2010. The values in bold are the best performance for each row.

D: Effects of the methodological issues in prior literature

In this appendix, we empirically examine the effects of the three key methodological issues in prior literature on the estimates of the ML methods. The tests and the results are described in Section 5.3.

Table D.1 Monte Carlo simulation to examine the effects of collinearity among the predictors

Measure	*Predictors*	*Accuracy*	*Noise-to-signal*
dropped serial frauds	-	51.55 (6.54)	12.69%
AUC	11 ratios	69.95% (7.36%)	10.52%
	28 raw vars	61.93% (6.96%)	11.24%
sensitivity @1	11 ratios	3.23% (4.91%)	152.01%
	28 raw vars	1.08% (3.02%)	279.63%
Precision @1	11 ratios	0.64% (0.95%)	148.44%
	28 raw vars	0.21% (0.59%)	280.95%
Specificity @1	11 ratios	99% (0.01%)	0.01%
	28 raw vars	99% (0.01%)	0.01%
NDGC @1	11 ratios	1.44% (2.18%)	151.39%
	28 raw vars	0.56% (1.77%)	316.07%
Count training positive	-	592.33 (7.48)	1.26%
Count testing positive	-	65.68 (7.48)	11.39%

Notes: In each Monte Carlo replication, the number of training and testing observations are 66,681 and 7,410 respectively. The positive cases in this scenario are removed from the training sample and are hence larger than total number of positive cases in the testing sample. The column "Accuracy" reports the measures (mean) of fraud detection effectiveness; in brackets, the standard deviation is reported. The column "Noise-to-signal" presents the ratio of standard deviation over the mean for each of the measures of fraud detection effectiveness.

Appendix

Table D2. Effect of ignoring serial fraud

Technique	*AUC*	*Sensitivity*	*Specificity*	*Precision*	*F1 Score*	*NDCG*	*ECM*
SVM-FK-23	60.78% (7.7%)	2.6% (3.51%)	99.01% (0.04%)	2.9% (4.14%)	2.49% (3.25%)	3.61% (4.93%)	23.69% (0.85%)
RUSBoost-28	**75.72% (7.95%)**	**5.56% (4.83%)**	**99.05% (0.07%)**	**7.2% (7.1%)**	**6.12% (5.57%)**	**7.78% (6.85%)**	**22.97% (1.19%)**
SVM	62.7% (6.2%)	2.76% (3.22%)	99.0% (0.02%)	2.53% (2.08%)	2.55% (2.5%)	2.26% (2.29%)	23.67% (0.77%)
Logit	62.9% (6.68%)	4.09% (3.93%)	99.01% (0.03%)	3.41% (2.55%)	3.55% (3.01%)	3.11% (2.57%)	23.35% (0.94%)
SGD	63.21% (6.37%)	3.79% (4.18%)	99.01% (0.03%)	3.11% (2.97%)	3.25% (3.31%)	3.11% (3.15%)	23.42% (1.0%)
LogitBoost	62.72% (6.68%)	4.14% (3.99%)	99.01% (0.03%)	3.65% (2.85%)	3.67% (3.14%)	3.63% (3.21%)	23.34% (0.96%)
MLP	63.53% (7.04%)	3.31% (3.9%)	99.01% (0.03%)	3.08% (2.73%)	3.08% (3.15%)	3.36% (3.62%)	23.54% (0.94%)
FML	62.99% (6.48%)	3.92% (4.0%)	99.01% (0.03%)	3.15% (2.48%)	3.34% (3.04%)	3.05% (2.87%)	23.39% (0.96%)

Notes: The table reports the performance of the alternative ML methods over the whole sample (2001-2010) when serial treatments are ignored. Except for AUC, the performance metrics are based on predictions at top 1 percentile level. The values in bold represent the best performance for each column. In brackets, we report the standard deviation of the measures.

Table D3. Effect of using a chronological approach to the validation of the hyperparameters

Panel A: Optimal hyperparameters by chronological validation

Technique	*Parameter*	*Range*	*Optimized specification*
SVM-FK	Class penalty C^+/C^-	$\{1, 2, 5, 10, 20, 50, 100, 200, 500\}$	100
RUSBoost	Number of estimators	$\{10, 20, 50, 100, 200, 500, 1000\}$	500
	Learning rate	$\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$	10^{-1}
SVM	Kernel (ϕ)	$\phi_1 = x.\, x^T$, $\phi_2 = \gamma(x.\, x^T)^3$, $\phi_3 = \exp(-\gamma\|x - x^T\|^2)$	ϕ_1
	Class penalty C^+/C^-	$\{1, 2, 5, 10, 20, 50, 100, 200, 500\}$	100
SGD	Regularization term	$R_1(w) = \sum_{j=1}^{m}\|w_j\|$, $R_2(w) = \frac{1}{2}\sum_{j=1}^{m} w_{i2}$,	$R_1(w)$
	Loss function	$L_1(y, \hat{y}) = \log(1 + \exp[-y_i\, \hat{y}_i])$, $L_2(y, \hat{y}) = \begin{cases} \max(0, [1 - y_i\, \hat{y}_i]^2), & if\ y_i\, \hat{y}_i \geq -1 \\ -4\ y_i\, \hat{y}_i, & otherwise \end{cases}$	$L_1(y, \hat{y})$
	Class penalty C^+/C^-	$\{1, 2, 5, 10, 20, 50, 100, 200, 500\}$	2
LogitBoost	Class penalty C^+/C^-	$\{1, 2, 5, 10, 20, 50, 100\}$	1
	Number of estimators	$\{10, 20, 50, 100, 200, 500\}$	20
	Learning rate	$\{0.1, 0.5, 0.9, 1\}$	0.9
MLP	Hidden layer size	$\{1,\ 2,\ 5,\ 10\}$	2
	Solving algorithm	stochastic gradient descent, Adam as in Kingma and Ba (2014)	Adam
	Activation function	$f_1(x) = x$, $f_2(x) = \frac{1}{1 + \exp(-x)}$	$f_1(x)$

Panel B: Performance of alternative ML models specified by chronological validation

Technique	*AUC*	*Sensitivity*	*Specificity*	*Precision*	*F1 Score*	*NDCG*	*ECM*
SVM-FK-23	51.91% (11.06%)	2.45% (6.0%)	98.99% (0.01%)	0.66% (1.33%)	0.98% (2.05%)	1.78% (4.38%)	23.74% (1.41%)
RUSBoost-28	60.93% (11.42%)	0.45% (1.36%)	98.99% (0.01%)	0.26% (0.77%)	0.33% (0.98%)	0.32% (0.96%)	24.22% (0.33%)
SVM	63.12% (11.44%)	2.58% (5.42%)	98.99% (0.02%)	0.89% (1.92%)	1.32% (2.84%)	1.55% (3.47%)	23.72% (1.29%)
Logit	63.79% (11.9%)	7.33% (11.47%)	**99.0% (0.03%)**	1.8% (2.77%)	2.74% (4.14%)	3.25% (4.82%)	22.61% (2.7%)
SGD	62.85% (11.54%)	**7.46% (11.46%)**	**99.0% (0.03%)**	**1.81% (2.78%)**	**2.77% (4.15%)**	**3.66% (5.48%)**	**22.58% (2.7%)**
LogitBoost	**63.87% (11.68%)**	4.0% (7.63%)	**99.0% (0.03%)**	1.47% (2.77%)	2.14% (4.06%)	2.16% (4.32%)	23.39% (1.81%)
MLP	63.86% (11.97%)	7.33% (11.47%)	**99.0% (0.03%)**	1.8% (2.77%)	2.74% (4.14%)	3.44% (5.12%)	22.61% (2.7%)

FML	63.19% (11.47%)	6.74% (11.68%)	**99.0% (0.03%)**	1.53% (2.8%)	2.37% (4.21%)	2.94% (5.11%)	22.75% (2.75%)

Notes: The table reports optimal hyperparameters from the chronological validation approach (Panel A) and the performance of the alternative ML models over the testing sample 2001-2010 (Panel B). Except for AUC, the performance metrics are based on predictions at top 1 percentile level. The values in bold represent the best performance for each column. In brackets, we report the standard deviation of the measures.