ECON JOURNAL WATCH
Scholarly Comments on Academic Economics

Econ Journal Watch 18(1)
March 2021: 61–70

Critique of an Article on Machine Learning in the Detection of Accounting Fraud

Stephen Walker[1]

Link to Abstract

This critique treats an article in *Journal of Accounting Research* entitled “Detecting Accounting Fraud in Publicly Traded U.S. Firms Using a Machine Learning Approach” by authors Yang Bao, Bin Ke, Bin Li, Y. Julia Yu, and Jie Zhang (Bao et al. 2020). In addition to the published paper, the authors provide their Matlab code with an associated dataset in a CSV file and other documents hosted at the code-sharing service Github (**link**). This paper applies their code and dataset to replicate the results and studies the key assumption driving those results.

Within the fields of accounting and finance, corporate fraud detection models have been the subject of a significant volume of work. The literature follows a long line of prediction and detection models found in the literature on capital markets. Parties with interest in these models include the investing public and regulatory bodies such as the Securities and Exchange Commission. Previous corporate frauds including Enron and Worldcom left significant damage in their wake, affecting not only their employees and investors but also the public’s trust and faith in capital-market institutions. The great hope is that an early warning system can alert the Securities and Exchange Commission and investors to potential fraud and act before the fraud grows too large.

The previous standard in the accounting literature for detecting accounting fraud is known as the F-Score, which is based on a seven-variable logistic regression model published by Patricia Dechow and collaborators (2011). For modeling pur-

[1] Graduate student, University of California, Berkeley, CA 94720.

poses, the best proxy for accounting fraud is the SEC-issued Accounting and Auditing Enforcement Release (AAER), an enforcement action that describes the fraud and typically orders a restatement of previously issued financial reports (e.g., 10-Ks). The observable covariates to these fraud models are financial statement ratios that might include changes in sales, accounts receivables, and inventories, in addition to indicator variables for capital-markets activity including share or debt issuances. These ratios are based on a long line of theoretical and empirical work. A novel innovation of the Bao et al. (2020) paper is that they do not use financial ratios, but rather apply raw financial variables taken directly from the financial statements.

The authors provide a dataset that includes a total of 146,045 firm-year observations from 1991–2014. The data comes from the CompuStat database. AAER data is sourced from the USC Marshall School of Business (previously the Haas School of Business). Unique AAER cases total 413 (each of which may last multiple years), and the sample's total number of fraud-case firm-years is 964. Taking the 964 AAER-affected firm-years and dividing by the total of 146,045 firm-years gives an approximation for the unconditional probability of finding fraud for any firm in any given year as 0.7 percent. Fraud is a rare event, and comparing detection rates against this unconditional expectation is important within accounting research.

Replicating the paper

Replicating the paper is relatively simple. The software Matlab is required. The Matlab code file is called "run_RUSBoost28." The dataset is a CSV file called "uscecchini28.csv." The column headers are shown in Table 1.

The dependent variable is an indicator variable equaling 1 if the AAER covered the firm-year in the data, and zero otherwise which is in the dataset's column 9, labeled *misstate*. The independent variables are 28 raw financial statement variables reported by the company in their annual report and shown in columns 10–37, which include items such as total assets and ending price per share for the period. In the Matlab code, the dataset will be divided into a training and test set. For example, the first looped-trained model was based on data covered by the period from 1991 through 2001. The model was then applied out of sample, e.g., to the year 2003, and that application generated a probabilistic score for each firm in that year. The top 1 percent of the probability scores were taken from this selection and if there is a firm in this subset with an actual AAER for that year, it is counted as a correctly identified positive hit. The fraction of correct hits is the positive predictive value. The model was run iteratively for each year in the study's test period, 2003 through 2008.

TABLE 1. CSV file (the dataset)

Position	Column	Description
1	fyear	Fiscal Year
2	gkvey	Compustat firm identifier
3	sich	4-digit Standard Industrial Classification Code (SIC)
4	insbnk	An indicator variable for financial institutions between SIC 6000–6999
5	understatement	An indicator variable if the misstate indicator involved an understatement
6	option	Not used
7	p_aaer	Identifier for AAER
8	new_p_aaer	New Identifier for AAER
9	misstate	Indicator variable for misstatement
10	act	Current Assets - Total
11	ap	Accounts Payable - Trade
12	at	Assets - Total
13	ceq	Common/Ordinary Equity - Total
14	che	Cash and Short-Term Investments
15	cogs	Cost of Goods Sold
16	csho	Common Shares Outstanding
17	dlc	Debt in Current Liabilities
18	dltis	Long-Term Debt Issuance
19	dltt	Long-Term Debt Total
20	dp	Depreciation and Amortization
21	ib	Income Before Extraordinary Items
22	invt	Inventories - Total
23	ivao	Investment and Advances Other
24	ivst	Short-Term Investments - Total
25	lct	Current Liabilities - Total
26	lt	Liabilities - Total
27	ni	Net Income (Loss)
28	ppegt	Property, Plant and Equipment - Total (Gross)
29	pstk	Preferred/Preference Stock (Capital) - Total
30	re	Retained Earnings
31	rect	Receivables Total
32	sale	Sales/Turnover (Net)
33	sstk	Sale of Common and Preferred Stock
34	txp	Income Taxes Payable
35	txt	Income Taxes - Total
36	xint	Interest and Related Expense - Total
37	prcc_f	Price Close - Annual - Fiscal

Machine learning requires measuring results using a hold-out test sample because machine learning can overfit training datasets and produce results that are too good to be true. An iterative approach is preferable because it shows results as it steps through time, which is what would be experienced in the real world, and thus adds validity to the model. A two-year (or longer) gap between the training sample and test sample is required because AAERs are not immediately known when financial reports are issued. In fact, many years can pass between the financial report and the AAER issuance. A modeler must ask (in the spirit of Senator Howard Baker): What can the model know, and when can the model know it?

One issue related to that question involves serial frauds. Some serial frauds may traverse both training and test periods since they cover more than the gap period. To address this issue, the readme file that accompanies the data and code (**link**) notes:

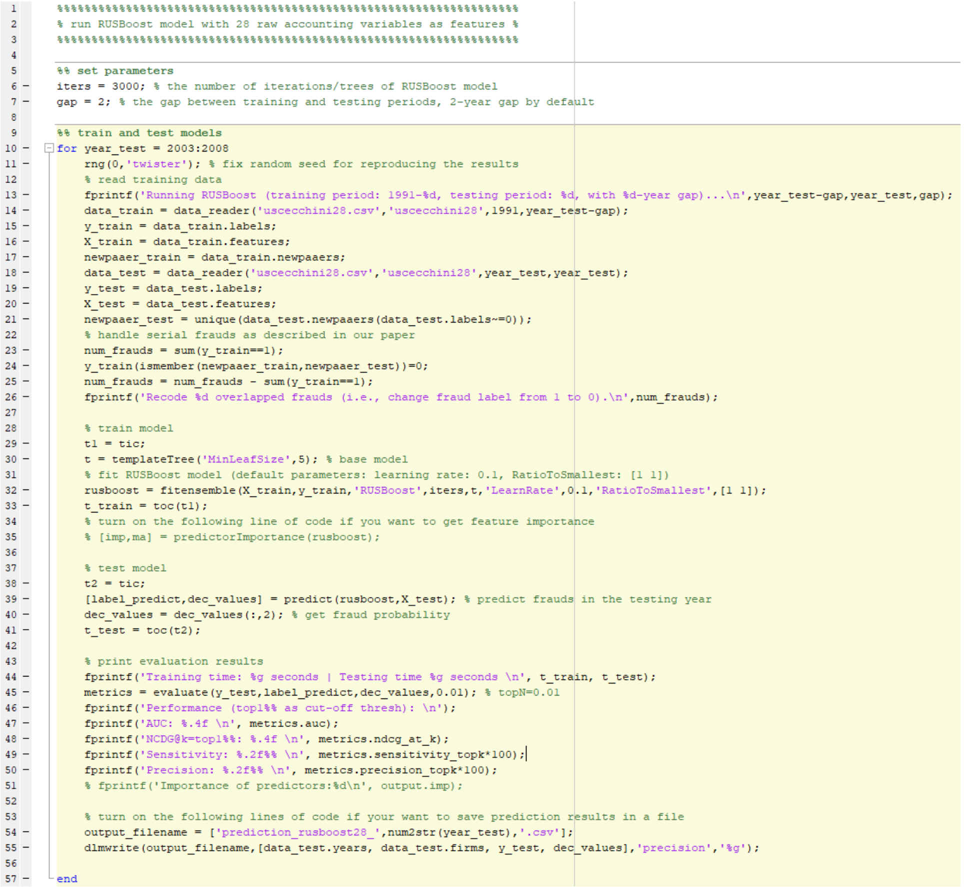
> The variable new_p_aaer is used for identifying serial frauds as described in Section 3.3 (see the code in "RUSBoost28.m" for more details).

Section 3.3 from their paper is reported in its entirety below, with boldface added to emphasize the action described.

> 3.3 SERIAL FRAUD
>
> Accounting fraud may span multiple consecutive reporting periods, creating a situation of so-called "serial fraud." In our sample, the mean, median, and 90th percentile of the duration of the disclosed accounting fraud cases is two years, two years, and four years, respectively, suggesting that it is **common for a case of fraud to span multiple consecutive reporting periods.** Such serial fraud may overstate the performance of the ensemble learning method if instances of fraudulent reporting span both the training and test periods. This is because ensemble learning is more flexible and powerful than the logistic regression model, and may therefore be better able to fit a fraudulent firm than a fraudulent firm-year. **Hence, enhanced performance of the ensemble learning method may result from the fact that both the training and test samples contain the same fraudulent firm**; the ensemble learning model may not perform as well when the sample contains different firms. **To deal with this concern, we break up those cases of serial fraud that span both the training and test periods. Because we have a small number of fraudulent firm-years relative to the number of non-fraudulent firm-years in any test year, we recode all the fraudulent years in the training period to zero for those cases of serial fraud that span both the training and test periods.** Although this approach helps us avoid the problems associated with serial fraud, it may also introduce measurement errors into the training data. (Bao et al. 2020, 211–212, my emphases)

In summary, serial fraud concerns AAER cases that span multiple reporting periods. However, the section does not directly address why the column *new_p_aaer* was created. Returning to the Matlab code for an explanation, Figure 1 shows the code for the model.

Figure 1. The Matlab code



```
 1      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 2      % run RUSBoost model with 28 raw accounting variables as features %
 3      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 4
 5      %% set parameters
 6  -   iters = 3000; % the number of iterations/trees of RUSBoost model
 7  -   gap = 2; % the gap between training and testing periods, 2-year gap by default
 8
 9      %% train and test models
10  -   for year_test = 2003:2008
11  -       rng(0,'twister'); % fix random seed for reproducing the results
12          % read training data
13  -       fprintf('Running RUSBoost (training period: 1991-%d, testing period: %d, with %d-year gap)...\n',year_test-gap,year_test,gap);
14  -       data_train = data_reader('uscecchini28.csv','uscecchini28',1991,year_test-gap);
15  -       y_train = data_train.labels;
16  -       X_train = data_train.features;
17  -       newpaaer_train = data_train.newpaaers;
18  -       data_test = data_reader('uscecchini28.csv','uscecchini28',year_test,year_test);
19  -       y_test = data_test.labels;
20  -       X_test = data_test.features;
21  -       newpaaer_test = unique(data_test.newpaaers(data_test.labels~=0));
22          % handle serial frauds as described in our paper
23  -       num_frauds = sum(y_train==1);
24  -       y_train(ismember(newpaaer_train,newpaaer_test))=0;
25  -       num_frauds = num_frauds - sum(y_train==1);
26  -       fprintf('Recode %d overlapped frauds (i.e., change fraud label from 1 to 0).\n',num_frauds);
27
28          % train model
29  -       t1 = tic;
30  -       t = templateTree('MinLeafSize',5); % base model
31          % fit RUSBoost model (default parameters: learning rate: 0.1, RatioToSmallest: [1 1])
32  -       rusboost = fitensemble(X_train,y_train,'RUSBoost',iters,t,'LearnRate',0.1,'RatioToSmallest',[1 1]);
33  -       t_train = toc(t1);
34          % turn on the following line of code if you want to get feature importance
35          % [imp,ma] = predictorImportance(rusboost);
36
37          % test model
38  -       t2 = tic;
39  -       [label_predict,dec_values] = predict(rusboost,X_test); % predict frauds in the testing year
40  -       dec_values = dec_values(:,2); % get fraud probability
41  -       t_test = toc(t2);
42
43          % print evaluation results
44  -       fprintf('Training time: %g seconds | Testing time %g seconds \n', t_train, t_test);
45  -       metrics = evaluate(y_test,label_predict,dec_values,0.01); % topN=0.01
46  -       fprintf('Performance (top1%% as cut-off thresh): \n');
47  -       fprintf('AUC: %.4f \n', metrics.auc);
48  -       fprintf('NCDG@k=top1%%: %.4f \n', metrics.ndcg_at_k);
49  -       fprintf('Sensitivity: %.2f%% \n', metrics.sensitivity_topk*100);
50  -       fprintf('Precision: %.2f%% \n', metrics.precision_topk*100);
51          % fprintf('Importance of predictors:%d\n', output.imp);
52
53          % turn on the following lines of code if your want to save prediction results in a file
54  -       output_filename = ['prediction_rusboost28_',num2str(year_test),'.csv'];
55  -       dlmwrite(output_filename,[data_test.years, data_test.firms, y_test, dec_values],'precision','%g');
56
57  -   end
```

Line 10 starts the loop that runs the model iteratively stepping through each year of the test period from 2003–2008. Line 21 creates a list of unique values of AAER identifiers where the *misstate* column is not equal to zero (equal to 1) for the test set. Line 24 performs the action described in Section 3.3 and sets the *y_train* indicator values to zero where there is a match in the AAER identifiers in the training sample to the previously created list from the test sample.

The intention of Section 3.3 appears to be correctly coded in Matlab. However, what is the *new_p_aaer* field? In Table 1, the 7th position contains another field called *p_aaer*. The *p_aaer* field is the AAER number that matches the SEC issued number, which can be searched on the SEC website (**link**). When comparing these two columns, it appears that *new_p_aaer* takes the original AAER number and adds a '1' or '2.' In fact, all but 17 AAER cases take the original AAER number and add a '1.'

I sent an email to the authors of the paper copying their editor and asked specifically about this issue. Professor Ke Bin sent the following response on behalf of the author group to all recipients of the original email (boldface added):

> As we discussed in Section 3.3 of our paper, "we recode all the fraudulent years in the training period to zero for those cases of serial fraud that span both the training and test periods." **Our serial frauds have two requirements: (1) have the same AAER id, and (2) are consecutive in our sample. "1" and "2" are suffix to distinguish serial frauds with the same AAER id but not consecutive in our sample.**

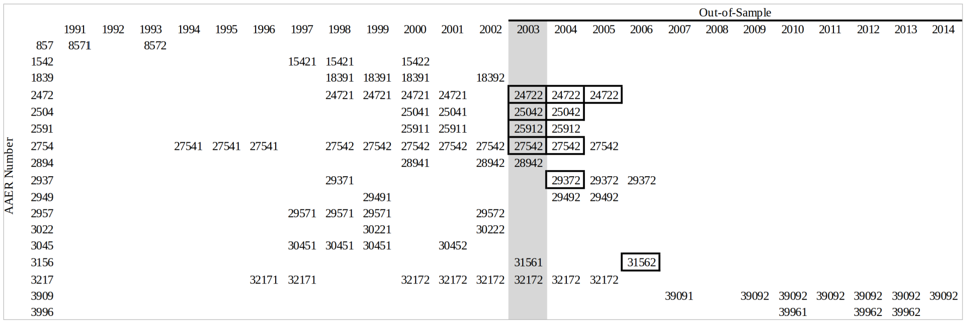
I understand the first part of the requirement. However, I do not understand the second part to the requirement—which was not described in the paper or in the online supporting documents. The serial fraud issue is a problem with the span of the fraud itself, not whether it is consecutive in their sample.

The reason that some cases are not consecutive in the sample was provided by the next explanation, given by Professor Ke when I asked why there were a few missing firm-year observations in the sample.

> We require all observations to contain non-missing values for the 28 raw accounting variables, consistent with prior studies cited in our paper. Those observation [related to the 17 AAERs] are dropped because one of the 28 raw variables are missing in WRDS COMPUSTAT database. For example, firm-years of AAER No. 2472, 2504, 2591, and 2894 are missing DLTIS (Long-term debt issuance) and firm-years of AAER No. 2754 and 3217 are missing XINT (Interest and related expense, total).

To show what Professor Ke is speaking to, Figure 2 shows the AAERs at issue. There are only 17 AAERs where *new_p_aaer* changes values because of the "not consecutive in our sample" issue out of a total of 413 unique AAERs in their sample. Additionally, a large fraction of correct cases identified by the model are related to these 17 AAERs. The number of firm-years correctly identified by the AAERs from 2003–2008 total 10 firm-years and are shown in the bolded boxes. The total correct cases identified by their model are 16 firm-years. So, 63 percent of the correct cases are associated with this issue.

Figure 2. Seventeen AAER cases with two different new AAER identifiers



												Out-of-Sample											
AAER Number	1991	1992	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014
857	8571		8572																					
1542							15421	15421		15422														
1839								18391	18391	18391		18392												
2472								24721	24721	24721	24721		24722	24722	24722									
2504										25041	25041		25042	25042										
2591										25911	25911		25912	25912										
2754				27541	27541	27541		27542	27542	27542	27542	27542	27542	27542	27542									
2894										28941		28942	28942											
2937								29371						29372	29372	29372								
2949									29491					29492	29492									
2957							29571	29571	29571			29572												
3022									30221			30222												
3045							30451	30451	30451		30452													
3156													31561			31562								
3217						32171	32171			32172	32172	32172	32172	32172	32172									
3909																	39091		39092	39092	39092	39092	39092	39092
3996																				39961		39962	39962	

Professor Ke's explanation is not consistent with how other variables are handled in the dataset. The statement suggests a rule that an observation is dropped if it has a missing CompuStat variable. According to the "SAS coding.pdf" file (**link**), the authors recoded *txp*, *ivao*, *ivst*, and *pstk* to 0 if they were missing. If done for these four variables, why are variables *dltis* and *xint* inconsistently handled?

However, the real issue is not these missing observations per se. Rather, it is the additional requirement that a consecutive sample be required for serial fraud identification. Section 3.3 of their paper describes the bias in machine learning related to serial fraud occurring when "both the training and test samples contain the same fraudulent firm" (Bao et al. 2020, 211–212). To illustrate, take for example AAER No. 2504. This AAER affected Delphi Corporation for the years 2000–2004 and was issued by the SEC in 2006. Summarizing Delphi in context of the Matlab code,

- If an AAER identifier from the test set matches the same identifier from the training set, the Matlab model recodes AAER's *misstate* = 1 in the training set to 0.
- As shown in Figure 2, the AAER identifier for Delphi **changes** to 25041 in the training set and to 25042 in the test set.
- Because Delphi 25042 is not in the training set, the Matlab code **will not** recode Delphi 25041's *misstate* = 1 to 0.

Because the Matlab code treats Delphi AAER No. 2504 as two different AAERs 25041 and 25042, the same fraudulent firm is contained in both the training and test samples. Therefore, the Bao et al. (2020) results are still susceptible to the problem they addressed in Section 3.3. In fact, if Delphi's AAER had not been changed, their machine learning model would not have identified the fraud for the year 2003 or 2004 contributing significantly to the published results.

Re-running the dataset: a simple change

I investigated how the authors' AAER identifier change affected the results. I return the AAER identifiers to their original values by replacing the column *new_p_aaer* with data from the *p_aaer* column in the CSV file. This avoids making any code changes within Matlab. Running their original code on this modified dataset excludes from training the additional firm-years associated exactly with these 17 unique AAER cases, but changes nothing else.

TABLE 2. Three model scenarios

Panel A. Correct cases predicted to be positive			
	(1)	(2)	(3)
Year	Published model	Re-run model	Recoded model
2003	8	7	4
2004	4	4	3
2005	2	2	1
2006	1	1	0
2007	1	1	1
2008	0	0	0
Total	16	15	9
Panel B. Positive predictive values (correct cases / # predicted positive)			
	(1)	(2)	(3)
Year	Published model	Re-run model	Recoded model
2003	13.3 percent	11.7 percent	6.7 percent
2004	6.7 percent	6.7 percent	5.0 percent
2005	3.4 percent	3.4 percent	1.7 percent
2006	1.7 percent	1.7 percent	0.0 percent
2007	1.7 percent	1.7 percent	1.7 percent
2008	0.0 percent	0.0 percent	0.0 percent
Total	4.5 percent	4.2 percent	2.5 percent

The updated results are reported in Table 2. The first column reports the results by year from output files provided by the authors in the "prediciton_rusboost28_2003-2008.zip" file (**link**). Correct cases total 16 for the 2003–2008 out-of-sample test, corresponding to a 4.5 percent positive predictive value, matching the reported values published. Positive predictive value, also known as precision, is calculated as the proportion of correct AAER firm-years out of the cases predicted to experience an AAER. The second column reports the results I obtain when running their original code on their original dataset, showing 15 correct cases corresponding to a 4.2 percent positive predictive value (I'm not sure why it is 15 rather than 16 as in the published paper). The third column reports the results I obtain when running their original code on the dataset with the AAER identifiers replaced by their original values, showing only 9 correct cases corresponding to a 2.5 percent positive predictive value. This value is critical because their published model compared the machine learning result with the result from a parsimonious logit model based on prior literature, which their paper reports to be 2.63 percent for positive predictive value. The updated result shows that the prior model in the literature outperforms this machine learning approach.

Conclusion

The crucial issue in the present critique is to address whether it is appropriate to give new identifiers to the AAER because there is a break in the series resulting from missing data. Since the serial fraud issue concerns the span of the AAER itself and not the sample data, there does not appear to be a logical purpose for the recoding done by the authors. Giving a new AAER identifier to these 17 unique cases out of a total of 413 disproportionately improved their reported results. Without the change, results do not improve upon the prior literature.

Appendix

Data and code related to this research is available from the journal website (**link**).

References

Bao, Yang, Bin Ke, Bin Li, Y. Julia Yu, and Jie Zhang. 2020. Detecting Accounting Fraud in Publicly Traded U.S. Firms Using a Machine Learning Approach. *Journal of Accounting Research* 58(1): 199–235.

Dechow, Patricia M., Weili Ge, Chad R. Larson, and Richard G. Sloan. 2011. Predicting Material Accounting Misstatements. *Contemporary Accounting Research* 28(1): 17–82.

About the Author

Stephen Walker is a Ph.D. candidate in Accounting at UC Berkeley Haas School of Business. Prior to his Ph.D. studies, Stephen worked in sellside equity research for Sanford C. Bernstein in New York City covering the transportation industry, and he trained investment professionals with Training the Street. Stephen was also a small business entrepreneur. He holds an MBA from Columbia Business School and started his career in the finance department with CSX Corporation. He can be reached via his personal website at stephenwalker.me (link).

Bao, Ke, Li, Yu, and Zhang's reply to this article
Go to archive of Comments section
Go to March 2021 issue

Discuss this article at Journaltalk:
https://journaltalk.net/articles/6027/