applied sciences

MDPI

Article

A Powerful Predicting Model for Financial Statement Fraud Based on Optimized XGBoost Ensemble Learning Technique

Amal Al Ali [1], Ahmed M. Khedr [2,*], Magdi El-Bannany [3,4] and Sakeena Kanakkayil [2]

[1] Department of Information Systems, University of Sharjah, Sharjah 27272, United Arab Emirates
[2] Department of Computer Science, University of Sharjah, Sharjah 27272, United Arab Emirates
[3] College of Business Administration, Umm Al Quwain University, Umm Al Quwain 536, United Arab Emirates
[4] Department of Accounting and Auditing, Faculty of Business, Ain Shams University, Cairo 11566, Egypt
* Correspondence: akhedr@sharjah.ac.ae

Abstract: This study aims to develop a better Financial Statement Fraud (FSF) detection model by utilizing data from publicly available financial statements of firms in the MENA region. We develop an FSF model using a powerful ensemble technique, the XGBoost (eXtreme Gradient Boosting) algorithm, that helps to identify fraud in a set of sample companies drawn from the Middle East and North Africa (MENA) region. The issue of class imbalance in the dataset is addressed by applying the Synthetic Minority Oversampling Technique (SMOTE) algorithm. We use different Machine Learning techniques in Python to predict FSF, and our empirical findings show that the XGBoost algorithm outperformed the other algorithms in this study, namely, Logistic Regression (LR), Decision Tree (DT), Support Vector Machine (SVM), AdaBoost, and Random Forest (RF). We then optimize the XGBoost algorithm to obtain the best result, with a final accuracy of 96.05% in the detection of FSF.

Keywords: financial statement fraud (FSF); fraud detection; ensemble approach; Middle East and North Africa (MENA); machine learning (ML); XGBoost (eXtreme Gradient Boosting); Synthetic Minority Oversampling Technique (SMOTE)

Citation: Ali, A.A.; Khedr, A.M.; El-Bannany, M.; Kanakkayil, S. A Powerful Predicting Model for Financial Statement Fraud Based on Optimized XGBoost Ensemble Learning Technique. *Appl. Sci.* **2023**, *13*, 2272. https://doi.org/10.3390/app13042272

Academic Editor: Amelia Zafra

Received: 21 January 2023
Revised: 6 February 2023
Accepted: 8 February 2023
Published: 10 February 2023

1. Introduction

Financial Statement Fraud (FSF) is a global concern. FSF is characterized as major omissions or false representations in financial statements caused by a deliberate failure to report financial data in conformity with generally accepted accounting standards. FSF can cause significant impacts on stakeholders of both fraudulent enterprises and non-fraudulent firms if it is not recognized and prevented in a timely manner [1]. Unfortunately, FSF is not easy to spot. Furthermore, even when discovered, significant damage has generally occurred already. As a result, regulators, auditors, and investors would benefit greatly from more efficient and effective techniques that can detect FSF. 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 statement 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 financial statements are constantly questioning financial statements, and regulatory bodies cannot say with confidence that financial statements are credible and prepared in compliance with the regulatory and ethical mandates of the practices of accountants and auditors [2,3]. Consequently, the detection of fraud or deception is important in order to ensure the authenticity of financial statements. In this context, the present study is of practical importance to businesses and auditors, as the global market is witnessing an upsurge in financial accounting fraud that costs businesses billions

of dollars a year. Financial turmoil has a significant impact on a country's businesses and creditors, and consequently on its economy [1,4,5]. As an outcome, the detection and prediction of financial accounting fraud are becoming an emerging topic for academic studies and industry experts.

Motivation and Contribution

Globally, the cost of FSF has continued to rise over the years. The most direct victims are the investors and financiers who supply funds to firms under false pretenses. There are greater costs to society at the macro-level, such as loss of trust in financial systems, even though the economic impact at the company level can be highly variable. Because FSF causes significant property harm to investors, stakeholders, and society, a great number of studies have been conducted. Most of the works in the literature on FSF apply traditional regression analysis; see, for instance, Andrew and Robin [6–10].

In recent years, a number of experts and researchers have attempted to use machine learning (ML) and data mining methods to carry out research in this field as a way of reducing detection errors. While many business strategies are based on the accuracy of financial statements, there are not enough resources to analyze all of them thoroughly. As auditors have been found culpable in multiple examples of FSF, credible financial fraud detection models should be offered in an easy-to-use manner to auditors, investors, regulators, and other stakeholders. Because of the inherent reliance on limited distributional assumptions, parametric techniques such as LR lack the general applicability that non-parametric approaches may provide [11]. Existing research on quantitative methods of financial fraud detection, on the other hand, is primarily focused on the banking and financial services industry, mostly on the detection of insurance and credit card fraud. Quantitative techniques to identify and discourage FSF should receive significant attention in respected academic journals. The current scientific and academic literature currently seeks further rules or classifications from previous data to achieve the purpose of prediction or detection. Machine learning (ML) with an appropriate amount of data can yield more accurate prediction and classification outcomes compared to the traditional approach.

Contribution: This study aims to establish a superior model for financial fraud prediction using a powerful ensemble technique, the XGBoost (eXtreme Gradient Boosting) algorithm, to help identify fraud on a set of sample companies drawn from the MENA region by spotting the early signs. This approach can help to mitigate losses to investors, auditors, and all stakeholders in the financial market. This research enables researchers and practitioners to gain a deeper understanding and make informed decisions based on a financial statement fraud detection model. We focus on utilizing data from publicly available financial statements of firms in the Middle East and North Africa (MENA) region. We selected a powerful ML ensemble model, namely, the XGBoost algorithm, to model our proposed method for a number of reasons. Ensemble algorithms have been successfully used in many fields of research [12], though they have been less utilized in financial fraud studies.

The characteristics of XGBoost fit very well with our small dataset, which is characterized by many missing values and high class imbalance. XGBoost facilitates the tuning of a range of hyperparameters in order to further improve the efficiency of the model. Because of the high class imbalance in this one-of-a-kind domain, the samples selected in past studies have a tendency to be processed less realistically. XGBoost is able to effectively learn from imbalanced data with the help of Synthetic Minority Oversampling Technique (SMOTE) sampling. We chose expert-defined financial ratios [13] along with raw financial data for our research. FSF detection models based on financial ratios may be more effective, as the ratios determined by domain experts are mostly based on assumptions that provide a strong prediction of situations in which corporate managers are encouraged to commit fraud [14]. Fernandez-Delgado, Cernadas, Barro, and Amorim [15] demonstrate that there may be no single right model that can be applied to all data environments; therefore, there is uncertainty about whether ensemble algorithms perform better than conventional finan-

cial fraud detection methods in our particular context. Hence, we selected five other ML techniques that are widely used in this area and modeled them for performance analysis and comparison with our model: Logistic Regression (LR), Support Vector Machine (SVM), Decision Tree (DT), AdaBoost, and Random Forest (RF). We think that the results of this study can be useful to investors, regulators, stock exchanges, boards of directors, external auditors, and other key stakeholders as they seek to prevent, deter, and detect fraudulent financial reporting. Benford's Law is an effective method and analytical technique to help detect accounting fraud; see, for instance, [16–18]. Benford's Law is a mathematical tool used to determine whether investigated financial statements contain unintentional errors or fraud. When using Benford's law, counterfeit numbers have a slightly different pattern than valid or random samples.

The rest of this research is organized as follows: Section 2 is dedicated to a brief review of the related research 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; finally, Section 5 includes our conclusions and future research directions.

2. Related Research

Financial Statement Fraud (FSF) detection has been under the limelight for the past few decades, with emphasis placed on accounting anomalies broadly and on financial statement fraud specifically. While initial research made use of statistical or traditional techniques that are both time-consuming and expensive, more recently the focus has drifted with the emergence of big data and ML [19]. The statistical approaches are centered on traditional mathematical methods, while methods involving ML are focused on learning and intelligence. While both categories have many similarities, the key difference between them is that statistical methods are more rigid, while ML methods are able to learn from and adapt to the problem domain [20].

In addition, previous studies have shown the superior efficiency of ML approaches over conventional statistical approaches [21]. According to the literature, there is no one-size-fits-all strategy for detecting financial statement fraud [22]. The findings of [23] show that models built using ML approaches can efficiently detect financial statement fraud, keep up with the continuous evolution of financial reporting fraudulent behavior, and respond with the most up-to-date technology. In [24], the authors demonstrate that detection models developed using ML approaches are more accurate than traditional methods. Therefore, our review of past research is limited to papers that have used only ML techniques for FSF detection. Most of the research in the previous literature has formulated the detection of FSF as a binary classification problem, sometimes as a multi-class problem and other times as a clustering problem. Researchers have conducted both quantitative and qualitative FSF analyses. Text mining has been used extensively for qualitative research. Here, we focus on papers that perform quantitative analysis using ML techniques. In the initial stages, research mainly included the Neural Network (NN) [25–29], LR, DT, SVM, Discriminant Analysis (DA), and Bayesian Belief Network approaches. Supervised learning techniques have been selected for analysis more than unsupervised ones, with studies from the USA, China, Taiwan, and Spain making up 65% of such papers [30]. A considerable number of studies have analyzed the performance of classifiers on FSF detection, showing that SVM [25,26,31–34], NN [35–40], and DT [41,42] perform well in FSF detection/prediction.

In recent years, ensemble ML techniques have begun to be be used in studies, mostly outperforming single classifiers. Ensemble classifiers integrate the predictions of multiple base models. Numerous empirical and theoretical findings have shown that combining different models can improve predictive accuracy [43]. Moreover, ensemble models are well known for their capability to reduce bias and variance. Many researchers have shown interest in studying ensemble models incorporating boosting [14], bagging [44–47], and other hybrid methods [48] on both balanced and unbalanced data. It has been determined that the performance of such models depends on the selection of the base classifiers. An illustration of the reviewed papers is provided in Table 1.

Table 1. This is a very wide table.

Year & Ref.	Methods	Data Source (Fraud: Non-Fraud)	Input Features	Best Model (Performance in %)	Limitations
2010 [25]	Probit, NN, LR, SVM	AAER (205:6427)	23 raw variables	SVM (AUC – 87.8)	Unable to provide adequate fraud detection capabilities, limited features germane to the domain of fraud detection.
2011 [26]	LR, LDA, C4.5, MLP, RBF, SVM	Taiwan Stock Exchange (25:50)	15 financial ratios + 3 raw variables	SVM (Acc - 92)	Limited selection of input features in analyzing financial statement data.
2011 [31]	SVM, NB, KNN	McGreggor-BFA (123:2888)	14 financial ratios	SVM (Acc-95.9)	The entire data sample was used in both preprocessing and classification algorithm evaluation, and ensemble learning methods are not examined.
2019 [34]	SVM, CART, NN, LR, NB, KNN	Shanghai and Shenzhen Stock exchanges (134:402)	17 financial ratios + 7 non-financial variables	SVM(Acc-81.88)	This study did not cover overall companies that listed in the Shanghai Stock Exchange and Shenzhen Stock Exchange and may also require some necessary modification when it is applied to other countries.
2011 [37]	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)	Class imbalance in data set is not handled and limited selection of input features.
2015 [36]	LR, DT, NN	Taiwan and China sources (129:447)	3 financial ratios + 21 other factors	ANN(Acc-92.8)	Earlier financial statements are difficult to access due to the prolonged study period of time and incomplete samples are eliminated, which may affect the prediction rate.
2016 [42]	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)	Class imbalance in data set is not handled and limited selection of input features.
2012 [44]	Probit regression, Logit regression, SGB, RF, Rule ensemble	AAER (114:114)	12 financial ratios + 1 variable	RF (AUC-90.1)	Test on how this methodology holds up with different sets of fraudulent firms is not analyzed.
2017 [47]	LR, BN, DT, SVM, NN, Bagging, RF, AdaBoost	AAER (311:311)	24 financial variables +8 other variables	RF (TP-86.93)	Feature selection is performed using a filter method, and the study is limited by the use of a balanced sample of fraudulent and non-fraudulent firms.
2014 [45]	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)	The factors including sample, period and changes in the Chinese economy may influence the prediction modes and is unable to provide adequate fraud detection capabilities.
2018 [46]	SVM, RF, DT, ANN, LR	China Securities Regulatory Commission (120:120)	17 financial variables+5 non-financial	RF (Acc-75)	The data is not large enough that only Chinese companies contained and the variables are not various and innovative.
2020 [14]	LR, SVM, RUSBoost, AdaBoost	AAER (1171:204855)	28 raw financial variables	RUSBoost (AUC-72.5)	Raw financial variables are only considered.
Proposed	XGBoost, LR, DT, SVM, AdaBoost, and RF	Osiris database	26 financial attributes including financial attributes from Beneish model	Optimized XGBoost (Acc-96.05)	Non-financial attributes are not considered and is treated as future work.

Although prior research shows that ensemble classifiers are best at detecting FSF, there is less research on them compared to single classifiers. Most previous studies have used imbalanced datasets for evaluation, as is the case with real-world data. Consequently, because of the common problem of class imbalance, traditional ensemble models must generally be coupled with sampling techniques such as oversampling or undersampling in order to balance the class distribution. Only a few studies have considered the imbalance issue during modeling. While most researchers have used financial ratios for prediction, others have argued that raw variables produce better results [49–51]. Various metrics can be used to assess classifier performance, with the prevalent ones being sensitivity or recall,

precision, and accuracy [52]. In this paper, we evaluate the different classifiers that can be used in FSF detection while accounting for the class imbalance issue. We take into account both raw financial variables and financial ratios. In this way, the problem can be solved by ensuring that the related data are distributed among a set of sites across different networks [53–57], which will form part of our future work.

3. Data and Methodology

3.1. Data

Our experimental dataset included 950 companies in the MENA region. All of the selected companies are from different sectors, including manufacturing, technology, energy, telecommunications, real estate, and insurance. Data were collected from the Osiris global company database (source: https://www.bvdinfo.com/en-gb/our-products/data/international/Osiris (accessed on 20 December 2022)). Based on the availability of the data, we selected two consecutive years from 2012 to 2019 for each company, for a final tally of 102 fraudulent years and 1798 non-fraudulent years. The financial indicators in the database are taken from the respective companies' financial statements and balance sheets. The details of the 26 financial attributes, including financial attributes from the Beneish model [13], are provided 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 margins, soaring operating expenses, and increasing leverage encourage companies to manipulate profits. Such companies are most likely to alter profits by speeding up sales recognition, increasing accruals and cost deferrals, and minimizing depreciation. Therefore, we have taken these attributes into account in our study.

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 administrative 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	Day's 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)

3.2. Methodology

A schematic representation of the stages involved in the rest of this study is shown in Figure 1. The first phase is the data preprocessing phase, followed by the classifier modeling phase, and finally the optimization phase.

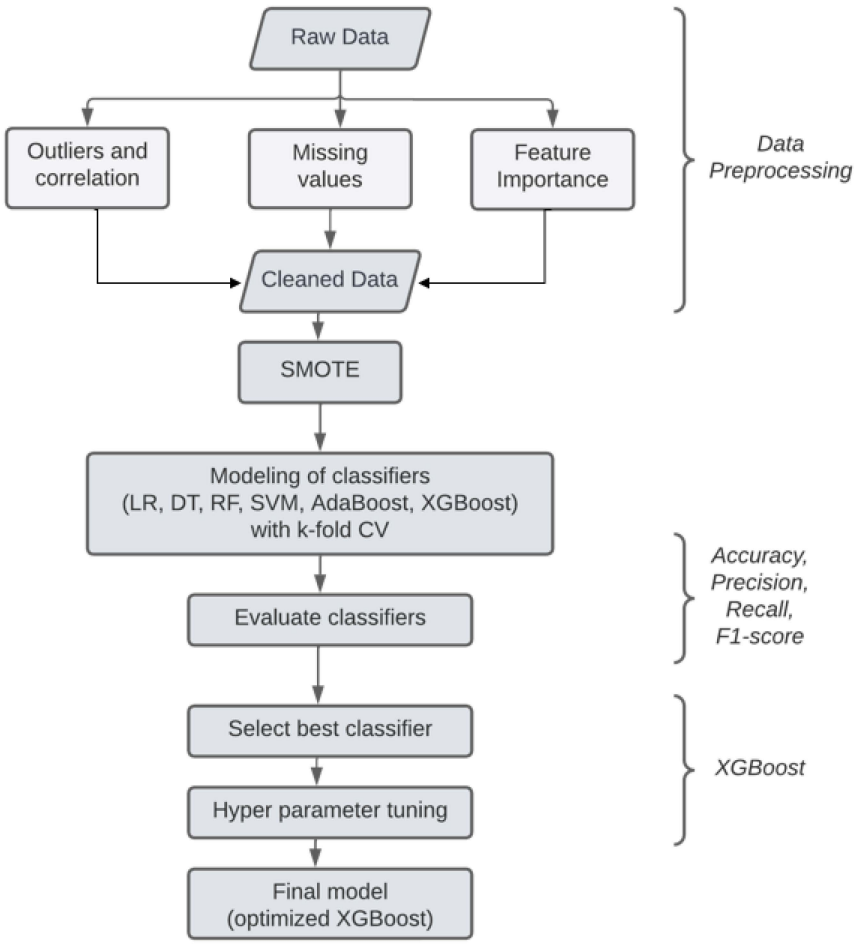

Figure 1. Stages involved in this study.

3.2.1. Data Preprocessing

Our dataset is limited, as there are not many publicly available companies in the MENA region. Rather than list-wise deletion, it is more suitable to deal with missing values by replacing them [58–61]. In this work, the missing values in the dataset were replaced using the within-country mean values. Normalization was carried out using MinMaxScaler.

Detailed descriptive statistics were obtained, and outliers in the dataset were detected and excluded. Feature Importance was calculated by calculating the Gini importance value for each feature. Each feature was then sorted in descending order and the top k features were selected. The 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 dataset was imbalanced and small, we balanced it with synthetic minority data by oversampling [62], which has been extensively and successfully used in the literature for similar datasets [63–65].

3.2.2. Synthetic Minority Oversampling Technique (SMOTE)

In the case of fraud detection problems the minority class needs to receive 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 standard classifiers is biased towards the majority class, as they are programmed to minimize the

overall inaccuracy of classification regardless of class distribution. This bias problem can be overcome by excluding examples of the majority class, known as undersampling, or by including new examples of the minority class, known as oversampling. Due to the small size of our dataset, we chose the latter.

An effective oversampling technique for producing new examples is SMOTE [62], which is one of the more advanced sampling methods. SMOTE can be implemented independently of the classifier being used. This algorithm addresses the challenge of overfitting caused by random oversampling. 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 [66]. In particular, a random sample of the minority class is chosen first. Then, for this sample, the value k of the nearest neighbors is found (usually k = 5). A randomly selected neighbor is chosen and a synthetic example is constructed at a randomly selected point in between two examples in the feature space.

It is evident from prior studies that SVM, NN, DT, and LR have good performance for the task of fraud detection. For comparison with XGBoost, we selected SVM, DT, and LR. From the ensemble methods, we selected the RF and AdaBoost algorithms, as decision tree-based algorithms are considered best for small- and medium-sized data. Model training and testing of the comparison phase algorithms was performed with the help of the Scikit-learn package from Python.

After the comparison phase, it was clear that XGBoost, a tree-based algorithm, was the best dataset. In the next stage, we optimized the XGBoost algorithm to obtain an optimal hyperparameter combination with the help of RandomizedSearchCV from Scikit-learn, which performs a randomized search of hyperparameters, to further enhance the performance of the algorithm. The estimator parameters used to implement these methods were optimized by performing cross-validation on the parameter settings search.

3.2.3. Base Classifiers: SVM, DT, and LR

Support Vector Machine (SVM)

An SVM [67,68] is a discriminative classifier, typically explained as a separating hyperplane. Put another way, this algorithm generates an optimal hyperplane that classifies new instances from labelled training data (supervised learning). The data points or vectors nearest to the hyperplane, which influence the direction of the hyperplane, are referred to as the Support Vector, as these vectors support the hyperplane. SVM has high predictive accuracy and generalization capabilities, particularly for small, nonlinear, and high-dimensional samples [45]. Here, we used linear SVM.

Decision Tree (DT)

DTs are among the most successful ML algorithms thanks to their intelligibility and clarity [69]. DT approaches are used for estimation, clustering, and classification tasks [70]. The best attribute is placed at the root node. This attribute is selected based on a measure of information gain that is subsequently used at each stage of tree building. The training dataset is then split into two or more subsets based on the values of the chosen attribute. This process is repeated for each of the subsets, selecting the best attribute for each and creating child nodes. The process continues until a stopping criterion is met, such as reaching a maximum depth or all instances belonging to the same class. The leaf nodes in the decision tree reflect the class, while the decision nodes determine the rules. The test data class is predicted by the decision rules. Among the key benefits of Decision Trees are that they offer a model that meaningfully describes the acquired knowledge and enables the extraction of if–then classification rules [36].

Logistic Regression (LR)

LR as a general statistical model was originally developed and popularized primarily by Joseph Berkson [71]. LR is conducted when the dependent variable is binary. It is

a discriminative classifier that is linear in its parameters, and is used to describe the relationship between a single dependent binary variable and one or more independent variables. LR can handle both nominal and numerical data. It predicts the likelihood of a binary response based on one or more predictor attributes [72].

3.2.4. Ensembles: RF, AdaBoost, and XGBoost

Predictions from previously developed individual base estimators can be integrated using ensemble strategies to improve robustness and generalization ability; they can often deliver better results compared to a single estimator. Even when the individual models in the ensembles are fairly simple, the power of ensembling can lead to the creation of strong ensemble models.

Random Forest (RF)

Random Forest (RF) is a bagging ensemble technique proposed particularly for trees [73]. The base model of an RF is a DT. RF addresses the issue of high variance in DTs by combining the predictions of multiple DTs, with each DT trained on a different subset of the data and a different subset of the features. The resulting combination of trees leads to a more robust and less variable model. Random subsets of the data are generated via replacement, and each subset is trained with the help of a DT. While expanding the trees, RF adds more randomness to the structure [74]. As a result, there is a wide range of diversity, which contributes to a successful model in general. As a result, in an RF the algorithm only considers a random subset of the features when dividing a node. The randomness of the trees can be increased by using additional random thresholds for every feature, instead of looking for the highest suitable thresholds as in a normal DT.

AdaBoost

AdaBoost, or Adaptive boosting, was the first really successful boosting algorithm developed for binary classification [75]. It is an approach to minimize the error of a weak learning algorithm. Theoretically, any algorithm can be a weak learning algorithm if it can produce classifiers that need to be slightly more consistent than random guessing [76]. AdaBoost helps to combine multiple "weak classifiers" into a single "strong classifier" [77]. The most common algorithms used with AdaBoost are DTs. A weak classifier (decision stump) is prepared using weighted samples from the training data. Only binary (two-class) classification problems are supported; thus, 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 and trained with weighted training data. The method progresses until a predetermined number of weak learners has been generated or no more improvements can be achieved on the training dataset.

XGBoost

XGBoost, or eXtreme Gradient Boosting, is a tree-based algorithm [78]. XGBoost has shown great success in terms of both 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 in such a way 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", whenever a weak learner is added. Because of this 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. This technique performs well even when there are missing values or many zero values, demonstrating a good ability to deal with sparsity. XGBoost uses an algorithm called the "weighted quantile sketch algorithm" that helps the classifier to concentrate on data that are incorrectly classified. In each iteration, the aim of each new learner is to learn how to classify the incorrect data.

4. Implementation and Analysis

4.1. Implementation

We used Python 3.8 for implementation. Detailed descriptive statistics were obtained using pandas_profiling in Python. The outliers in the dataset were detected with the help of IsolationForest, while the most important features were enumerated using the ExtraTreesClassifier in sklearn. SMOTE was implemented using the imblearn package, with k_neighbors = 5. All models were implemented using the Scikit Learn library and evaluated using ten-fold cross-validation.

The Pearson's correlation and feature importance of the attributes are depicted in Figures 2 and 3, respectively. A high correlation means values between −0.50 and −1.00. Feature importance order is depicted in Figure 3. We included all the attributes, as omitting the least important ones resulted in lower performance; extreme outliers in the dataset were detected, analyzed (e.g., to determine whether they resulted from errors in data entry), and removed.

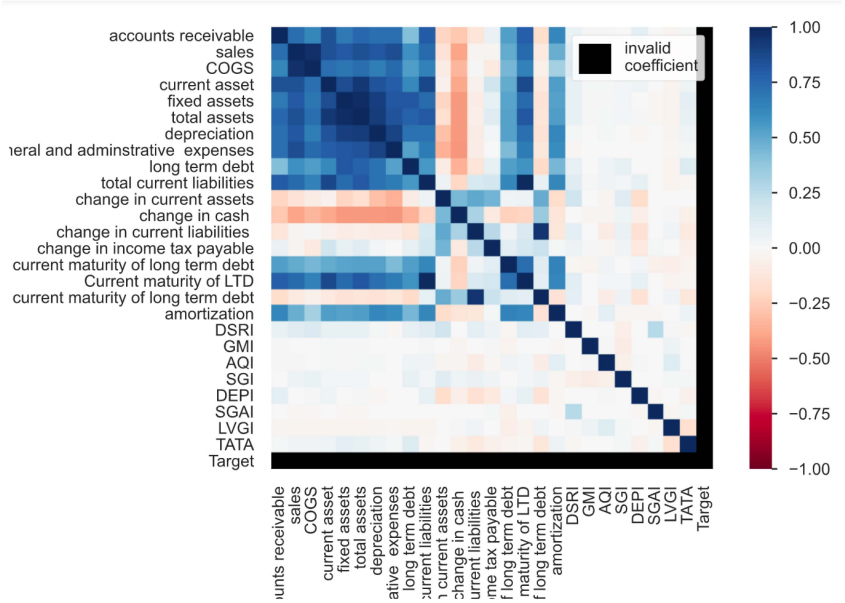

Figure 2. Pearson's r correlation.

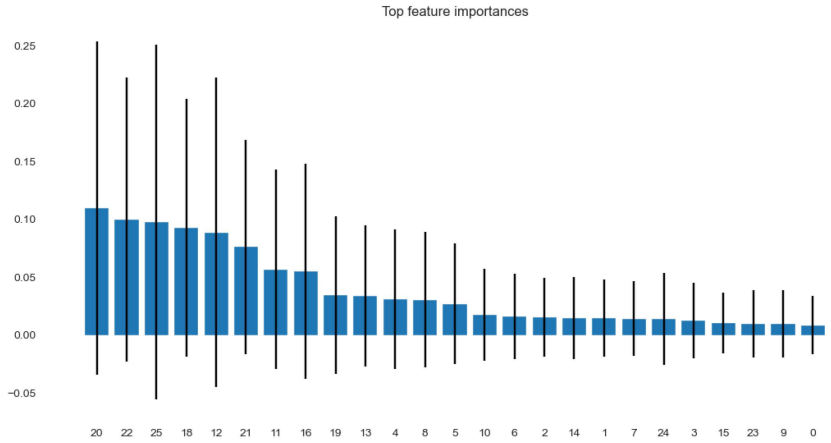

Figure 3. Feature importance of the attributes.

All classifiers were modelled using sklearn. LR was implemented using the LogisticRegression method. DT was implemented using the DecisionTreeClassifier method with max_depth = 4 and the criterion set to 'entropy'. SVM was implemented using LinearSVC. For the ensemble classifiers, RF was modelled using RandomForestClassifier with

n_estimators = 100. AdaBoost was modelled using AdaBoostClassifier with sigmoid kernel SVC as the base estimator and default n_estimators and learning_rate. Finally, XGBoost was modelled using XGBClassifier() with default parameters, then the hyperparameters were fine tuned in the subsequent phase.

4.2. Performance Measures

The predictive performance of data mining classifiers is measured in terms of accuracy, precision, recall, and F1-score, the common evaluation metrics of ML. The testing accuracy and F1-score measures are used for performance evaluation. The k-fold cross-validation score, with k set to 10, was used for all the models (XGBoost has inbuilt CV). Accuracy is the ratio of the 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 the actual class; and F1 Score is the weighted average of Precision and Recall.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \quad (1)$$

$$Precision = \frac{TruePositive}{(TotalPredictedPositive)} \quad (2)$$

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

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

4.3. Analysis

The accuracy of any ML algorithm is highly dependent on the problem at hand, as well as on the integrity and complexity of the training dataset. The prediction performance of all six models on the dataset with SMOTE applied are listed in Table 3, with graphical representations displayed in Figures 4–6.

Table 3. Prediction results of classifiers after SMOTE.

Classifier	Accuracy		Precision	Recall	F1-Score	
	Mean	Std. Dev.			Mean	Std. Dev.
LR	0.7388	0.049	0.8055	0.8344	0.8196	0.067
DT	0.8222	0.06	0.4111	0.5000	0.4513	0.071
SVM	0.8888	0.046	0.8034	0.8411	0.8218	0.076
RF	0.8055	0.051	0.4027	0.5000	0.4461	0.08
AdaBoost	0.8333	0.044	0.4166	0.5000	0.4545	0.066
XGBoost	0.9366	0.05	0.7938	0.8637	0.8272	0.082

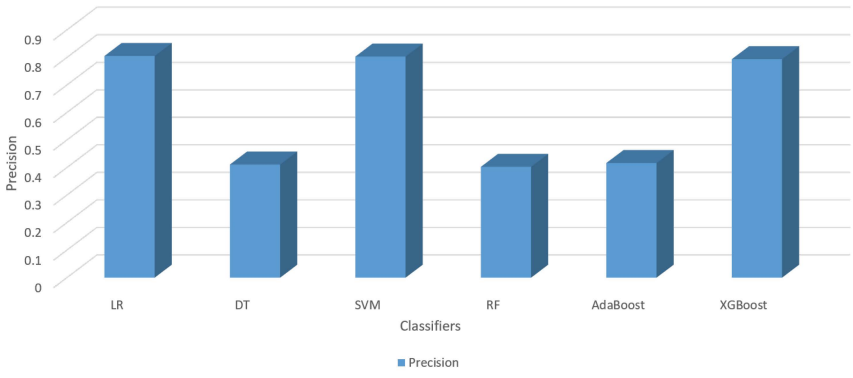

Figure 4. Evaluation results for the classifiers (Precision).

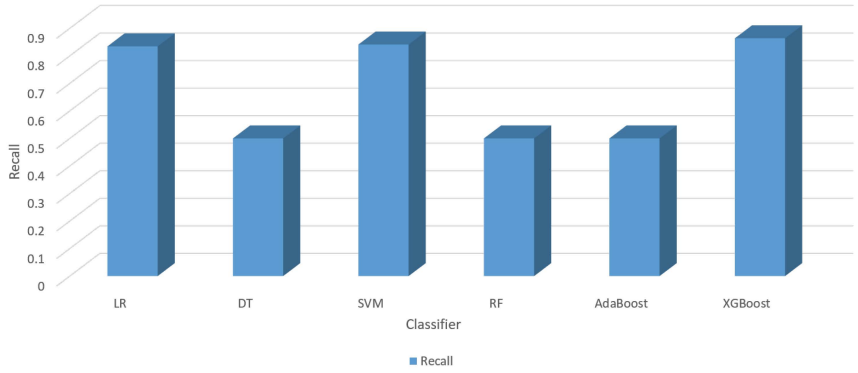

Figure 5. Evaluation results for the classifiers (Recall).

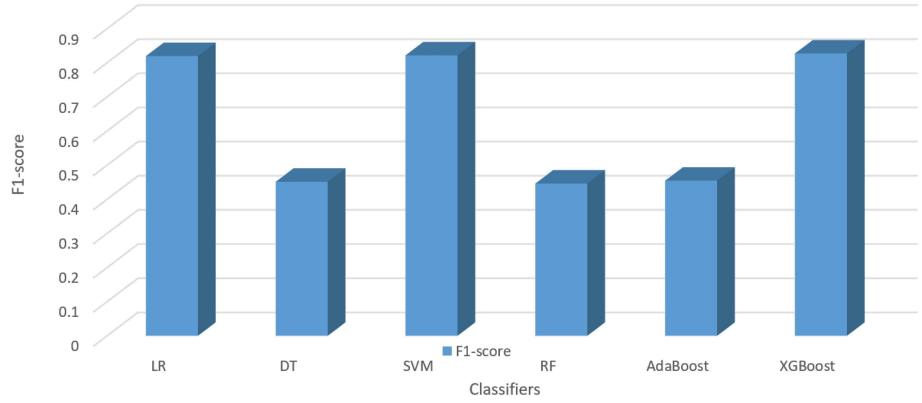

Figure 6. Evaluation results for the classifiers (F1-score).

The highest mean accuracy rate was achieved by the XGBoost algorithm, followed by SVM and then AdaBoost. LR had a low accuracy rate compared to the others, although it had a high F1-score of 0.8196, as seen in Figure 6 and Table 3. SVM displayed an accuracy rate of 0.8888, second to XGBoost, and from Figures 4–6 it can be seen that it had good performance on the basis of Precision, Recall, and F1-score. DT, RF, and AdaBoost had average performance on all four metrics.

It is evident from these results 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 was split into training and testing sets with test_size = 0.3. SVM performed well, though not as well as the XGBoost algorithm. Accuracy and F1-score were obtained after k-fold cross-validation on the training dataset, with k set to 10.

4.4. XGBoost Optimization

Based on preliminary observations, XGBoost is the best model for the detection of fraud using the financial statements in our dataset. Next, we further optimized the performance of XGBoost via hyperparameter tuning using RandomizedSearchCV on the accuracy scores with n_iter = 1000 and three-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],
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 were run and the model was trained until validation_0-error improved in ten rounds. The fitting was achieved by three-fold cross-

validation for each of 1000 candidates, totaling 3000 folds. The best iteration for each round was the one with the least validation error. The list of the best parameters is provided below:

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 the XGBoost algorithm on our SMOTE-sampled MENA dataset was 0.9605, which is a significant improvement over the accuracy score of 0.9366 in the previous stage. The XGBoost optimization process is summarized in Table 4.

Table 4. XGBoost optimization process.

Steps	Description
Hyperparameter tuning using RandomizedSearchCV on accuracy scores with n_iter = 1000 and three-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], '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]
Model training until validation_0- error improvement in ten rounds	Fitting achieved by three-fold cross-validation for each of 1000 candidates, totalling 3000 folds. The best iteration for each round is the one with the least validation error.
Select parameters	'learning_rate': 0.03, 'min_child_weight': 3, 'gamma': 1.5, 'subsample': 0.8, 'colsample_bytree': 1.0, 'max_depth': 9, 'reg_lambda': 1
Obtain the best accuracy score	0.9605, a significant improvement on the accuracy score of 0.9366 in the previous stage.

5. Conclusions

This study proposed a better FSF detection model by utilizing data from publicly available financial statements of firms in the MENA region. An FSF prediction model was developed using the ensemble technique and the XGBoost algorithm while investigating the utilization of various ML techniques in detecting financial statement fraud using published financial disclosures. Upsampling was performed on the dataset using the SMOTE technique to prevent class imbalance issues. In our experiments, SMOTE proved to be a beneficial metric for sampling data with a large class imbalance. When learning on an unbalanced dataset, this study shows that XGBoost outperforms other techniques for the task of FSF detection. We conducted analysis and comparison of three individual ML classifiers and three ensemble techniques used widely in FSF detection using a dataset comprising companies from the MENA region. 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 from Table 4 indicate that the XGBoost classifier is the most efficient model for financial statement fraud detection when using these settings. In the next phase of the study, the XGBoost classifier was further optimized by hyperparameter tuning with cross-validation to obtain the best model

for the problem. The simulation results indicated that the proposed model has higher performance compared to both classic ML models and ensemble models. In this study, we have considered only financial attributes. Analysis of the decentralized model and of non-financial attributes may be incorporated in the future.

Author Contributions: All authors contributed together to realize this work, and their exact contributions are difficult to specify. Conceptualization and methodology by A.A.A., S.K. and A.M.K.; Software and original draft preparation by A.A.A. and M.E.-B.; Validation, writing, reviewing, editing, and supervision by M.E.-B., S.K. and A.M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available on request.

Conflicts of Interest: The authors declare that they have no competing interests.

Abbreviations

The following abbreviations are used in this manuscript:

MENA	Middle East and North Africa
XGBoost	eXtreme Gradient Boosting
SMOTE	Synthetic Minority Oversampling Technique
FSF	Financial Statement Fraud
ML	Machine Learning
LR	Logistic Regression
DT	Decision Tree
SVM	Support Vector Machine
RF	Random Forest
NN	Neural Network
ANN	Artificial Neural Network
NB	Naive Bayes
GP	Genetic Programming
LDA	Linear Discriminant Analysis
MLP	Multi-Layer Perceptron
RBF	Radial Basis Function
KNN	K Nearest Neighbors
BN	Bayesian Network
BBN	Bayesian Belief Network
CART	Classification And Regression Tree
DA	Discriminant Analysis
SGB	Stochastic Gradient Boosting
ACFE	Association of Certified Fraud Examiners
CAQ	Center for Audit Quality
AAER	Accounting and Auditing Enforcement Releases
AUC	Area Under the receiver operating characteristic Curve
Acc	Accuracy
TP	True Positive

References

1. El-Bannany, M.; Sreedharan, M.; Khedr, A.M. A robust deep learning model for financial distress prediction. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*, 170–175. [CrossRef]
2. Kulikova, L.; Satdarova, D. Internal control and compliance-control as effective methods of management, detection and prevention of financial statement fraud. *Acad. Strateg. Manag. J.* **2016**, *15*, 92.
3. Deebak, B.; Memon, F.H.; Dev, K.; Khowaja, S.A.; Wang, W.; Qureshi, N.M.F. TAB-SAPP: A trust-aware blockchain-based seamless authentication for massive IoT-enabled industrial applications. *IEEE Trans. Ind. Inform.* **2022**, *19*, 243–250. [CrossRef]
4. Sreedharan, M.; Khedr, A.M.; El Bannany, M. A Multi-Layer Perceptron Approach to Financial Distress Prediction with Genetic Algorithm. *Autom. Control. Comput. Sci.* **2020**, *54*, 475–482. [CrossRef]

5. Kumar, R.; Tripathi, R. Secure healthcare framework using blockchain and public key cryptography. In *Blockchain Cybersecurity, Trust and Privacy*; Springer: Cham, Switzerland, 2020; pp. 185–202.
6. Andrew, C.; Robin. Detecting Fraudulent of Financial Statements Using Fraud S.C.O.R.E Model and Financial Distress. *Int. J. Econ. Bus. Account. Res. (IJEBAR)* **2022**, *6*, 211–222. [CrossRef]
7. Paulo Sérgio Gomes Macedo, H.C.I.; Vieira, E.S. A model to detect financial statement fraud in Portuguese companies by the auditor. *Contaduría Adm.* **2022**, *67*, 185–209.
8. Wadhwa, A.V.K.; Kumar, S. Financial Fraud Prediction Models: A Review of Research Evidence. *Int. J. Sci. Technol. Res.* **2020**, *9*, 677–680.
9. Amar, I.A.A.B.; Jarboui, A. Detection of Fraud in Financial Statements: French Companies as a Case Study. *Int. J. Acad. Res. Bus. Soc. Sci.* **2013**, *3*, 456–472. [CrossRef]
10. Alsinglawi, M.M.A.S.M.A.O.; Almari, M.O.S. Predicting Fraudulent Financial Statements Using Fraud Detection Models. *Acad. Strateg. Manag.* **2021**, *20*, 1–17.
11. Schreiber-Gregory, D.; Bader, K. Logistic and Linear Regression Assumptions: Violation Recognition and Control. In Proceedings of the SESUG Conference, St. Pete Beach, FL, USA, 14–17 October 2018; pp. 1–6.
12. Pintelas, P.; Livieris, I. Ensemble learning and their applications. *Algorithms* **2020**, 1–184.
13. Beneish, M.D. The detection of earnings manipulation. *Financ. Anal. J.* **1999**, *55*, 24–36. [CrossRef]
14. Bao, Y.; Ke, B.; Li, B.; Yu, Y.J.; Zhang, J. Detecting accounting fraud in publicly traded US firms using a machine learning approach. *J. Account. Res.* **2020**, *58*, 199–235. [CrossRef]
15. Fernández-Delgado, M.; Cernadas, E.; Barro, S.; Amorim, D. Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.* **2014**, *15*, 3133–3181.
16. Gorenc, M. Empirical evidence of financial statement manipulation during economic recessions. *Management* **2019**, *14*, 19–31. [CrossRef]
17. Tilden, C.; Janes, T. Benford's Law as a Useful Tool to Determine Fraud in Financial Statements. *J. Financ. Account.* **2012**, *14*, 1–15.
18. Saville, A. Using Benford's Law to Detect Data Error and Fraud: An Examination Of Companies Listed on the Johannesburg Stock Exchange. *SAJEMS* **2006**, *9*, 341–354. [CrossRef]
19. Mohammadi, M.; Yazdani, S.; Khanmohammadi, M.H.; Maham, K. Financial reporting fraud detection: An analysis of data mining algorithms. *Int. J. Financ. Manag. Account.* **2020**, *4*, 1–12.
20. Humpherys, S.L.; Moffitt, K.C.; Burns, M.B.; Burgoon, J.K.; Felix, W.F. Identification of fraudulent financial statements using linguistic credibility analysis. *Decis. Support Syst.* **2011**, *50*, 585–594. [CrossRef]
21. West, J.; Bhattacharya, M.; Islam, R. Intelligent financial fraud detection practices: An investigation. In Proceedings of the International Conference on Security and Privacy in Communication Networks, Beijing, China, 24–26 September 2014; Springer: Cham, Switzrland, 2014; pp. 186–203.
22. Hamal, S.; Senvar, Ö. Comparing performances and effectiveness of machine learning classifiers in detecting financial accounting fraud for Turkish SMEs. *Int. J. Comput. Intell. Syst.* **2021**, *14*, 769–782. [CrossRef]
23. Craja, P.; Kim, A.; Lessmann, S. Deep learning for detecting financial statement fraud. *Decis. Support Syst.* **2020**, *139*, 113421. [CrossRef]
24. Gupta, S.; Mehta, S.K. Data mining-based financial statement fraud detection: Systematic literature review and meta-analysis to estimate data sample mapping of fraudulent companies against non-fraudulent companies. *Glob. Bus. Rev.* **2021**, 1–26. [CrossRef]
25. Cecchini, M.; Aytug, H.; Koehler, G.J.; Pathak, P. Detecting management fraud in public companies. *Manag. Sci.* **2010**, *56*, 1146–1160. [CrossRef]
26. Pai, P.F.; Hsu, M.F.; Wang, M.C. A support vector machine-based model for detecting top management fraud. *Knowl.-Based Syst.* **2011**, *24*, 314–321. [CrossRef]
27. Alfaiz, N.S.; Fati, S.M. Enhanced Credit Card Fraud Detection Model Using Machine Learning. *Electronics* **2022**, *11*, 662. [CrossRef]
28. Strelcenia, E.; Prakoonwit, S. Improving Classification Performance in Credit Card Fraud Detection by Using New Data Augmentation. *AI* **2023**, *4*, 172–198. [CrossRef]
29. Kumar, S.; Ahmed, R.; Bharany, S.; Shuaib, M.; Ahmad, T.; Tag Eldin, E.; Rehman, A.U.; Shafiq, M. Exploitation of Machine Learning Algorithms for Detecting Financial Crimes Based on Customers' Behavior. *Sustainability* **2022**, *14*, 13875. [CrossRef]
30. Albashrawi, M. Detecting financial fraud using data mining techniques: A decade review from 2004 to 2015. *J. Data Sci.* **2016**, *14*, 553–569. [CrossRef]
31. Perols, J. Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Audit. J. Pract. Theory* **2011**, *30*, 19–50. [CrossRef]
32. Asimit, A.V.; Kyriakou, I.; Santoni, S.; Scognamiglio, S.; Zhu, R. Robust Classification via Support Vector Machines. *Risks* **2022**, *10*, 154. [CrossRef]
33. Moepya, S.O.; Akhoury, S.S.; Nelwamondo, F.V. Cost-sensitive classification for financial fraud detection under high class-imbalance. In Proceedings of the 2014 IEEE international conference on data mining workshop, Shenzhen, China, 14–17 December 2014; IEEE: New York, NY, USA, 2014; pp. 183–192.
34. Yao, J.; Pan, Y.; Yang, S.; Chen, Y.; Li, Y. Detecting fraudulent financial statements for the sustainable development of the socio-economy in China: A multi-analytic approach. *Sustainability* **2019**, *11*, 1579. [CrossRef]

35. Han, D. Researches of Detection of Fraudulent Financial Statements Based on Data Mining. *J. Comput. Theor. Nanosci.* **2017**, *14*, 32–36. [CrossRef]
36. Lin, C.C.; Chiu, A.A.; Huang, S.Y.; Yen, D.C. Detecting the financial statement fraud: The analysis of the differences between data mining techniques and experts' judgments. *Knowl.-Based Syst.* **2015**, *89*, 459–470. [CrossRef]
37. Ravisankar, P.; Ravi, V.; Rao, G.R.; Bose, I. Detection of financial statement fraud and feature selection using data mining techniques. *Decis. Support Syst.* **2011**, *50*, 491–500. [CrossRef]
38. Rizki, A.A.; Surjandari, I.; Wayasti, R.A. Data mining application to detect financial fraud in Indonesia's public companies. In Proceedings of the 2017 3rd International Conference on Science in Information Technology (ICSITech), Bandung, Indonesia, 25–26 October 2017; IEEE: New York, NY, USA, 2017; pp. 206–211.
39. Murorunkwere, B.F.; Tuyishimire, O.; Haughton, D.; Nzabanita, J. Fraud Detection Using Neural Networks: A Case Study of Income Tax. *Future Internet* **2022**, *14*, 168. [CrossRef]
40. Pérez López, C.; Delgado Rodríguez, M.; de Lucas Santos, S. Tax Fraud Detection through Neural Networks: An Application Using a Sample of Personal Income Taxpayers. *Future Internet* **2019**, *11*, 86. [CrossRef]
41. Gupta, R.; Gill, N.S. Prevention and detection of financial statement fraud–An implementation of data mining framework. *Editor. Pref.* **2012**, *3*, 150–160. [CrossRef]
42. Chen, S. Detection of fraudulent financial statements using the hybrid data mining approach. *SpringerPlus* **2016**, *5*, 1–16. [CrossRef] [PubMed]
43. Misstatements; Bertomeu, J.; Cheynel, E.; Floyd, E.; Pan, W. *Ghost in the Machine: Using Machine Learning to Uncover Hidden*; Springer: Cham, Switzrland, 2018; Volume 4, pp. 233–241.
44. Whiting, D.G.; Hansen, J.V.; McDonald, J.B.; Albrecht, C.; Albrecht, W.S. Machine learning methods for detecting patterns of management fraud. *Comput. Intell.* **2012**, *28*, 505–527. [CrossRef]
45. Song, X.P.; Hu, Z.H.; Du, J.G.; Sheng, Z.H. Application of machine learning methods to risk assessment of financial statement fraud: Evidence from China. *J. Forecast.* **2014**, *33*, 611–626. [CrossRef]
46. Yao, J.; Zhang, J.; Wang, L. A financial statement fraud detection model based on hybrid data mining methods. In Proceedings of the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 26–28 May 2018; IEEE: New York, NY, USA, 2018; pp. 57–61.
47. Hajek, P.; Henriques, R. Mining corporate annual reports for intelligent detection of financial statement fraud—A comparative study of machine learning methods. *Knowl.-Based Syst.* **2017**, *128*, 139–152. [CrossRef]
48. Li, H.; Wong, M.L. Financial fraud detection by using Grammar-based multi-objective genetic programming with ensemble learning. In Proceedings of the 2015 IEEE Congress on Evolutionary Computation (CEC), Sendai, Japan, 25–28 May 2015; IEEE: New York, NY, USA, 2015; pp. 1113–1120.
49. Ragab, Y. Financial Ratios and Fraudulent Financial Statements Detection: Evidence from Egypt. *Int. J. Acad. Res.* **2017**, *4*, 1–6.
50. Kanapickiene, R.; Grundiene, Z. The Model of Fraud Detection in Financial Statements by Means of Financial Ratios. *Procedia Soc. Behav. Sci.* **2015**, *213*, 321–327. [CrossRef]
51. Grove, H.; Basilico, E. Fraudulent Financial Reporting Detection Key Ratios Plus Corporate Governance Factors. *Int. Stud. Mgt. Org.* **2008**, *38*, 10–42. [CrossRef]
52. Gu, Q.; Zhu, L.; Cai, Z. Evaluation measures of the classification performance of imbalanced data sets. In Proceedings of the Computational Intelligence and Intelligent Systems: 4th International Symposium, ISICA 2009, Huangshi, China, 23–25 October 2009; Proceedings 4. Springer: Cham, Switzerland, 2009; pp. 461–471.
53. Khedr, A.M.; Mahmoud, R. Agents for integrating distributed data for function computations. *Comput. Inform.* **2012**, *31*, 1101–1125.
54. Khedr, A.M.; Mahmoud, R. Decomposable naive Bayes classifier for partitioned data. *Comput. Inform.* **2012**, *31*, 1511–1531.
55. Khedr, A.M.; Raj, P.P. DRNNA: Decomposable Reverse Nearest Neighbor Algorithm for Vertically Distributed Databases. In Proceedings of the 2021 18th International Multi-Conference on Systems, Signals and Devices (SSD), Monastir, Tunisia, 22–25 March 2021; pp. 681–686.
56. Khedr, A.M. Decomposable algorithm for computing k-nearest neighbours across partitioned data. *Int. J. Parallel Emergent Distrib. Syst.* **2016**, *31*, 334–353. [CrossRef]
57. Khedr, A.M.; Osamy, W.; Salim, A.; Salem, A. Privacy preserving data mining approach for IoT based WSN in smart city. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 555–563. [CrossRef]
58. Palanivinayagam, A.; Damaševičius, R. Effective Handling of Missing Values in Datasets for Classification Using Machine Learning Methods. *Information* **2023**, *14*, 92. [CrossRef]
59. Kim, T.; Ko, W.; Kim, J. Analysis and Impact Evaluation of Missing Data Imputation in Day-ahead PV Generation Forecasting. *Appl. Sci.* **2019**, *9*, 204. [CrossRef]
60. França, C.M.; Couto, R.S.; Velloso, P.B. Missing Data Imputation in Internet of Things Gateways. *Information* **2021**, *12*, 425. [CrossRef]
61. Weed, L.; Lok, R.; Chawra, D.; Zeitzer, J. The Impact of Missing Data and Imputation Methods on the Analysis of 24-Hour Activity Patterns. *Clocks Sleep* **2022**, *4*, 497–507. [CrossRef]
62. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]

63. Elreedy, D.; Atiya, A.F. A comprehensive analysis of synthetic minority oversampling technique (SMOTE) for handling class imbalance. *Inf. Sci.* **2019**, *505*, 32–64. [CrossRef]
64. Rivera, W.A.; Xanthopoulos, P. A priori synthetic over-sampling methods for increasing classification sensitivity in imbalanced data sets. *Expert Syst. Appl.* **2016**, *66*, 124–135. [CrossRef]
65. Goyal, A.; Rathore, L.; Kumar, S. A survey on solution of imbalanced data classification problem using smote and extreme learning machine. In *Communication and Intelligent Systems: Proceedings of ICCIS 2020*; Springer: Cham, Switzerland, 2021; pp. 31–44.
66. Mishra, S. Handling imbalanced data: SMOTE vs. random undersampling. *Int. Res. J. Eng. Technol. (IRJET)* **2017**, *4*, 317–320.
67. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
68. Alhashmi, S.M.; Khedr, A.M.; Arif, I.; El Bannany, M. Using a Hybrid-Classification Method to Analyze Twitter Data During Critical Events. *IEEE Access* **2021**, *9*, 141023–141035. [CrossRef]
69. Wu, X.; Kumar, V.; Quinlan, J.R.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G.J.; Ng, A.; Liu, B.; Philip, S.Y.; et al. Top 10 algorithms in data mining. *Knowl. Inf. Syst.* **2008**, *14*, 1–37. [CrossRef]
70. Khedr, A.M.; Arif, I.; P V, P.R.; El-Bannany, M.; Alhashmi, S.M.; Sreedharan, M. Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. *Intell. Syst. Account. Financ. Manag.* **2021**, *28*, 3–34. [CrossRef]
71. Cramer, J.S. The origins of logistic regression. *SSRN* **2002**, *119*, 1–16. [CrossRef]
72. Randhawa, K.; Loo, C.K.; Seera, M.; Lim, C.P.; Nandi, A.K. Credit card fraud detection using AdaBoost and majority voting. *IEEE Access* **2018**, *6*, 14277–14284. [CrossRef]
73. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Lausanne, Switzerland, 5–10 September 1995; IEEE: New York, NY, USA, 1995; Volume 1, pp. 278–282.
74. Lin, W.; Wu, Z.; Lin, L.; Wen, A.; Li, J. An ensemble random forest algorithm for insurance big data analysis. *IEEE Access* **2017**, *5*, 16568–16575. [CrossRef]
75. Freund, Y.; Schapire, R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
76. Sun, J.; Jia, M.Y.; Li, H. AdaBoost ensemble for financial distress prediction: An empirical comparison with data from Chinese listed companies. *Expert Syst. Appl.* **2011**, *38*, 9305–9312. [CrossRef]
77. Sreedharan, M.; Khedr, A.M.; El Bannany, M. A comparative analysis of machine learning classifiers and ensemble techniques in financial distress prediction. In Proceedings of the 2020 17th International Multi-Conference on Systems, Signals & Devices (SSD), Monastir, Tunisia, 20–23 July 2020; IEEE: New York, NY, USA, 2020; pp. 653–657.
78. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the KDD 16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]