ARPHA Preprints

PENSOFT

PREPRINT

Author-formatted, not peer-reviewed document posted on 01/06/2021

DOI: https://doi.org/10.3897/arphapreprints.e69590

An Ensemble Model for Financial Statement Fraud Detection

Ahmed M. Khedr, Magdi El Bannany, Sakeena Kanakkayil

ARPHA Preprints Author-formatted, not peer-reviewed document posted on 01/06/2021 DOI: https://doi.org/10.3897/arphapreprints.e69590

Journal of Universal Computer Science, ,
submitted: , accepted: , appeared:

An Ensemble Model for Financial Statement Fraud Detection

Ahmed M. Khedr
(Department of Computer Science, University of Sharjah, UAE
akhedr@sharjah.ac.ae)

Magdi El Bannany
(Department of Accounting, College of Business Administration,
University of Sharjah, UAE
melbannany@sharjah.ac.ae)

Sakeena Kanakkayil
(Department of Computer Science, University of Sharjah, UAE
sakeena.k@gmail.com)

Abstract: Fraudulent financial statements are deliberate furnishing and/or reporting incorrect statistics, and this has become a major economic and social concern as the global market is witnessing an upsurge in financial accounting fraud, costing businesses billions of dollars a year. Identifying companies that manipulate financial statements remains a challenge for auditors, as fraud strategies have become increasingly sophisticated over the years. We evaluate machine learning techniques for financial statement fraud detection, particularly a powerful ensemble technique, the XGBoost algorithm, that help to identify fraud on a set of sample companies drawn from the MENA region. The issue of the class imbalance in the dataset is addressed by applying the SMOTE algorithm. We found that XGBoost algorithm outperformed other algorithms in this study: Logistic Regression (LR), Decision Tree (DT), Vector Machine Support (SVM), Adaboost, and RandomForest. The XGBoost algorithm is then optimised to obtain the optimum performance.

Keywords: Financial Statement Fraud (FSF), Middle East and North Africa (MENA), Machine Learning (ML), XGBoost
Categories: I.2
DOI: /jucs.<SubmissionNumber>

1 Introduction

The Association of Certified Fraud Examiners (ACFE) states that financial statement fraud is the intentional misrepresentation of an enterprise's financial condition by deliberate distortion or omission of the amounts or disclosures in the financial statements to mislead the users of financial statements. According to the Center for Audit Quality (CAQ), individuals or companies are involved in financial statements manipulation for a variety of reasons, including monetary benefits, the need to fulfill short-term financial targets or to cover up unfortunate news. External and internal consumers of the financial statements are constantly questioning the financial statements, and regulatory bodies cannot say with confidence that the financial statements are credible and prepared in compliance with both the regulatory and ethical mandates of the practices of accountants and auditors [1]. Consequently, the detection of fraud or deception in the financial statements is important in order to ensure the authenticity of the financial statements.

This study is of practical importance to businesses and auditors, as the global market is witnessing an upsurge in financial accounting fraud, costing businesses billions of dollars a year. Financial turmoil has a significant impact on a country's businesses, creditors, and as a result, its economy [2, 3]. As an outcome, detection and prediction of financial accounting fraud is becoming an emerging topic for academic studies and industry experts. The objective of this study is therefore to develop a better FSF detection model by utilising data from publicly available financial statements of firms in the MENA region. We selected a powerful ensemble model in ML - the XGBoost algorithm to model our proposed method for a number of reasons. First, although ensemble algorithms have been successfully used in many other fields of research, there is less use in the financial fraud study. Second, the characteristics of XGBoost fit very well with our small dataset, a lot of missing values, class imbalances, etc. It also facilitates tuning a range of hyperparameters to further improvise the efficiency of the model.

We choose expert-defined financial ratios [4] along with some raw financial data for our research. FSF detection models on financial ratios may be more effective, since the ratios determined by domain experts are mostly based on assumptions that provide a strong prediction as to when corporate managers are encouraged to commit fraud [5]. Fernandez-Delgado, Cernadas, Barro, and Amorim [6] demonstrate that there may be no single right model over all data environments; therefore, there is an uncertainty about if the ensemble algorithm will perform better than the conventional financial fraud detection methods in our particular context. To ensure that XGBoost is the best solution to our problem, we have selected five other ML techniques - Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), Adaboost and Random Forest (RF) that are widely used in this area and modelled for their performance analysis.

The rest of the research is organized as follows: Section 2 is dedicated to a brief review of the related researches in the literature. the data and methodology used for the study are discussed in Section 3. Section 4 is reserved for empirical studies and results and Section 5 contains the conclusion and future research.

2 Related Research

Financial Statement Fraud (FSF) detection is not a new area of interest for researchers. It has been under the limelight for the past few decades, and emphasis has been put on accounting anomalies broadly and on financial statements fraud specifically. While in the initial years' researchers made use of statistical or traditional techniques that are time-consuming and expensive, the focus has drifted with the emergence of big data and Machine Learning (ML) [7]. The statistical approaches are centered on traditional mathematical methods, while the methods of ML are focused on modern intelligence. Both categories have many similarities, but the key difference between them is that, while statistical methods are more rigid, the latter methods are able to learn from and adapt to the problem domain [8].

Prior studies have shown greater efficiency of ML approaches over conventional statistical approaches [9]. Therefore, our review of past research is limited to papers that have used only ML techniques for FSF detection. Most of the research in previous literature formulated the detection of FSF as a binary classification problem, some as a multi-class classification, while others as a clustering problem. Researchers have conducted both quantitative and qualitative FSF analysis. Text mining has been used extensively for qualitative research. We focus on papers that perform quantitative analysis using ML techniques.

In the initial stages, researchers mainly included Neural Network (NN), Linear regression (LR), Decision Tree (DT), Support Vector machine (SVM), Discriminant Analysis (DA) and Bayesian Belief Network. Supervised learning techniques were selected for analysis more than unsupervised ones, with 65% of the articles from US, China, Taiwan and Spain [10]. A considerable number of studies that analysed the performance of classifiers on FSF detection have shown that SVM [11, 12, 13, 14, 15], NN [16, 17, 18, 19], DT [20, 21] perform well in FSF detection/prediction.

In recent years, ensemble ML techniques have been used in studies and have mostly outperformed single classifiers. Ensemble classifiers integrate the predictions of multiple base models. Numerous empirical and theoretical findings have shown that model combinations improve predictive accuracy [22]. They are also well known for their capability to reduce bias and variance. Many researchers have shown interest in studying ensembles with boosting [23], bagging [24, 25, 26, 27] and other hybrid methods [28] on both balanced and unbalanced data. It was seen that the performance of the models was dependent on the base classifiers selected. An illustration of the papers reviewed is given in Table 1.

Prior research shows that ensemble classifiers are the best in detecting FSF, but there is less research with ensembles compared to single classifiers. Most of the studies have used imbalance data sets for evaluation, as it is the case with real-world data. Consequently, because of the common problem of class imbalances, traditional ensemble models must generally be coupled with sampling techniques such as over-sampling or under-sampling for balancing the class distribution with only a few studies considering the imbalance issue while modeling. While most of the researchers have taken financial ratios for prediction, some argued that raw variables have produced better results. Various metrics are often used to assess classifier performance, but the prevalent ones are sensitivity or recall, precision, and accuracy. In this paper, we evaluate the different classifiers that can be used in the FSF detection along with the class imbalance issue, taking into account both raw financial variables and financial ratios.

3 Data and Methodology

3.1 Data

Our experimental data set includes 950 companies in the MENA region. All of the companies selected come from different sectors, including manufacturing, technology, energy, telecommunications, real estate and insurance. Data is collected from the global company database-Osiris[1]. Based on the availability of the data, we selected two consecutive years from 2012 to 2019 for each company. There are 102 fraudulent years and 1798 non-fraudulent years. The financial indicators are taken from the respective companies' financial statements and balance sheets. The details of the 26 financial attributes including financial attributes from Beneish model [4] used are given in Table 2. All attributes are quantitative, with the target value being discrete and the others being continuous.

Professor Messod Beneish, in June 1999, published his study ”The Detection of Earnings Manipulation” in which he argued that high sales growth, declining gross

[1] Source: https://www.bvdinfo.com/en-gb/our-products/data/international/Osiris

Year and reference number	Methods	Data source (Fraud: non-fraud)	Input features	Best model (performance in %)
2010 [11]	Probit, NN, LR, SVM	AAER (205:6427)	23 raw variables	SVM (AUC – 87.8)
2011 [3]	LR, LDA, C4.5, MLP, RBF, SVM	Taiwan Stock Exchange (25:50)	15 financial ratios + 3 raw variables	SVM (Acc - 92)
2011 [10]	SVM, NB, KNN	McGreggor-BFA (123:2888)	14 financial ratios	SVM (Acc-95.9)
2019 [20]	SVM, CART, NN, LR, NB, KNN	Shanghai and Shenzhen Stock exchanges (134:402)	17 financial ratios + 7 non-financial variables	SVM(Acc-81.88)
2011 [23]	LR, SVM, GP, NN	Chinese Stock Exchange (1:1)	28 financial ratios + 7 raw variables	NN (AUC-98), GP with feature selection (AUC-92.9)
2015 [19]	LR, DT, NN	Taiwan and China sources (129:447)	3 financial ratios + 21 other factors	ANN(Acc-92.8)
2016 [25]	DT, BBN, SVM, NN	Taiwan's listed and OTC companies (44:132)	21 financial ratios + 2 raw variables + 7 non-financial variables	DT (Acc-87.97)
2012 [4]	Probit regression, Logit regression, SGB, RF, Rule ensemble	AAER (114:114)	12 financial ratios + 1 variable	RF (AUC-90.1)
2017 [14]	LR, BN, DT, SVM, NN, Bagging, RF, Adaboost	AAER (311:311)	24 financial variables +8 other variables	RF (TP-86.93)
2014 [6]	Logit regression, DT, NN, SVM, Ensemble of LR, DT, NN and SVM	Shanghai and Shenzhen stock exchanges (110:440)	23 financial ratios	Ensemble (Acc-88.9)
2018 [13]	SVM, RF, DT, ANN, LR	China Securities Regulatory Commission (120:120)	17 financial variables+5 non-financial	RF (Acc-75)
2019 [34]	LR, SVM, RUSBoost, Adaboost	AAER (1171:204855)	28 raw financial variables	RUSBoost (AUC-72.5)

Table 1: Comparison of studies on FSF with ML Techniques

Table 2: Financial attributes

Features	Description
a1	accounts receivable
a2	sales
a3	Cost of Goods Sold (COGS)
a4	current asset
a5	fixed assets
a6	total assets
a7	depreciation
a8	general and adminstrative expenses
a9	long term debt
a10	total current liabilities
a11	change in current assets
a12	change in cash
a13	change in current liabilities
a14	change in income tax payable
a15	current maturity of long term debt
a16	current maturity of LTD
a17	change in current maturity of long term debt
a18	amortization
a19	Days' Sales in Receivables Index (DSRI)
a20	Gross Margin Index (GMI)
a21	Asset Quality Index (AQI)
a22	Sales Growth Index (SGI)
a23	Depreciation (DEPI)
a24	Sales, General and Administrative Expenses (SGAI)
a25	Leverage Index (LVGI)
a26	Total Accruals to Total Assets (TATA)

ARPHA Preprints Author-formatted, not peer-reviewed document posted on 01/06/2021 DOI: https://doi.org/10.3897/arphapreprints.e69590

margins, soaring operating expenses, and increasing leverage encourage companies to manipulate profits. They'll most probably alter profits by speeding up sales recognition, rising accruals and cost deferrals, and minimizing depreciation. We have included those attributes in our study.

3.2 Methodology

This section describes the method implemented to conduct the study. A schematic representation of the stages involved throughout the study is shown in Figure 1. The first phase is the data pre-processing phase followed by the classifiers modeling phase and the final stage is the optimization phase.

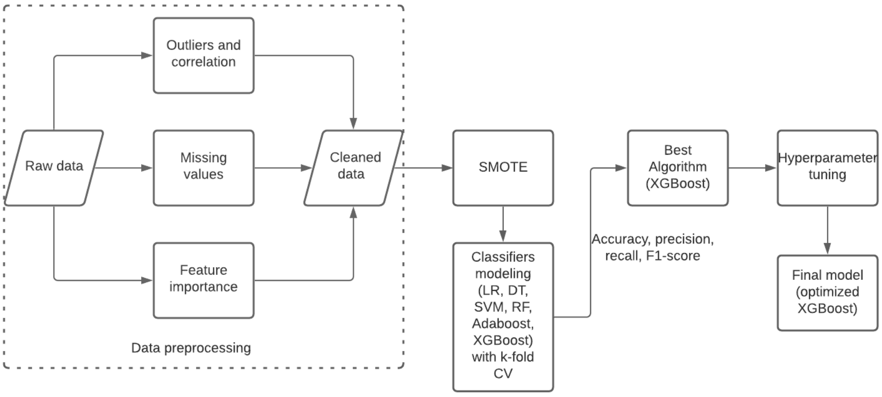

Figure 1: Stages involved in the study

Our dataset is limited as there are not many publicly available companies in the MENA region. The missing values in the dataset are replaced by mean or 0. A detailed descriptive statistics is obtained and the outliers in the data set are detected and excluded. Feature Importance is calculated by calculating the Gini importance value for each feature. Each feature is then sorted in descending order and the top k features are selected. Top features are highly linked to the target variable. We found that keeping all attributes yielded better results than avoiding the least important ones. As our data is imbalanced and small, we are balancing it with synthetic minority data by oversampling [29], that has been extensively and successfully used in the literature for similar datasets [30].

3.2.1 Synthetic Minority Oversampling TEchnique (SMOTE)

In the case of fraud detection problems, the minority class needs to be given special consideration as it defines the phenomena which we aim to anticipate from a multitude of majority class structures that reflect correct processes. The performance of the standard classifiers is biassed towards the majority class as they are programmed to minimise the overall inaccuracy of classification regardless of the distribution of classes. This bias problem can be overcome by excluding some examples of the majority class, known as under-sampling, or by including new examples of the minority class, known as over-sampling. As we have a small dataset, we have chosen the latter one.

An effective over-sampling technique for producing new examples is SMOTE [29] which can be implemented independently of the classifier being used. This algorithm addresses the challenge of over-fitting caused by random over-sampling. It relies on the feature space to create new instances with the aid of interpolation between positive instances that lie together. SMOTE starts by finding examples near the feature space, connecting the dots between the examples, and drawing a new instance at a point along that line [31]. In particular, a random sample of the minority class is chosen first. Then for this sample, k of the nearest neighbours is found (usually k=5). Choose a randomly selected neighbour and construct a synthetic example at a randomly selected point in between two examples in the feature space.

It is evident from prior studies that SVM, NN, DT, LR have performed well in the detection of fraud. For comparison with XGBoost, we selected SVM, DT and LR. NN was not considered as it demands a large data set for training. From the ensemble methods, we selected random forest and Adaboost algorithms, as decision-tree based algorithms are considered best for small to medium-sized data. Model training and testing of the comparison-phase algorithms is done with the help of the Scikit-learn packages from Python.

After the comparison phase, it was clear that XGBoost, a tree-based algorithm, is the best one for our dataset. In the next stage, we optimised the algorithm to obtain an optimal hyperparameter combination with the help of RandomizedSearchCV from Scikit-learn, which performs a randomised search of hyperparameters to further enhance the performance of the algorithm. The estimator parameters used to implement these methods are optimised by cross-validating the parameter settings search.

3.2.2 Base classifiers - SVM, DT, and LR

3.2.3 SVM

A Support Vector Machine (SVM) [32] is a discriminative classifier typically explained as a separating hyperplane.To put it another way, the algorithm generates an optimal hyperplane that classifies new instances, given labelled training data (supervised learning). The data points or vectors nearest to the hyperplane that influence the direction of the hyperplane are referred to as the Support Vector. Since these vectors support a hyperplane, it is called a support vector. SVM has high predictive accuracy and generalisation capabilities, particularly for small, non-linear and high-dimensional samples [25]. We use the linear SVM for our problem.

3.2.4 DT

Decision trees are among the most successful machine learning algorithms given their intelligibility and clarity [33]. A Decision Tree (DT) is used for estimation, clustering, and classification tasks. At first, the entire dataset is put at the root node. The best attribute is put at the root node. The training dataset is then separated into subsets such that each subset includes data with the same value as the root node attribute. A new node is created by each branch. This process is replicated until the nodes of the leaf are located at each branch or the full depth is reached. The knowledge gain is used to determine the best attribute to be split at each stage of tree building. The attribute with the maximum knowledge gain is considered as the selection attribute for each node. The leaf nodes in the decision tree reflect the class and the decision nodes determine the rules. The test data class is predicted by the decision rules. The key benefits of Decision Trees are that this method offers a meaningful model to describe the acquired knowledge and thus enables the extraction of IF–THEN classification rules [17].

3.2.5 LR

The logistic regression as a general statistical model was originally developed and popularized primarily by Joseph Berkson [34]. Logistic regression is to be conducted when the dependent variable is binary. It is a discriminative classifier which is linear in its parameters that is used to describe the relationship between a single dependent binary variable and one or more independent variable. Logistic Regression (LR) can handle both nominal and numerical data. It predicts the likelihood of a binary response based on one or more predictor attributes [35].

3.2.6 Ensembles - RF, Adaboost, and XGBoost

Predictions from previously developed individual base estimators are integrated using ensemble strategies to improve robustness/generalization over one estimator and deliver better results. Even when the individual models in the ensembles are fairly simple, the power of ensembling allows us to create strong ensemble models.

3.2.7 RF

Random Forest (RF) is a bagging ensemble technique proposed particularly for trees [36]. The base model of RF is the decision tree, which seeks to minimise the variance of the DT model. Random subsets of the data are generated with replacement, and each subset is trained with the help of a decision tree. While expanding the trees, Random Forest adds more randomness to the structure. Rather than looking for the most significant feature when dividing a node, it looks for the best feature across a random subset of features [37]. As a result, there is a wide range of diversity, which contributes to a successful model in general. As a result, in random forest, the algorithm only considers a random subset of the features when dividing a node. Trees can also be turned further random by using additional random thresholds for every feature instead of looking for the highest suitable thresholds (like a normal decision tree does).

3.2.8 Adaboost

Adaboost or Adaptive boosting was the first really successful boosting algorithm developed for binary classification [38]. It is an approach to minimise the error of a weak learning algorithm. Theoretically, a weak learning algorithm could be any one as long as it can produce classifiers that need to be a bit more consistent than random guessing [39]. Adaboost helps to combine multiple “weak classifiers” into a single “strong classifier” [40]. The most common algorithm used with AdaBoost are decision trees. A weak classifier (decision stump) is prepared using the weighted samples on the training data. Only binary (two-class) classification problems are supported, so each decision stump makes a decision on one input variable and outputs a value of +1 or -1 for the first or second class. Weak models are sequentially added, trained with weighted training data. The method progresses until a pre-determined number of weak learners has been generated or no more improvements can be achieved on the training data set.

3.2.9 XGBoost

XGBoost or eXtreme Gradient Boosting is a tree-based algorithm [41]. XGBoost has proved its prowess in terms of performance and speed. Boosting is an ensemble strategy with the key goal of reducing bias and variance. The aim is to sequentially build weak trees so that each new tree (or learner) works on the flaw (misclassified data) of the preceding tree. The data weights are re-adjusted, known as ”re-weighting,” once a weak learner is added. Because of the auto-correction after every new learner is introduced, the whole forms a strong model after convergence. The loss function of the model is characterized as penalizing the complexity of the model with regularization in order to decrease the possibility of overfitting. The technique performs well even with missing values or a lot of zero values with an understanding of the sparsity. XGBoost uses an algorithm called the ”weighted quantile sketch algorithm,” which facilitates the classifier to concentrate on data that is incorrectly classified. The aim of each new learner is to learn how to classify the incorrect data with each iteration.

4 Implementation and analysis

4.1 Implementation

We have used Python 3.8 for implementation. A detailed descriptive statistics is obtained using the pandas_profiling in Python. The outliers in the dataset are detected with the help of IsolationForest, while the most important features are enumerated using ExtraTreesClassifier of sklearn. SMOTE is implemented using the imblearn package with k_neighbors=5. All the models are implemented using Scikit Learn library and evaluated using 10-fold cross-validation. The descriptive statistics of the attributes are given in Table 3.

All the classifiers are modelled with the help of sklearn. LR is implemented using the LogisticRegression method; DT using the DecisionTreeClassifier method with max_depth = 4 and criterion set to ’entropy’ and SVM using LinearSVC. In the case of ensemble classifiers, RF is modelled using RandomForestClassifier with n_estimators=100; Adaboost using AdaBoostClassifier with sigmoid kernel SVC as the base estimator and default n_estimators and learning_rate. Finally, XGBoost is modelled using XGBClassifier() with default parameters and then the hyperparameters are fine-tuned in the subsequent phase.

4.2 Performance measures

Predictive performance of data mining classifiers is measured in terms of accuracy, precision, recall and F1-score, the common evaluation metrics of machine learning. Testing accuracy and F1-score measures are used for performance evaluation. A k-fold cross-validation score, with k set to 10 is used for all the models (XGBoost has inbuilt CV). Accuracy is the ratio of number of correct predictions to the total number of input samples. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall or sensitivity is the ratio of correctly predicted positive observations to all the observations in actual class. F1 Score is the weighted average of Precision and Recall.

ARPHA Preprints *Author-formatted, not peer-reviewed document posted on 01/06/2021* DOI: https://doi.org/10.3897/arphapreprints.e69590

Table 3: Descriptive statistics of all financial attributes

	Mean	Min	Max
a1	81634.16	0.22	6720657.5
a2	400980.45	14.08	45100892.0
a3	284715.12	24.85	25907758.0
a4	295165.66	38.93	28690088.0
a5	530127.73	5.63	57441212.0
a6	825293.40	139.85	85256240.0
a7	26143.20	1.01	4281863.0
a8	16282.29	0.17	2756804.75
a9	240527.33	0.67	13707662.0
a10	230428.77	6.25	10101471.0
a11	6918.41	-3557134.5	5104900.0
a12	-4777.54	-4386089.0	2802615.06
a13	5804.93	-4728200.0	5758000.0
a14	271.20	-721997.27	603964.31
a15	75383.46	5.31	2549428.75
a16	251655.17	6.25	11078738.75
a17	3643.33	-4814336.46	5844136.46
a18	9086.84	-19970.17	577853.06
a19	1.2048	0.0025	63.4548
a20	0.9727	-15.667	13.038
a21	1.0666	-51.018	232.397
a22	1.06069	0.12715	6.3276
a23	1.26359	0.00282	52.7857
a24	1.6997	0.0047	188.678
a25	1.0792	0.0675	15.3626
a26	-0.4812	-6.6897	1.3688

$$Accuracy = \frac{True\ Positive + True\ Negative}{(True\ Positive + True\ negative + False\ positive + False\ Negative)}$$
$$= \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (1)$$

$$Precision = \frac{True\ Positive}{(True\ Positive + False\ Positive)} = \frac{True\ Positive}{(Total\ Predicted\ Positive)} \quad (2)$$

$$Recall = \frac{True\ Positive}{(True\ Positive + False\ Negative)} = \frac{True\ Positive}{(Total\ Actual\ Positive)} \quad (3)$$

$$F1 - score = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (4)$$

4.3 Analysis

Accuracy of any ML algorithm is highly dependent on the problem, and the integrity and complexity of the training dataset. The prediction performance of all the six models on SMOTE applied dataset are listed in Table 4 and their graphical representation is displayed in Figure 3.

Table 4: Prediction results of classifiers after SMOTE

Classifier	Mean Accuracy	Precision	Recall	Mean F1-score
LR	0.7388	**0.8055**	0.8344	0.8196
DT	0.8222	0.4111	0.5000	0.4513
SVM	0.8888	0.8034	0.8411	0.8218
RF	0.8055	0.4027	0.5000	0.4461
AdaBoost	0.8333	0.4166	0.5000	0.4545
XGBoost	**0.9366**	0.7938	**0.8637**	**0.8272**

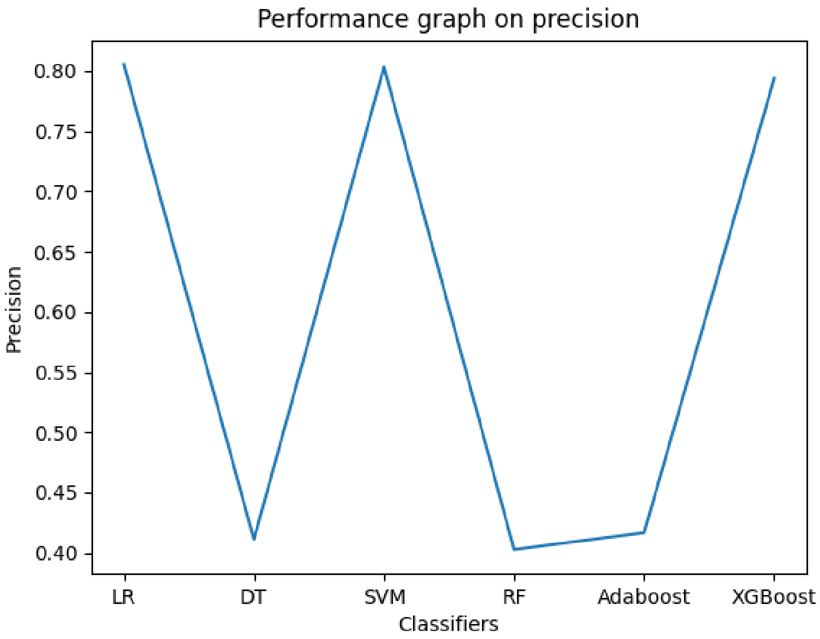

Figure 2: Evaluation results for the classifiers on precision

The highest mean accuracy rate was given by XGBoost algorithm followed by SVM and then Adaboost, in figure **??**. LR had a low accuracy rate compared to others while it gave a high F1-score of 0.8196 as seen in figure 4. SVM displayed an accuracy rate of 0.8888, second to XGBoost, and from figures **??** we can see that it also has a good performance on the basis of precision, recall and F1-score. DT, RF and Adaboost gave an average performance on all the 4 metrics.

It is evident that XGBoost delivers consistent performance on all four metrics with the highest mean accuracy and F1-score on our SMOTE applied MENA dataset. The dataset is split into training and testing set with test_size= 0.3. SVM also performed better but is not upto the XGBoost algorithm. Accuracy and F1-score are obtained after a k-fold cross validation on the training data with k set to 10.

4.4 XGBoost optimization

Based on preliminary observations, XGBoost is the best model for the detection of fraud in the financial statements in our dataset. Next, we further optimised the performance of XGBoost with the help of hyperparameter tuning using RandomizedSearchCV on accuracy scores with n_iter=1000 and 3-fold cross validation:

’learning_rate’: [0.03, 0.01, 0.003, 0.001],
’min_child_weight’: [1,3, 5,7, 10],
’gamma’: [0, 0.5, 1, 1.5, 2, 2.5, 5],

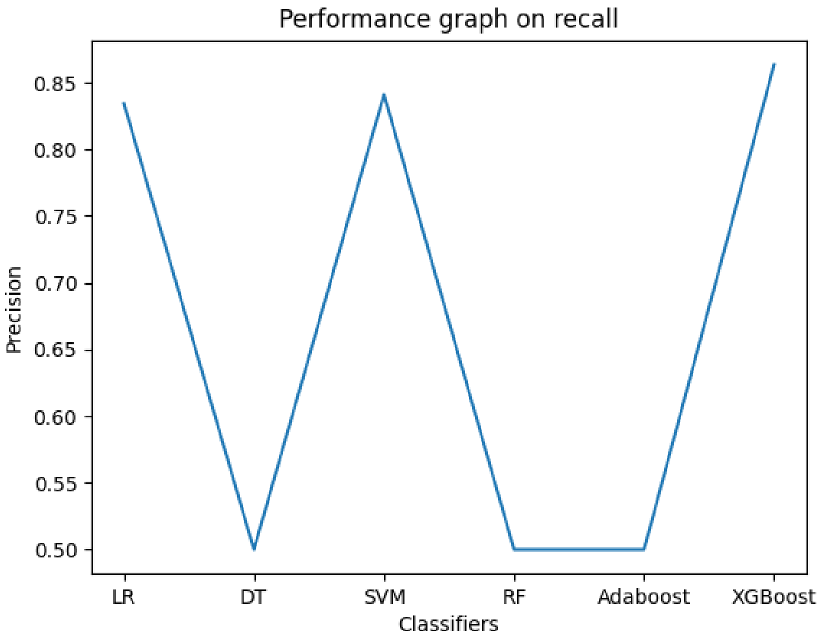

Figure 3: Evaluation results for the classifiers on recall

'subsample': [0.6, 0.8, 1.0, 1.2, 1.4],
'colsample_bytree': [0.6, 0.8, 1.0, 1.2, 1.4],
'max_depth': [3, 4, 5, 6, 7, 8, 9 ,10, 12, 14],
'reg_lambda':[0.4, 0.6, 0.8, 1, 1.2, 1.4]

All possible parameter combinations are run and the model is trained until validation_0-error has improved in 10 rounds. The fitting is achieved by 3 folds for each of 1000 candidates, totalling 3000 folds. The best iteration for each round is the one with the least validation error. Below is the list of the best parameters found:

'learning_rate': 0.03,
'min_child_weight': 3,
'gamma': 1.5,
'subsample': 0.8,
'colsample_bytree': 1.0,
'max_depth': 9,
'reg_lambda': 1

The best accuracy score across all the parameter combinations for XGBoost algorithm on our SMOTE sampled MENA dataset is **0.9605** which is a significant improvement on the accuracy score of 0.9366 in the previous stage.

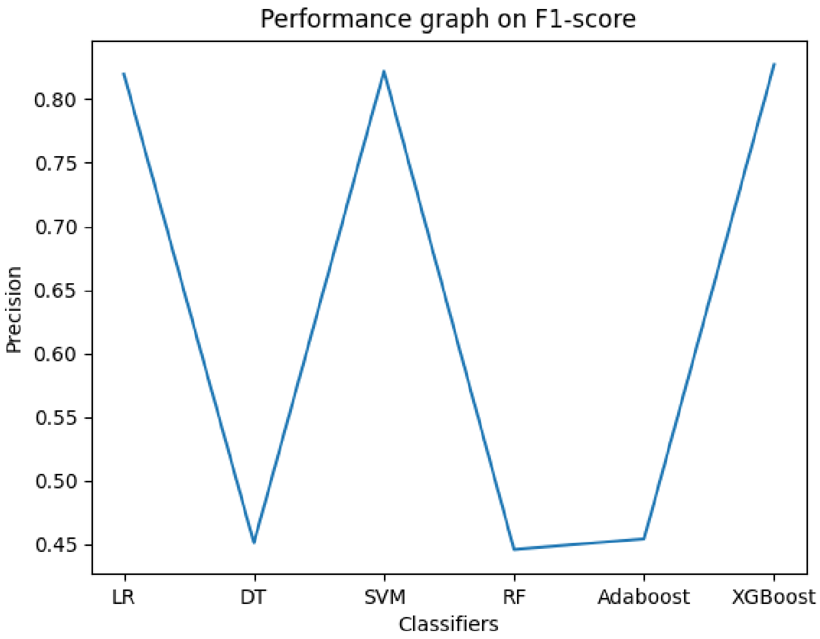

Figure 4: Evaluation results for the classifiers on F1-score

5 Conclusion

Finding the best-performing financial statement fraud detection model has always been an important subject of study, and several FSF detection and prediction models have been developed since then. In this paper, we conducted an analysis and comparison of three individual machine learning classifiers and three ensemble techniques used widely in FSF detection using a dataset comprising companies from the MENA region. We also up-sampled our dataset using the SMOTE technique to prevent class imbalance issues. We used SVM, DT and LR as individual classifiers and RF, Adaboost and XGBoost as ensemble techniques. While all the classifier models yielded an acceptable accuracy rate, the simulation results indicate that the XGBoost classifier is the most accurate model for financial statement fraud detection in our settings. In the later phase of this study, XGBoost classifier is further optimised by hyperparameter tuning with cross validation to get the best model for our problem. The simulation results also indicate that the proposed model has higher performance compared to classic ML models and ensemble models. In our study, we have considered only financial attributes. Non-financial attributes may also be incorporated in the future.

References

1. LI Kulikova and DR Satdarova. Internal control and compliance-control as effective methods of management, detection and prevention of financial statement fraud. Academy of Strategic

ARPHA Preprints Author-formatted, not peer-reviewed document posted on 01/06/2021 DOI: https://doi.org/10.3897/arphapreprints.e69590

Management Journal, 15:92, 2016.

2. Meenu Sreedharan, Ahmed M Khedr, and Magdi El Bannany. A multi-layer perceptron approach to financial distress prediction with genetic algorithm. Automatic Control and Computer Sciences, 54(6):475–482, 2020.

3. Magdi El-Bannany, Meenu Sreedharan, and Ahmed M Khedr. A robust deep learning model for financial distress prediction.

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

5. Yang Bao, Bin Ke, Bin Li, Y Julia Yu, and Jie Zhang. Detecting accounting fraud in publicly traded us firms using a machine learning approach. Journal of Accounting Research, 58(1):199–235, 2020.

6. Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? The journal of machine learning research, 15(1):3133–3181, 2014.

7. Mahmood Mohammadi, Shohreh Yazdani, Mohammad Hamed Khanmohammadi, and Keyhan Maham. Financial reporting fraud detection: An analysis of data mining algorithms. International Journal of Finance & Managerial Accounting, 4(16):1–12, 2020.

8. Sean L Humpherys, Kevin C Moffitt, Mary B Burns, Judee K Burgoon, and William F Felix. Identification of fraudulent financial statements using linguistic credibility analysis. Decision Support Systems, 50(3):585–594, 2011.

9. Jarrod West, Maumita Bhattacharya, and Rafiqul Islam. Intelligent financial fraud detection practices: an investigation. In International Conference on Security and Privacy in Communication Networks, pages 186–203. Springer, 2014.

10. Mousa Albashrawi. Detecting financial fraud using data mining techniques: A decade review from 2004 to 2015. Journal of Data Science, 14(3):553–569, 2016.

11. Mark Cecchini, Haldun Aytug, Gary J Koehler, and Praveen Pathak. Detecting management fraud in public companies. Management Science, 56(7):1146–1160, 2010.

12. Ping-Feng Pai, Ming-Fu Hsu, and Ming-Chieh Wang. A support vector machine-based model for detecting top management fraud. Knowledge-Based Systems, 24(2):314–321, 2011.

13. Johan Perols. Financial statement fraud detection: An analysis of statistical and machine learning algorithms. Auditing: A Journal of Practice & Theory, 30(2):19–50, 2011.

14. Stephen O Moepya, Sharat S Akhoury, and Fulufhelo V Nelwamondo. cost-sensitive classification for financial fraud detection under high class-imbalance. In 2014 IEEE international conference on data mining workshop, pages 183–192. IEEE, 2014.

15. Jianrong Yao, Yanqin Pan, Shuiqing Yang, Yuangao Chen, and Yixiao Li. Detecting fraudulent financial statements for the sustainable development of the socio-economy in china: a multi-analytic approach. Sustainability, 11(6):1579, 2019.

16. Dan Han. Researches of detection of fraudulent financial statements based on data mining. Journal of Computational and Theoretical Nanoscience, 14(1):32–36, 2017.

17. Chi-Chen Lin, An-An Chiu, Shaio Yan Huang, and David C Yen. Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments. Knowledge-Based Systems, 89:459–470, 2015.

18. Pediredla Ravisankar, Vadlamani Ravi, G Raghava Rao, and Indranil Bose. Detection of financial statement fraud and feature selection using data mining techniques. Decision support systems, 50(2):491–500, 2011.

19. Adila Afifah Rizki, Isti Surjandari, and Reggia Aldiana Wayasti. Data mining application to detect financial fraud in indonesia's public companies. In 2017 3rd International Conference on Science in Information Technology (ICSITech), pages 206–211. IEEE, 2017.

20. Rajan Gupta and Nasib Singh Gill. Prevention and detection of financial statement fraud–an implementation of data mining framework. Editorial Preface, 3(8):150–160, 2012.

21. Suduan Chen. Detection of fraudulent financial statements using the hybrid data mining approach. SpringerPlus, 5(1):1–16, 2016.

22. J Bertomeu, E Cheynel, E Floyd, and W Pan. Ghost in the machine: Using machine learning to uncover hidden misstatements.

23. Yang Bao, Bin Ke, Bin Li, Y Julia Yu, and Jie Zhang. Detecting accounting fraud in publicly traded us firms using a machine learning approach. Journal of Accounting Research, 58(1):199–235, 2020.

24. David G Whiting, James V Hansen, James B McDonald, Conan Albrecht, and W Steve Albrecht. Machine learning methods for detecting patterns of management fraud. Computational Intelligence, 28(4):505–527, 2012.

25. Xin-Ping Song, Zhi-Hua Hu, Jian-Guo Du, and Zhao-Han Sheng. Application of machine learning methods to risk assessment of financial statement fraud: evidence from china. Journal of Forecasting, 33(8):611–626, 2014.

26. Jianrong Yao, Jie Zhang, and Lu Wang. A financial statement fraud detection model based on hybrid data mining methods. In 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), pages 57–61. IEEE, 2018.

27. Petr Hajek and Roberto Henriques. Mining corporate annual reports for intelligent detection of financial statement fraud–a comparative study of machine learning methods. Knowledge-Based Systems, 128:139–152, 2017.

28. Haibing Li and Man-Leung Wong. Financial fraud detection by using grammar-based multi-objective genetic programming with ensemble learning. In 2015 IEEE Congress on Evolutionary Computation (CEC), pages 1113–1120. IEEE, 2015.

29. Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16:321–357, 2002.

30. Dina Elreedy and Amir F Atiya. A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance. Information Sciences, 505:32–64, 2019.

31. Satwik Mishra. Handling imbalanced data: Smote vs. random undersampling. International Research Journal of Engineering and Technology (IRJET), 4(8), 2017.

32. Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine learning, 20(3):273–297, 1995.

33. Xindong Wu, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. Top 10 algorithms in data mining. Knowledge and information systems, 14(1):1–37, 2008.

34. Jan Salomon Cramer. The origins of logistic regression. 2002.

35. Kuldeep Randhawa, Chu Kiong Loo, Manjeevan Seera, Chee Peng Lim, and Asoke K Nandi. Credit card fraud detection using adaboost and majority voting. IEEE access, 6:14277–14284, 2018.

36. Tin Kam Ho. Random decision forests. In Proceedings of 3rd international conference on document analysis and recognition, volume 1, pages 278–282. IEEE, 1995.

37. Weiwei Lin, Ziming Wu, Longxin Lin, Angzhan Wen, and Jin Li. An ensemble random forest algorithm for insurance big data analysis. Ieee access, 5:16568–16575, 2017.

38. Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1):119–139, 1997.

39. Jie Sun, Ming-yue Jia, and Hui Li. Adaboost ensemble for financial distress prediction: An empirical comparison with data from chinese listed companies. Expert systems with applications, 38(8):9305–9312, 2011.

40. Meenu Sreedharan, Ahmed M Khedr, and Magdi El Bannany. A comparative analysis of machine learning classifiers and ensemble techniques in financial distress prediction. In 2020 17th International Multi-Conference on Systems, Signals & Devices (SSD), pages 653–657. IEEE, 2020.

41. Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794, 2016.