UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Advancing the power of machine learning in financial decision-making

Anomaly detection, fraud identification, and earnings forecasting

Bhattacharya, I.

Publication date
2023

Document Version
Final published version

Link to publication

Citation for published version (APA):
Bhattacharya, I. (2023). *Advancing the power of machine learning in financial decision-making: Anomaly detection, fraud identification, and earnings forecasting*. [Thesis, fully internal, Universiteit van Amsterdam].

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:28 Nov 2023

ADVANCING THE POWER OF MACHINE LEARNING IN FINANCIAL DECISION MAKING:

ANOMALY DETECTION
FRAUD IDENTIFICATION
EARNINGS FORECASTING

INDRANIL BHATTACHARYA

Advancing the Power of Machine Learning in Financial Decision-Making:
Anomaly Detection, Fraud Identification, and Earnings Forecasting

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op woensdag 11 oktober 2023, te 10.00 uur

door Indranil Bhattacharya
geboren te Arambagh

Promotiecommissie

Promotor:	prof. dr. E.E.O. Roos Lindgreen	Universiteit van Amsterdam
Copromotor:	prof. dr. J.F.M.G. Bouwens	Universiteit van Amsterdam
Overige leden:	prof. dr. A.H. Gold	Vrije Universiteit Amsterdam
	prof. dr. A. Shahim	Vrije Universiteit Amsterdam
	prof. dr. S.I. Birbil	Universiteit van Amsterdam
	prof. dr. E. Kanoulas	Universiteit van Amsterdam
	prof. dr. S. Klous	Universiteit van Amsterdam

Faculteit Economie en Bedrijfskunde

Bhattacharya, I

Advancing the power of Machine Learning in Financial Decision-making: Anomaly Detection, Fraud Identification, and Earnings Forecasting

Thesis, University of Amsterdam, Amsterdam

ISBN: 978-94-6469-554-0

©I.Bhattacharya, 2023

All rights reserved. No part of this thesis may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, mechanically, by photocopying, recording, or otherwise, without prior permission of the author.

Cover design by: Saagnik Paul

Printed by: ProefschriftMaken || https://www.proefschriftmaken.nl

Contents

List of Tables

List of Figures

1 Introduction

Financial accounting is an important information source that helps stakeholders to decide the extent to which they can conduct business within a firm. Researchers and practitioners leverage their experience and domain expertise to analyze and investigate both structured (example: accounting numbers) and unstructured data (example: business texts) to draw actionable inferences. However, such investigations are costly and require a significant amount of time and effort [Van den Bogaerd and Aerts, 2011].

Advanced machine learning techniques can be trained to draw inferences akin to the human brain by processing data. Nonetheless, researchers argue that the financial accounting domain is yet to leverage the advanced machine learning algorithms to their full potential [Pratt, 2015, Lev and Gu, 2016, Bertomeu, 2020]. In this dissertation, I aim to create the foundation of a strong relationship between the state-of-the-art applied machine learning literature and the literature on financial accounting that produces actionable insights in a both cost and time-efficient manner.

Because of its black-box nature, studies argue that practitioners should be careful about the insights drawn from advanced machine learning algorithms [Dickey et al., 2019]. To reduce the risk of finding inaccurate or illogical patterns, it is important to optimize human participation in decision-making. Hence, I also focus to explore if, alongside the knowledge, the domain expertise can also be employed to validate the model outputs. Overall, I propose machine learning frameworks that are not only effective in capturing patterns but also more efficient in drawing actionable insights. This also provides me with the opportunity to explore state-of-the-art algorithms and introduce them into the accounting domain.

Accountants and auditors play pivotal roles in producing and verifying financial reports that are used to evaluate the financial performance of companies. Shareholders and other stakeholders rely on these financial reports produced by the companies to strategize their investments. However, these financial reports are the aggregation of big transaction-level data from day-to-day business. Hence, a small number of anomalous observations in such big data can potentially cause inaccuracy in the financial reports. Auditors sift through these data to find if there are anomalies present in data which takes significant effort and time to scrutinize.

In chapter 2, co-authored with my supervisor Edo Roos Lindgreen, we propose a semi-supervised machine learning framework that detects anomalies in such big data to help produce more accurate financial reports [Bhattacharya and Lindgreen, 2020]. Our proposed method combines both supervised and unsupervised frameworks. The unsupervised algorithm, i.e. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is first applied to a representative subset of the data to generate a training set based on pseudo labels of anomalies. Afterward, the training set is used to direct the supervised Gradient boosting algorithm, i.e. LightGBM for anomaly detection in the remaining data. This approach is applied to an insurance policy dataset consisting of approximately 32 million records. Our proposed framework helps capture 90% and 96% of anomalous observations by investigating 5% and 10% of the data respectively. Comprehensive details are provided throughout chapter 2 to present the practical applicability and widespread potential of the proposed semi-supervised approach for similar problem categories.

Although financial reports are heavily relied upon by the shareholders and stakeholders, we have historically observed how some companies (Enron, WorldCom) delve into fraudulent activities and therefore resulting in significant economic damage. To prevent such a scenario, auditors and financial regulators investigate firms and their reports in-depth to detect such fraudulent activities. These investigation exercises are not only very costly but also take a significant amount of time (on average 2-3 years) to complete [Dyck et al., 2010, Karpoff et al., 2017].

In chapter 3 which is co-authored with Ana Mickovic, we explore how textual contents from financial reports help in detecting these accounting frauds. Pre-trained contextual language learning models, such as BERT, have significantly advanced natural language processing in recent years. We fine-tune the BERT model on Management Discussion and Analysis (MD&A) sections of annual 10-K reports from the Securities and Exchange Commission (SEC) database. Our final model outperforms the textual benchmark model [Brown et al., 2020] and the quantitative benchmark model [Bao et al., 2020] from the previous literature by 15% and 12%, respectively. Furthermore, our model identifies five times more fraudulent firms than the textual benchmark and three times as many as the quantitative benchmark, despite investigating the same number of firms. Optimizing this investigation process, where more fraudulent firms are detected in the same size of the investigation sample, would be of great economic significance for regulators, investors, financial analysts, and auditors.

Apart from the existing information present in the financial reports, investors and shareholders also tend to consider several other factors before strategizing their investments. One such aspect is to attempt to foresee how companies are going to perform in the future. This makes forecasting the earnings of companies for future years an essential subject to study. Predicting the future earnings of firms can also help in effectively allocating resources to society.

In chapter 4, co-authored with Sanjay Bissessur, we use machine learning techniques based on stack ensemble to improve earnings forecasting, combining both hard information from financial statements taken from the Compustat database as well as soft information taken from the management discussion and analysis (MD&A) sections of 10-K filings with the Securities and Exchange Commission. We find that our model outperforms the AR(1) model significantly. Furthermore, this outperformance improves over time. Finally, we introduce a scale-independent metric to evaluate forecasts and find that our models still outperform AR(1). Our results hold for subsamples of losses and non-surviving firms. Taken together, our results underscore the importance of incorporating the interaction between hard information and soft information in forecasting.

All three chapters aim to create more effective and time-efficient solutions for practical challenges in the financial accounting practice and reduce significant effort and time for the real-world practitioner. The method developed in chapter 2 helps in creating a fast solution in an internal audit setting to produce more accurate financial reports by creating a framework containing *human in the loop*. Chapter 3 introduces a model that helps financial investigators to prioritize their investigations in detecting financial accounting frauds. Moreover, the method is shown to be economically significant as it captures more fraudulent firms by investigating the same number of firms. The frameworks proposed in both chapters 2 and 3 contain *human in the loop* to validate the solution by the algorithms and if that aligns with the business expertise. Chapter 4 introduces an architecture that is equivalent to producing combined knowledge of human expertise from different industry domains to produce more accurate future earnings forecasts.

While several pieces of research [Loughran and McDonald, 2016, Bertomeu, 2020] indicate that advanced machine learning techniques are yet to be fully unleashed and explored in the area of financial accounting, all three chapters in this dissertation aim to bridge that gap by introducing the state-of-the-art machine learning algorithms and the best practices from the literature. Considering the vast application that machine learning algorithms have to offer, all three chapters lie in the intersection of both applied machine learning and financial accounting literature to help researchers and practitioners from the accounting domain. Overall these chapters not only elucidate on mimicking the human expertise in the financial accounting domain but also explore to what extent humans can learn and benefit from these proposed machine learning frameworks.

2 A semi-supervised machine learning approach to detect anomalies in big accounting data

2.1 Introduction

The Statement on Auditing Standards 99 (SAS 99, ISA 200) requires auditors to be assured about financial statements being free of material misstatements. Auditors, therefore, need to test the appropriateness of the data that is used in preparing financial statements. In order to achieve this, anomaly detection in large accounting data has become relevant in the audit practice. In the last few decades, advanced machine learning techniques have been encouraged by the accounting professionals to detect anomalies in a large-scale accounting dataset. Detected anomalous observations are investigated by the auditors to understand their suspicious behavior. This helps improve the qualitative rigor of the financial statements through a detailed investigation without having to manually sift through the data.

Although being quite successful, certain challenges in using the existing machine learning methods to detect anomalies in accounting data remain prevalent. Optimizing unsupervised techniques are computationally expensive due to large datasets [Kim, 2009] and the supervised algorithms are incapable to investigate the unlabeled (absence of anomaly flag) datasets often gathered in reality. Therefore, the primary goal of this study is to propose a semi-supervised machine learning approach for anomaly detection in an unlabeled big accounting dataset.

In this chapter, we discuss an innovative semi-supervised machine learning framework for anomaly detection in an unlabeled big accounting data. Our framework is based on a novel combination of DBSCAN [Ester et al., 1996] and LightGBM [Ke et al., 2017]. DBSCAN is a density-based spatial clustering technique with the application of noise. DBSCAN is designed in such a way that it can discover arbitrary shaped clusters in any data and at the same time detect anomalies [Birant and Kut, 2007]. LightGBM is a gradient boosting decision tree algorithm [Friedman, 2001] which has been lately implemented extensively in several data mining applications [Wang et al., 2017, Sun et al., 2018].

We perform experiments on a real-life large dataset consisting of approximately 32 million records, provided by a leading Dutch insurance company. First, a representative subset of the entire dataset is sampled. Then, the DBSCAN algorithm is used on this subset data to detect anomalies and generate pseudo labels. These anomalous observations are thoroughly examined by the domain experts to validate their suspicious behavior. Then, flags are created to identify the anomalous observations, detected by DBSCAN, which essentially becomes the training data to predict anomalies in the remaining data. Then a LightGBM model training is accomplished based on the previous training dataset to recognize the patterns for anomalous behavior. This distinctively enables us to predict the anomalies in the other part of the data. In our experiment, our framework captures 90% and 96% anomalous observations only by investigating 5% and 10% of the data respectively.

In this chapter, a novel semi-supervised machine learning framework based on DBSCAN and LightGBM is proposed to detect anomalies in big accounting data. This study further contributes to introducing pseudo-labeling in the audit practice. The anomaly detection framework relies on choosing a small representative subset of entire data which makes the architecture scalable in nature. Hence, it can be implemented computational cost-effectively if required, so the domain experts can investigate the detected anomalies to understand their nature. Moreover, the proposed framework is independent of accounting assumptions, hence, it holds the potential of wide-spread applicability in detecting anomalies in any unlabeled data.

The rest of the chapter is structured as follows: Section 2.2 describes the background and related work, Section 2.3 presents the framework of the semi-supervised model, section 2.4 describes the experimental setup of this study, section 2.5 discusses the experimental results, section 2.6 demonstrates the application of our proposed method beyond accounting data and finally Section 2.7 describes the conclusion and future work.

2.2 Background and Related Work

The term Anomaly refers to "pattern in data that do not conform to a well-defined notion of normal behavior" [Chandola et al., 2009]. Anomalies occur for several reasons such as different classes of data, natural variation in data or measurement or possible collection error [Tan, 2018]. These anomalous observations often deviate to a large extent indicating a possibility of different mechanisms [Hawkins, 1980]. Although the anomalous observations are not necessarily harmful [Agrawal and Agrawal, 2015], their detection is crucial and detailed investigations are required to understand their nature. In the current literature, a straightforward approach to detect anomalies is to define a region representing the normal behavior of the data and then declare any observation to be the anomaly if it fails to fall into the normal region [Chandola et al., 2009].

Anomaly detection has several applications in real life. It has been extensively used to identify possible intrusion attempts by monitoring network traffics and server applications [Portnoy, 2000, Garcia-Teodoro et al., 2009]. Anomaly detection as a form of financial fraud detection [Bolton et al., 2001] is also very popular where it helps detect fraudulent accounts by inspecting transaction data. Medical science has also adapted anomaly detection in several contexts, for example, it can be implemented in patient monitoring to detect acute medical conditions [Lin et al., 2005]. Apart from these, anomaly detection has been used in climate study [Çelik et al., 2011], geophysical signal processing [Chang and Chiang, 2002] studies, etc. Anomaly detection techniques can be classified into three categories depending on the presence of labels:

Supervised technique: This technique can only be used when historically labeled data is available. A classifier [Duda et al., 2012] is first used to train a model on this historical data in order to recognize the patterns in the data and that can then be used to predict anomalies in the future dataset. However, the initial anomaly flags or labels in the historical data are not always present in the practical settings. Although auditors can leverage their experience to go through the observations one by one and flag them whether they are anomalous or not; the datasets are often too large (compared to the number of anomalous observations) and that therefore causes extreme difficulty in generating all the necessary labels. Moreover, supervised algorithms are risky to implement for detecting anomalies as they assume that the labels are correct in the data [Goldstein and Uchida, 2016].

Unsupervised technique: Unsupervised algorithms are adapted when historically labeled data is unavailable and spatial groupings can be created in the dataset to detect anomalies. Clustering [Jain and Dubes, 1988] is one of the famous unsupervised techniques where similar observations belong to a single cluster and dissimilar observations belong to different clusters. There are many clustering algorithms that do not force every observation into a cluster, such as DBSCAN [Ester et al., 1996], ROCK [Guha et al., 2000], SNN clustering [Ertoz et al., 2004]. The observations which do not fall under any cluster or which are far away from the centers of the spatial clusters are usually identified as anomalies. However, unsupervised clustering algorithms mostly deal with the distance between two observations which leads to the problem of "curse of dimensionality" [Bellman, 1966] in a big data setup. Additionally, optimizing clustering algorithms being computationally expensive [Kim, 2009], its implementation for a big accounting data becomes difficult.

Semi-supervised technique: Semi-supervised learning is an intermediate/hybrid technique, often used where the entire data consists of two parts: a labeled and a non-labeled part [Chapelle et al., 2009]. As it is often very difficult to obtain labels in a large data set, while unlabeled data are abundant, the semi-supervised machine learning technique has proved to be an efficient solution with reduced human labor and improved accuracy [Zhu, 2005]. Several authors have therefore proposed the semi-supervised learning methods by training both the supervised and unsupervised modeling architectures simultaneously. First, a trained unsupervised model is used on unlabeled data to identify the high probability observations of being anomalous and pseudo-label them as true labels. Then a supervised model is developed to analyze the extended data based on the pseudo labels. This technique of pseudo-labeling has shown significant potential to improve the prediction results [Lee, 2013].

Assuming that the normal observations lie close to the cluster centroid and anomalous observations stay far away from their nearest cluster centroids [Tan, 2018], certain anomaly detection techniques have been proposed earlier. Smith et al. [2002] studied Self-organized maps, K-Means clustering, and Expectation Maximization to cluster training data and then use the clusters to classify the test data in a semi-supervised fashion. Self-organized maps [Kohonen, 1997] have also been widely used in semi-supervised techniques to detect anomalies in several applications including fraud detection [Brockett et al., 1998], intrusion detection [Ramadas et al., 2003] etc. Görnitz et al. [2013] suggested a new semi-supervised framework by reshaping an unsupervised task for anomaly detection which allows the inclusion of expert knowledge.

Detecting anomalies in accounting data has also been studied by several researchers in the accounting domain [Amani and Fadlalla, 2017]. Bay et al. [2006] proposed a system for identifying suspicious general ledger accounts using feature engineering based on the Naive Bayes classifier. Khan et al. [2010] suggested an unsupervised graph-based approach to create clusters concerning transaction profiles to detect suspicious user behavior. Their experiment was based on data set with approximately 300,000 records. Jans et al. [2010] experimented with univariate and multivariate latent class clustering on approximately 34,000 purchase order transaction data. Transactions diverting away from clusters were flagged as anomalous. Thiprungsri and Vasarhelyi [2011] used K-means clustering to detect anomalies in accounting data with approximately 40,000 records. Argyrou [2012] used Self-organised maps to detect anomalies in approximately 25,000 journal entries of a shipping company. Recently an autoencoder based semi-supervised approach to detect anomalies in journal entries was proposed by [Schreyer et al., 2017]. Their study was based on a dataset of approximately 300,000 records.

According to the previous research studies, the limitations of the supervised modeling technique seemed to be rather evident due to the absence of labels. The unsupervised algorithms were also found to be insufficient to optimize and train on large data due to the computational complexity. On the other hand, a semi-supervised modeling framework has shown its potential in detecting anomalies in a large accounting data set which can help the auditors in prioritizing their investigations on suspicious records in the accounting data and help producing more accurate financial statements of the organization. The possible efficiency and accuracy offered by this technique thus provided the necessary motivation behind this research.

2.3 Anomaly Detection Framework

In this section, the primary elements of the modeling architecture are discussed. The task of anomaly detection in an unlabeled dataset $X_{m,r}$ consisting of r numerical features $\{v_1, v_2, \ldots, v_r\}$ and m observations is considered.

2.3.1 Pseudo-Labeling using DBSCAN

A representative subset (X_1) of p observations is sampled from the aforementioned dataset X as: $X_{1(p,r)} \subset X$ and we define $X_{2(m-p,r)}$ as $X_2 = X \setminus X_1$, where $p << m$. In order to achieve the representativeness, distributional similarity for each v_i in X and X_1 should be tested. Performing Kolmogorov-Smirnov two-sample test [Smirnov, 1939] for each v_i from X and X_1 must yield a p-value more than 0.1 for each $i \in \{1, \ldots, r\}$.

Pseudo labels of anomalies on X_1 are generated by applying the DBSCAN [Ester et al., 1996] algorithm. DBSCAN is a density-based spatial clustering technique with the application of noise. It forms clusters based on spatial density. Observations that are not part of any cluster are defined as noise and we treat them as anomalies. DBSCAN has three parameters, epsilon (ϵ), MinPts (n) and distance metric. Two observations are called neighbors if they are within ϵ distance of each other. A cluster is a collection of minimum n observations where every observation has at least one neighbor. DBSCAN constructs clusters with observations delineating these two properties. Figure 2.1 demonstrates cases where DBSCAN algorithm forms spatial clusters in a 2-dimensional space and helps to detect anomalies.

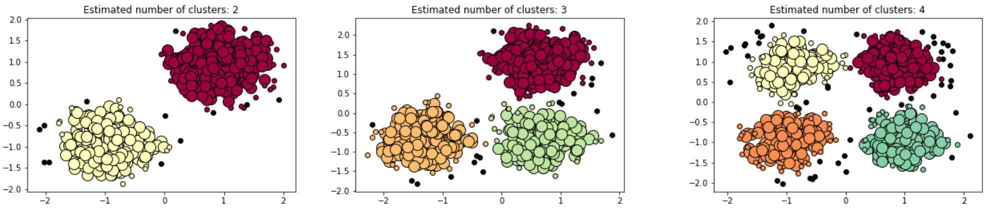


Figure 2.1: DBSCAN ($\epsilon = 0.3$, n=10) creates spatial clusters and detects anomalies using Euclidean distance in the above three simulated data space with 750 observations. Black points are considered as outliers as they are not part of any cluster. In the first example, DBSCAN detects 2 clusters and 12 anomalies, in the second example, it detects 3 clusters and 18 anomalies and in the last one, it detects 4 clusters and 47 anomalies.

It is essential to scale all the variables in X_1 before applying DBSCAN [Goldstein and Uchida, 2016]. Each of the r variables in X_1 is scaled using min-max transformation. This essentially put all the observations in an r- dimensional unit hypercube. The min-max transformation for variable $v_i \in X_1$ is defined by:

$$v_i = (v_i - min(v_i))/(max(v_i) - min(v_i)) \tag{1}$$

Parameter MinPts (n) is set to be $2r$ [Ester et al., 1996]. The optimal value of ϵ is obtained by using k-nearest neighbor plot [Rahmah and Sitanggang, 2016]. Euclidean distance is chosen as the distance metric for the task. For each of the m observations in X_1, average distance of n nearest observations is calculated and therefore m distance values are obtained corresponding to m observations in X_1. An ascending plot of these m distance values reveal a sharp elbow shape. Optimal ϵ value is set to be the distance value which corresponds to the point where the sharpness in the plot is observed.

DBSCAN with optimal parameters on X_1 creates multiple clusters and identifies anomalous observations that do not belong to any cluster. If cluster size is less than a pre-specified threshold then the observations are considered as collective anomalies [Chandola et al., 2009]. Thereafter, pseudo-labeling is performed by creating a binary variable y of length m where y_i is defined as:

$$y_i = \begin{cases} 1, & \text{if } i\text{th observation is anomalous,} \\ 0, & \text{otherwise.} \end{cases} \qquad \forall i \in \{1, \ldots, m\} \tag{2}$$

Anomalous observations as detected by DBSCAN can, therefore, be thoroughly examined by the domain experts to validate their suspicious behavior. The original values of X_1 before transforming using equation (1) is restored. Plugging in the y variable into X_1, sets the premise for supervised model to use $\{X_1, y\}$ as the training set.

2.3.2 LightGBM Model to predict anomalies

LightGBM is a popular gradient boosting decision tree algorithm which was proposed by Ke et al. [2017]. In supervised training set $\{X_1, y\} = \{(v_j, y)\}_{j=1}^{r}$, LightGBM tries to find a function $\hat{f}(v)$ which approximates $f(v)$, where $f(v)$ minimizes certain loss function [Breiman, 1997] $L(y, f(v))$ as follows:

$$\hat{f}(v) = \underset{f(v)}{\operatorname{argmin}} E_{y,v} L(y, f(v)) \tag{3}$$

LightGBM integrates T regression trees $\sum_{t=1}^{T} f_t(v)$ for approximation of the final model by:

$$f_T(v) = \sum_{t=1}^{T} f_t(v) \tag{4}$$

LightGBM uses a histogram-based split algorithm to create discrete buckets for continuous variables and grows the decision trees leaf-wise [Shi, 2007]. These help LightGBM in faster convergence and gaining high accuracy. The main parameters of the LightGBM model are:

- num_leaves: The number of leaves per tree.
- learning_rate: The learning rate of the algorithm.
- max_depth: Maximum learning depth of the algorithm, when max_depth ¡ 0 there is no limit on the learning depth.
- min_data_in_leaf: The minimum number of data in a leaf that can be used to control the fitting phenomenon.
- feature_fraction: The proportion of the selected feature to the total number of features, ranging from 0 to 1. When feature_fraction ¡ 0, the algorithm randomly selects partial features at each iteration, and feature_fraction is used to control the ratio of the total number of characteristics. This parameter can be used in order to accelerate the training speed and the control of over-fitting.
- bagging_fraction: The ratio of the selected data to the total data, ranging from 0 to 1. It is like the feature_fraction but is randomly and not repeatedly selected and must be greater than 0. This parameter can be used to accelerate the training speed as feature_fraction parameter and the control over the fitting phenomenon.
- num_trees: Number of boosting iterations or total number of trees to be formed.

To obtain the optimal parameters of LightGBM, out of fold cross validation [Kohavi et al., 1995] technique is implemented. Training set X_1 is split into 5 folds: $\{F_1$, F_2,.., $F_5\}$ using stratification of the binary label y such that:

$$F_1 \cup F_2 \cup F_3 \cup F_4 \cup F_5 = X_1 \quad and \quad F_i \cap F_j = \emptyset \quad \forall i \neq j$$

$$\bar{y_i} = \bar{y_j} \quad \forall i \neq j \quad where \quad \bar{y_i} = \sum_{y_j \in F_i} y_j / |F_i|$$

A list of potential values for each parameter is decided beforehand to obtain the optimal combination of LightGBM parameters using grid search method [Snoek et al., 2012].

ROC-AUC score is chosen for optimizing the LightGBM model. For each combination of parameters, the LightGBM model is trained on each $X_1 \setminus F_i$ set, to get the prediction for F_i for all $i \in \{1, 2, 3, 4, 5\}$. Therefore, we receive an out-of-fold prediction for the entire set X_1. ROC-AUC score is calculated using the predicted values and the original y values. An optimal combination of LightGBM parameters is selected by comparing the ROC-AUC scores for each combination.

LightGBM model with optimal parameters is trained on X_1. The trained model is used to predict the anomalies in the remaining set X_2. For each observation in X_2, the model generates the anomaly score. ROC-AUC score being scale-independent, observations with top anomaly scores become more investigation worthy. Model performance is validated by checking the overlap of true anomalous observations in X_1 with observations corresponding to top-out-of-fold predictions. Figure 2.2 illustrates the proposed semi-supervised framework for anomaly detection.

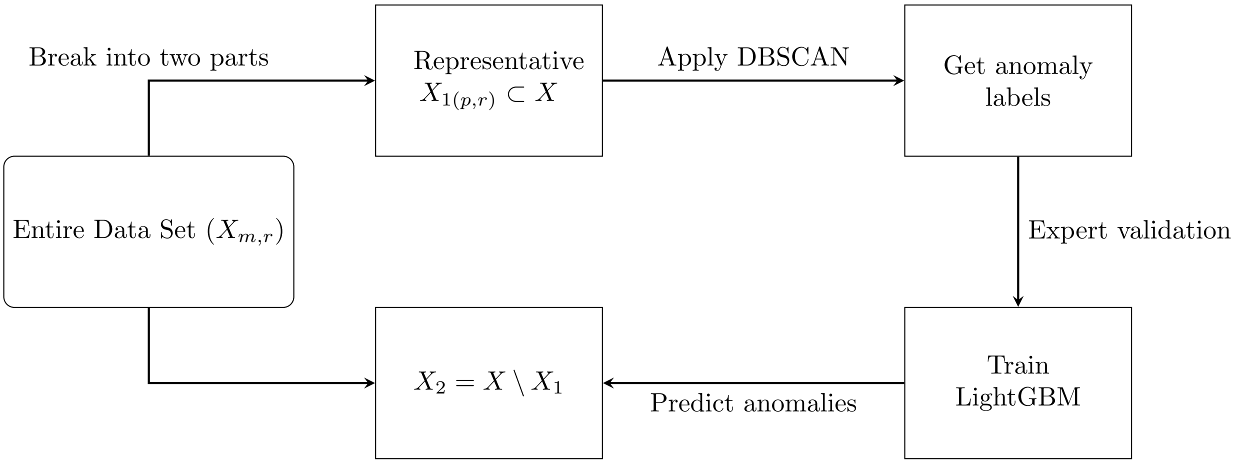


Figure 2.2: Semi-supervised anomaly detection framework based on DBSCAN and LightGBM on an unlabeled big data setup

2.4 Experimental Setup

In this section, the background of our experiment is discussed. Our study is based on insurance data from a major Dutch insurance company. It contains policy-level information for all the active policyholders. Particulars of this data are updated every month recording both changes in existing policy contracts and the addition of new policies. This database is used to prepare the financial statements of the company in every quarter. For our experiment, data spanning for one quarter has been used. It contains 31,998,736 numbers of records. After several meetings with the domain experts, nine financial variables and four non-financial variables were short-listed for our study. The variables were carefully chosen with the aim of identifying various anomalous patterns. For instance, extreme values in a variable or sensitive interactions within an observation can be easily detected, prompting further investigation. Selected variables are as follows:

1. Cost Loading Ratio: (gross premium/net premium)-1 , where the gross premium is the total premium amount (charged to the policyholder) including the cost of maintaining the corresponding policy and net premium which is the total premium amount excluding the cost of maintaining the corresponding policy.

2. Yearly Net Premium: Yearly net premium to be paid by the policyholder for the corresponding policy.

3. Reserved Expense Cost: Amount of money company reserved for the expense of maintaining the corresponding policy.

4. Reserved Administration Cost: Amount of money company reserved for the sake of administration costs for the corresponding policy.

5. Reserved Sum Assured Policy: The reserved amount of money for paying the policyholder during the event of an insurance claim for the corresponding policy.

6. Reserved Sum Assured: The reserved amount of money for paying the policyholder during the event of an insurance claim for the corresponding policyholder.

7. Sum Assured: Total sum assured amount for the corresponding policy, where the sum assured is the amount of money that the insurance company provides the policyholder for the insured event.

8. Total Sum Assured: Total sum assured amount for the corresponding policyholder.

9. Annual Delta Sum Assured: Change in sum assured amount for the corresponding policy from the current year to the previous year.

10. Age: Age of the corresponding policyholder.

11. Policy Age: Number of days since the policy started.

12. Coverage Age: Number of days since the policy coverage period started for the corresponding policy.

13. Premium Age: Number of days since the first premium date after the latest adjustment in the corresponding policy's terms and conditions.

The first nine financial variables (1 to 9) are important to consider as anomalous observations driven by these variables can directly impact the financial statements. For example, misreported extreme values in the variables - Reserved Sum Assured and Yearly Net Premium can result in inaccurate liability and income calculation respectively in the financial statements. The last four non-financial variables (10 to 13) were also selected so that the quality of the database can parallelly be tested. For example, maintaining an inactive policy account in the database may produce unreliable financial statements.

Anomalies, driven by these non-financial variables having the potential to concomitantly impact the quality of the financial statements can be followed up at the source of recording or collecting information about these instances. Moreover, with these selected variables, we can also experiment if anomalies are triggered by some strange inequalities or sensitive interactions. For instances, observations in which policy age exceeds the age of policyholders cannot coexist, or observations in which Yearly Net Premium is significantly smaller than Reserved Expense Cost can be identified and further investigated.

2.5 Experimental Results and Discussions

In this section, the results of our experiments are discussed in detail.

2.5.1 Framework Implementation

In our setup, the dataset (X) contains 31,998,736 (m) policy records with 13 (r) variables. The representative subset X_1 of 3,200,000 (p) observations from X, is sampled, which is approximately 10% of the entire dataset. Results from two sample Kolmogorov Smirnov test (see Table 2.1) established the representativeness of X_1 with respect to X. All 13 considered variables in X_1 and X had p-value more than 0.1 for the test.

Features	K-S test statistic	p-value
Cost Loading Ratio	0.00059	0.9999
Yearly Net Premium	0.00173	0.9458
Reserved Expense Cost	0.00103	0.9999
Reserved Administration Cost	0.00140	0.9941
Reserved Sum Assured Policy	0.00226	0.7404
Reserved Sum Assured	0.00226	0.7404
Sum Assured	0.00022	0.9999
Total Sum Assured	0.00238	0.6787
Annual Delta Sum Assured	0.00319	0.3105
Age	0.00224	0.7473
Policy Age	0.00226	0.7409
Coverage Age	0.00175	0.9417
Premium Age	0.00223	0.7537

Table 2.1: Variables in X_1 and X have similar distributions as the p-values are higher than 0.1 for all the variables while tested by two-sample Kolmogorov-Smirnov test

For implementing DBSCAN, the scikit-learn library in Python [Pedregosa et al., 2011] was used and optimized the DBSCAN algorithm was optimized on X_1. All the variables in X_1 were scaled individually using min-max (1) transformation. MinPts (n) was set as 26 and Euclidean distance was used to calculate the distance between two points for DBSCAN. Optimal ϵ value was chosen from the k-nearest neighbor distance plot (see Figure 2.3) which revealed en elbow shape. The inflection point was observed at 0.095. Therefore, optimal

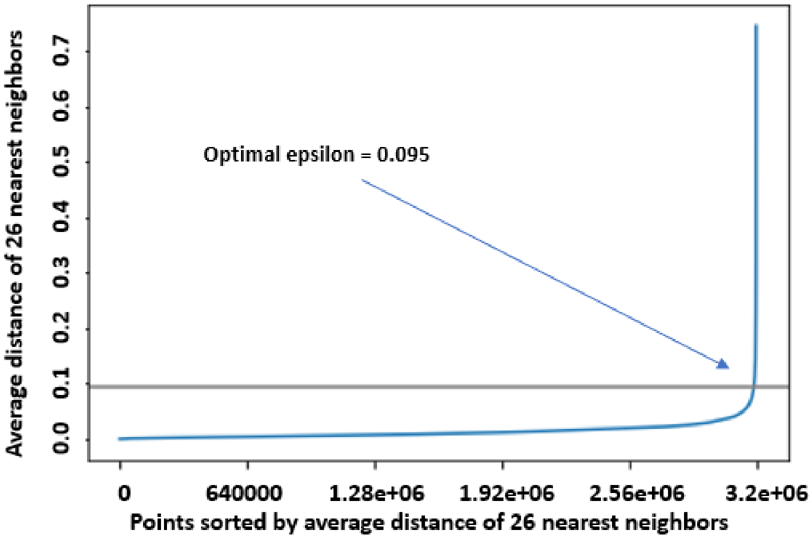


Figure 2.3: The average distance of 26 nearest neighbors for each point are calculated and their ascending plot reveals a sharp elbow shape at 0.095 which is chosen as the final ϵ parameter for DBSCAN.

value of ϵ parameter was set to 0.095.

DBSCAN ($n = 26$, $\epsilon = 0.095$, distance = Euclidean distance) was implemented on X_1. It identified twelve spatial clusters and 85,248 observations which do not belong to any cluster. These observations were labeled as anomalies. Threshold for cluster size was set to 1,600, which is 0.05% of the total number of observations in X_1. Therefore, observations within clusters with cluster size less than 1,600 were treated as collective anomalies. DBSCAN found 5,964 collective anomalies in 11 such clusters. Altogether 91,212 observations were pseudo-labeled as anomalies (see Table 2.2) in X_1.

These detected anomalous observations must thoroughly be investigated by the auditors for validation. In addition to these, a randomly selected 100,000 observations were selected from cluster 1 to check their non-abnormality to reduce the true negative error. After thorough investigations of the observations shared with the domain experts, some expected typical instances of these anomalous observations were as follows:

- Anomalies with extreme values: After the investigations, a set of anomalous observations were found to have extreme values. For example, observations with negative Cost Loading Ratio, observations with Cost Loading ratio as high as 179, observations

Clusters	No. of observations	labeled as
Cluster 1	3108788	0
Cluster 2	1104	1
Cluster 3	1017	1
Cluster 4	975	1
Cluster 5	801	1
Cluster 6	642	1
Cluster 7	501	1
Cluster 8	402	1
Cluster 9	336	1
Cluster 10	108	1
Cluster 11	42	1
Cluster 12	36	1
Anomalies	85248	1

Table 2.2: Cluster sizes with their corresponding labels for observations.

with age more than 120 years, etc.

- Anomalies with strange inequality: We examined that the collective anomalies mostly possess strange inequalities. For example, policies where Coverage Age is more than Policy Age, Policy Age is more than the age of the policyholder, etc.
- Anomalies with sensitive interactions: Another typical type of anomalies in the data could be observations with sensitive interactions. For example, observations with high Annual Delta Sum Assured values and low Yearly Net Premium which is very unusual to co-exist.

We created binary y variable in X_1 based on equation (2) which gave us the training set for LightGBM model. To optimize the LightGBM parameters, the following parameter space was searched:

- num_leaves: {15, 31}
- max_depth: {-1, 5}
- min_data_in_leaf: {40, 60, 80}
- feature_fraction: {0.6 , 0.8}
- bagging_fraction: {0.6, 0.8}

The learning rate was fixed as 0.1 for the entire experiment and the ROC-AUC score was optimized by the model. Altogether 48 parameter combinations were searched using out of fold validation score. The top 5 parameter combinations are presented in Table 2.3. The final parameters for the LightGBM model were as follows:

- num_leaves: 31

- max_depth: -1
- min_data_in_leaf: 40
- feature_fraction: 0.6
- bagging_fraction: 0.6
- num_trees: 100

LightGBM model with optimum parameters was trained on X_1. The best combination of parameters produced a high ROC-AUC score of 0.9945 in the out-of-fold prediction. Predicted anomaly scores for each record and their corresponding labels were compared. Predicted anomaly scores were split into 20 quantiles. The top quantile (95th - 100th) captured 83,114 anomalous observations, which is approximately 90% of all anomalous observations in X_1. Observations with top 10% anomaly scores (top 2 quantiles, that is 90th - 100) captured 87,631 anomalous observations which are approximately 96% of all the anomalies. The result shows that LightGBM with optimized parameters could mimic DBSCAN in X_1. The trained LightGBM model was used to predict the anomaly score for the rest of the data (X_2).

LightGBM having a tree-based architecture, we extracted the classifying rules of all the trees in LightGBM which helped fast-tracking our investigations. The performance of the framework was evaluated by investigating the 5% subset data of X_2, corresponding to the top 5% anomaly scores as predicted by LightGBM. Assuming the representativeness of X_1 and X, we consider this performance as quite successful and value-adding.

2.6 Benchmark Evaluation

Our choice of selecting the LightGBM model was challenged by several classifiers. We compared the performance of LightGBM with XGBoost [Chen and Guestrin, 2016], SVM [Suykens and Vandewalle, 1999], Multilayer perceptron [Friedman et al., 2001] and Random Forest [Breiman, 2001]. Our evaluation criteria was the out of fold ROC-AUC score. Parameters of all the models were tuned using 5 fold cross-validation setup using grid search. From our experiment, we observed that LightGBM outperformed SVM, Multilayer perceptron and Random Forest. The performance of XGBoost was comparable to LightGBM. However, LightGBM having better accuracy and faster training speed, it was chosen as the final classifier. Our comparison is shared in Table 2.3.

The objective of this exercise is to develop a high-performance classifier that is also capable of producing results with explanation powers. Tree-based methods are often considered to be a good option in this regard, as they produce a set of rules that can easily identify any sensitive interactions that drive anomaly detection. These rules can then be used to provide clear explanations of the results obtained and further investigations can be carried out.

Models	ROC-AUC
LightGBM	0.9945
XGBoost	0.9904
SVM	0.9861
MLP	0.9894
Random Forest	0.9833

Table 2.3: Comparison of LightGBM model with other classifiers. We compared the out-of-fold ROC-AUC score for each model. For a fair comparison, the same subsets F_i were used for this experiment.

2.7 An Excursion Beyond Accounting Data

Our anomaly detection framework is free of any accounting assumptions. Therefore, it can be deployed in other anomaly detection tasks as well. In this section, we investigate the performance of our framework on a famous publicly available credit card fraud detection Kaggle data [1]. The data contains credit card transaction amounts and 28 PCA components as features along with the fraud/non-fraud flag. Although the data is labeled, we deployed our framework as if this is unlabeled data and used the true labels to measure the performance of our model.

We sampled our representative subset X_1 using 10% observations from the data and implemented DBSCAN with optimized parameters on X_1. Thereafter, we generated the pseudo labels based on anomalies detected by DBSCAN. This gave us the training data to build our LightGBM model on labeled X_1 data to predict anomalies in the rest of the data (X_2). With optimized LightGBM parameters, we received competitive ROC-AUC scores on both X_1 and X_2. Our out-of-fold ROC-AUC score on X_1 and final ROC-AUC score on X_2 were 0.9626 and 0.9548 respectively. This performance strongly suggests that our anomaly detection framework is more generally applicable.

2.8 Conclusion and Future Work

In this study, a semi-supervised machine learning framework is presented to detect anomalies in big accounting data. This framework is based on a novel combination of an unsupervised model using DBSCAN and a supervised model using LightGBM. While auditing standards

[1] We took this data from the following source: `https://www.kaggle.com/mlg-ulb/creditcardfraud`. This data has originally been collected during a research collaboration of Worldline and the Machine Learning Group (http://mlg.ulb.ac.be) of ULB (Université Libre de Bruxelles) on big data mining and fraud detection.

require quantitative investigations of data to identify the risk of material misstatements in order to produce accurate financial reports, existing machine learning methods for anomaly detection in big accounting data using supervised or unsupervised techniques were found to be limited to implement in practice. Our proposed semi-supervised framework being scalable in nature can help auditors to detect suspicious observations in big accounting data for follow-up investigations.

Our study also contributes to introducing pseudo-labeling in accounting and audit practice. Pseudo labeling has lately received its popularity in computer vision and deep learning research [Aroyehun and Gelbukh, 2018, Ding et al., 2019]. Most of the accounting data being unlabeled, pseudo-labeling anomalies using an unsupervised anomaly detection algorithm can be of great importance to reduce the human intervention. Pseudo-labeling is introduced by implementing DBSCAN and then another state-of-the-art classifier that is LightGBM is used for the final anomaly detection.

In our experiment, our proposed modeling architecture captures 90% and 96% of anomalous observations only by investigating 5% and 10% data respectively. The classifying rules generated by the LightGBM model then contributed to speeding up the in-depth investigations of these anomalies by the auditors. While our framework enjoys a high anomaly detection rate, feedbacks received from the auditors also reveal the qualitative integrity of the architecture. Moreover, strictly from a design perspective, as our method is not confined to accounting assumptions, it holds the potential of widespread applicability in detecting anomalies in any unlabeled big data.

A limitation of this study lies in the inclusion of domain experts. Our method is dependent on their rigorous validation of the anomalies generated by DBSCAN. Additionally, a quid pro quo in our suggested method involves the subset size for DBSCAN to detect anomalies. While a larger subset can promise more precise anomaly detection, it also brings in more extensive manual labor to validate their surprising nature. Future research can address this practical issue by optimizing our framework.

3 Accounting fraud detection using contextual language learning

3.1 Introduction

Accounting fraud affects the shareholders of the fraudulent firms, as well as other participants in the capital markets. It is a widespread problem that causes significant damage in the economic market. However, detecting accounting fraud in a timely manner is extremely difficult, because it requires significant effort of regulators, and this takes time and considerable financial resources. Even when such fraud has been detected, that is often after the damage has already been done, such as in well-known examples of firms as WorldCom and Enron, which finally resulted in multi-billion losses for shareholders and many employees that lost their jobs. Therefore, detecting and preventing accounting fraud is a topic of great importance to regulators, investors, financial analysts, and auditors. While extensive research has been done to detect accounting fraud using quantitative information from the financial statements [Bao et al., 2020, Cecchini et al., 2010a, Dechow et al., 2011, Perols et al., 2017], recent studies based on textual analysis revealed that there are clues present in financial reports that can be analyzed to predict the likelihood of fraud.

However, the literature on the use of textual information from financial reports in order to detect accounting fraud is still scarce. Most of these studies focus on the investigation of the communication style used in the financial texts by capturing the tone or sentiment of the narratives [Goel et al., 2010, Purda and Skillicorn, 2015]. Recent studies also started to investigate the underlying topics mentioned in the texts that would indicate the possibility of misreporting [Brown et al., 2020, Minhas and Hussain, 2016].

While prior literature helps understand the facets of fraudulent texts, contemporary research argues that commonly used linguistic measures cannot adequately capture the context of management disclosures [Bushee et al., 2018, Loughran and McDonald, 2011, 2016], thereby limiting the inferences drawn from these measures. For example, when using *bag of words* methods such as LDA (Latent Dirichlet Allocation) for text analysis, the order of words in the text is not taken into account, therefore making the performance of such methods invariant to word permutations within each document. In contrast to this, the contextual analysis also encompasses the information on surrounding conditions and environment which are improving the understanding of the context surrounding the text. Moreover, Loughran and McDonald [2016] and Pratt [2015] called for research on the possibility of using deep learning based methods, where machine learns from enormous cloud-based data sets in order to capture the deeper meaning of the business text. However, the empirical research is still scarce. Therefore, we investigate whether a machine learning model that learns from the contexts present in the financial reports improves accounting fraud detection beyond what can be achieved by existing textual and quantitative models. Specifically, we address the following research questions:

RQ1: Does contextual learning from financial reports improve accounting fraud detection, relative to extant textual methods?

RQ2: How does contextual information supplement information obtained from existing quantitative methods?

We use Bidirectional Encoder Representations from Transformers (BERT) [Devlin et al., 2018] in order to detect fraudulent firms. BERT is a neural network model that is designed to learn the context of a language using textual inputs. It has been extensively employed in language-specific studies, including automation of question answering [Alberti et al., 2019, Yang et al., 2019] and language translation [Pires et al., 2019, Zhu et al., 2020]. In our study, we implement the BERT model in order to learn and capture the underlying contexts in the texts of financial reports. Finally, we train it to classify fraudulent firms' disclosures.

Our data is based on texts from the Management Discussion and Analysis (MD&A) section of annual 10-K reports from the Securities and Exchange Commission's (SEC) database. This section is commonly used by investors, and is recognized in the literature as an instrument that signals financial distress to investors [Holder-Webb and Cohen, 2007]. To address our research questions, we use two models from the previous literature as benchmark models, namely textual and quantitative benchmark model. We fine-tune the BERT model and also construct the ensemble model based on quantitative data from financial statements, along with textual data. We show that our final model outperforms the textual benchmark model and the quantitative benchmark model from the previous literature by 15% and 12%, respectively.

Since it is, because of time and financial constraints, unrealistic for regulators and corporate monitors to investigate all publicly traded firms for accounting fraud, we measure how many fraud samples are being captured in the top 1% predicted firms (highest likelihood of being fraudulent). Finally, our model identifies five times more fraudulent firms than the textual benchmark and three times more than the quantitative benchmark upon investigating the same number of firms. We, therefore, conclude that our model has a higher economic significance than the models used in the previous literature. We also perform a battery of robustness tests, to increase the confidence in our findings.

In summary, our main contributions are the following. First, we apply and fine-tune the BERT model to the accounting field and employ it in order to detect accounting fraud in publicly traded U.S. firms. Second, we show that context in the financial texts contains important information that helps detect accounting fraud. Our final model outperforms the textual benchmark model and the quantitative benchmark model from the extant literature by 15% and 12%, respectively. Third, we detect how many fraudulent firms are in the top 1%, and show that our model identifies five times more fraudulent firms than the textual benchmark by investigating the same number of firms, and three times more than the quantitative benchmark.

The rest of the chapter is organized as follows. In Section 3.2, we first introduce the related work. Then, we discuss our data construction and present the experimental setup in Sections 3.3 and 3.4. In Section 3.5, we present our results, and in Section 3.6 we show supplementary analysis. Finally, we discuss future research and conclude the chapter in Sections 3.7 and 3.8.

3.2 Related work

Over the last decade, researchers explored the predictive potential beyond the quantitative information of financial statements to predict financial anomalies like misreporting and bankruptcy. The main premise of this literature is to find patterns in the textual communication to describe managers' deliberate attempts to manipulate reporting. Rogers et al. [2011] examined the disclosure tone and shareholder litigation using firms' earnings announcement. [Larcker and Zakolyukina, 2012] developed a linguistic model to detect deceptions in earnings conference calls. We review studies that focus on exploring the textual contents of companies' financial reports to investigate financial irregularities. Readers can also consult [Loughran and McDonald, 2016] for an extensive literature review that covers advances of textual methods in the accounting field until the year 2016.

One stream of research examines the ease of reading financial texts. Li [2008] implemented the FOG index to measure the readability score of annual reports and found that firms with higher readability scores have more persistent positive earnings. Goel and Gangolly [2012] argued that the likelihood of fraud increases with the increasing complexity of sentences present in the financial reports. Goel et al. [2010] found that fraudulent annual reports contain more passive-voice sentences and are more difficult to read than the non-fraudulent annual reports. Similarly, Humpherys et al. [2011] investigated linguistic cues from financial disclosures, discovering that fraudulent disclosures use various cues such as activation language, imagery, and words, but less lexical diversity than the non-fraudulent statements.

Further, studies focus on deriving numerical features from the textual contents of financial reports and use them in a classifier to train a fraud detection model. Goel et al. [2010] extracted features from annual 10-K filings using the bag-of-words approach and implemented an SVM (Support Vector Machine) model to detect fraudulent activities. They also developed a list of detrimental words in the financial texts that help in separating fraud from non-fraud firms. Cecchini et al. [2010b] used TF-IDF features from the MD&A section of annual 10-K reports developing an SVM model for predicting financial frauds and bankruptcy events. Purda and Skillicorn [2015] studied the temporal change of annual and quarterly financial narratives using 200 most predictive words as features in an SVM model. Minhas and Hussain [2016] compared several classification algorithms after extracting n-gram features from narrative sections of annual 10-K reports. They also compared text readability tools for potential feature extraction from the documents.

Some studies particularly focus on finding patterns in topics and word combinations in order to investigate abnormal behavior. Moffit et al. [2010] derived the lexical bundles that are most frequently present in the management discussion and analysis section of the annual 10-K reports. Loughran and McDonald [2011] created a new financial dictionary and found that negative language in financial reports is associated with accounting misconduct. They developed a list of negative words that can be used to comprehend the tone and the sentiment of the annual 10-K reports. Hoberg and Lewis [2017] deployed topic modeling using LDA (Latent Dirichlet Allocation) to find that fraudulent managers use abnormal verbal tone while writing financial disclosures. Brown et al. [2020] studied the incremental contributions of thematic contents of financial narratives using topic modeling. Both Hoberg and Lewis [2017] and Brown et al. [2020] composed extensive lists of topics that help in detecting accounting frauds.

Recently, Craja et al. [2020] proposed a deep learning based approach to detect accounting frauds. Their study uses Hierarchical Attention Network (HAN) which utilizes structured hierarchy of MD&A sections. The model also allows for the use of attention mechanisms on both word and sentence levels, thereby providing indicators that could help stakeholders identify whether further investigation is needed. We build on the previous literature, and use a transformers based BERT model which is designed to capture contextual aspects from the text. Additionally, our study aims to provide pragmatic contributions by evaluating the model by practical measures such as Normalized Discounted cumulative gain (NDCG@k) and comparing the economic significance of the predictions.

Despite the current improvements in the natural language processing that could help understand the underlying communication style in financial statement texts, the majority of previous studies leverage either word categorizations (or dictionaries) or the discrete topics in the texts. There have been few studies concerning understanding the deeper meaning of the narratives and preserving the context of the writing. Our aim is to contribute to the existing literature by applying a model that could capture the contexts in the financial reports, and utilize this model in accounting fraud detection.

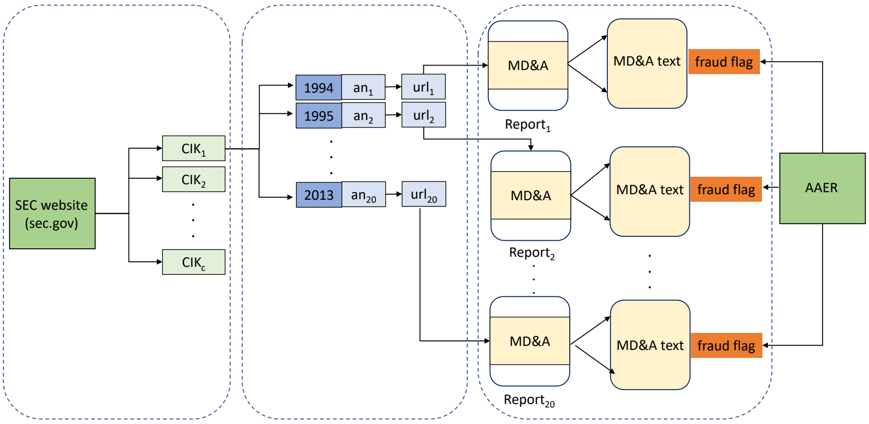


Figure 3.1: Data collection process.

3.3 Data and Sampling Design

Our experiment is based on texts from annual 10-K reports issued by U.S. firms between years 1994 and 2013. We retrieve texts from Item 7, namely the Management Discussion and Analysis (MD&A) section of annual 10-K reports from the Securities and Exchange Commission's (SEC) EDGAR database and parse the information from Item 7 into a machine-readable format. We use MD&A section to extract texts because analyzing the content of this section is a common practice for investors seeking informational advantage [Bryan, 1997, Durnev and Mangen, 2020, Loughran and McDonald, 2016, Muslu et al., 2015]. Holder-Webb and Cohen [2007] indicate that the MD&A section of 10-K reports is officially recognized as a source that contains valuable information which signals financial distress to investors. Additionally, since the contents of the MD&A section is unregulated and unstructured, but highly informative about the firm [Feldman et al., 2010], we develop a contextual machine learning model that analyzes information content in the MD&A section.

The fraud indicators used in this chapter are derived from the detected material accounting misstatements disclosed in the SEC's Accounting and Auditing Enforcement Releases (AAERs) provided by the USC Marshall School of Business (previously Berkeley Center for Financial Reporting and Management - CFRM) [Dechow et al., 2011]. Recent literature identifies this as a leading database that contains a comprehensive list of accounting fraud cases [Karpoff et al., 2017]. Other terms such as earnings management, manipulation, and misstatements, are often used interchangeably, even though the SEC often implies fraud in their allegations. Some important misstatement indicators identified by the SEC include: misstated revenue, misstatement of other expense/shareholder equity account, capitalized costs as assets, misstated accounts receivable, misstated inventory, misstated cost of goods sold, misstated reserve account, misstated liabilities, misstated marketable securities, misstated allowance for bad debt, misstated payables.

Our accounting fraud detection study is based on publicly traded U.S. firms. We construct two different datasets to address RQ1 and RQ2. To address RQ1, we construct the first dataset (referred to as the text data) based on raw texts from Item 7 of Management's Discussion and Analysis (MD&A) section of annual 10-K reports collected from the SEC EDGAR database. To address RQ2, we construct the second dataset (referred to as the ensemble data) based on features extracted from the Compustat data from the year 1994 to 2013, in order to obtain the quantitative features. We further combine these quantitative features with the text data mentioned above, and finally obtain ensemble data.

Our final datasets span from the year 1994 until the year 2013. We use 1994 as the starting year since 10-K filings are available from that year on the SEC website. Although the dataset contains all the AAERs issued concerning misreporting that happened before the end of 2016, we use 2013 as the cutoff date, since it takes multiple years for the SEC to investigate presumed accounting fraud cases [Karpoff et al., 2017]. Specifically, Dyck et al. [2010] find that the average time gap between the misreporting and initial disclosure of accounting fraud is two years. Hence, we find it necessary to limit our data until 2013. In the next two Sections, we discuss the construction procedure of text data and ensemble data.

3.3.1 Text Data Construction

The process of collecting the text data is presented in Figure 3.1. First, from the SEC website we collect the list of all CIKs (Central Index Keys), which are unique for each publicly traded U.S. firm. For each CIK, we collect annual 10-K reports filing dates for 20 years in total, spanning from the year 1994 until 2013, along with the corresponding accession numbers (an_i in Figure 3.1), which are unique for each 10-K report. For each 10-K filing, we create the URL using CIK and the accession number that leads to the corresponding 10-K report. Following the method developed by Berns et al. [Berns et al., 2021b], the text parsing algorithm searches for the term "Item 7. Management Discussion and Analysis", and any one of the phrases "the following discussion", "this discussion and analysis", "should be read in conjunction", "should be read together with", "the following management's discussion and analysis" in the following five sentences, in order to identify the beginning of the MD&A section of 10-K reports. The end of the MD&A section is determined by searching the variations of "Item8. Consolidated Financial Statements".

Next, we find the list of fraudulent CIKs using the AAER data. We find altogether 289 fraud CIKs[2], and create a binary fraud flag ($fraud = 1$ for fraudulent CIK, otherwise 0) as an input for our classification algorithms. Our final text data contains 30,876 firm-year observations with 289 fraudulent observations spanning between the years 1994 and 2013. We find that the average MD&A section contains 8,619 words, 617 sentences, and 16 words per sentence. Table 3.1 represents the yearly distribution of fraudulent observations in the text data.

Year	Total number of firms	Number of fraud firms	Percentage of fraud firms
1994	161	1	0.62%
1995	211	2	0.95%
1996	343	4	1.17%
1997	542	10	1.85%
1998	626	13	2.08%
1999	729	14	1.92%
2000	794	22	2.77%
2001	859	25	2.91%
2002	1,012	31	3.06%
2003	1,438	32	2.23%
2004	1,523	24	1.58%
2005	1,625	20	1.23%
2006	1,775	13	0.73%
2007	1,883	10	0.53%
2008	2,167	8	0.37%
2009	2,767	11	0.40%
2010	2,840	11	0.39%
2011	3,049	12	0.39%
2012	3,194	16	0.50%
2013	3,338	10	0.30%

Table 3.1: Yearly distribution of fraudulent firms in text data.

3.3.2 Ensemble Data Construction

Our ensemble dataset is based on quantitative data from the Compustat database, along with text data described in the previous section. Following earlier literature, we use a list of 28 raw financial features, as adopted in the previous research [Bao et al., 2020]. Using readily available information from financial statements helps with the simplification of the fraud detection process, since it avoids calculations of more complex accounting ratios. The 28

[2] The parsing algorithm captures MD&A text for 219 fraudulent observations. We encounter some anomalous 10-K reports where the parsing algorithm does not work, for example, Item 8 is not found in the 10-K reports, the MD&A section is wrongly listed under Item 6, etc. In order to maximize the number of fraud samples in our data, we manually capture the remaining 70 MD&A sections from 10-K reports of fraudulent companies.

financial features can be divided into four groups, based on the source of information. Those are the items that originate from balance sheets, income statements, cash flow statements, and market value items.[3]

To implement the quantitative model, we collect 28 raw financial features mentioned above, along with their corresponding CIKs, between the years 1994 and 2013. In order to test our second research question, we merge the Compustat data with the text data from Section 3.1 to obtain the ensemble data. This means that the firm-year observations in ensemble data are essentially a subset of the observations in text data[4]. Our final ensemble data contains 25,853 firm-year observations with 283 fraudulent observations. Table 3.2 presents the yearly distribution of fraudulent companies in the ensemble data.

Year	Total number of firms	Number of fraud firms	Percentage of fraud firms
1994	147	1	0.68%
1995	194	2	1.03%
1996	301	4	1.33%
1997	486	10	2.06%
1998	552	13	2.36%
1999	642	14	2.18%
2000	706	22	3.12%
2001	768	25	3.26%
2002	905	31	3.43%
2003	1,282	32	2.50%
2004	1,374	24	1.75%
2005	1,468	20	1.36%
2006	1,592	13	0.82%
2007	1,685	9	0.53%
2008	1,798	7	0.39%
2009	1,905	9	0.47%
2010	1,952	11	0.56%
2011	2,084	12	0.58%
2012	2,161	16	0.74%
2013	2,282	8	0.35%

Table 3.2: Yearly distribution of fraudulent firms in ensemble data.

[3] The information from balance sheets includes 17 variables: Current assets, Property, plant, and equipment, Accounts payable, Cash and short-term investment, Related earnings, Inventories, Common/ordinary equity, Debt in current liabilities, Receivables, Assets, Long-term debt, Current liabilities, Income taxes payable, Investment and advances, Liabilities, Short-term investments, Preferred/preference stock (capital), from income statement 7 variables: Cost of goods sold, Income before extraordinary items, Depreciation and amortization, Interest and related expense, Income taxes, Sales/turnover (net), Net income (loss), from cash flow statement 2 variables: Sale of common and preferred stock, Long-term debt issuance, and from market-value 2 items: Common shares outstanding, Price close.

[4] We merge the Compustat data and text data using CIK and year as merging keys. This results in a loss of 6 fraudulent observations in ensemble data than the text data.

3.4 Experimental Setup

3.4.1 Method

We use the BERT-Base model (uncased)[5] from TensorFlow Hub [Abadi et al., 2015] which has been pre-trained for the English language using Wikipedia and BookCorpus. Furthermore, we use the WordPiece tokenizer that creates tokens, elementary lexical components, by splitting the text into words on punctuation and white spaces, and further tokenizing words into word pieces.

Following Devlin et al. [2019], we also use a special classification token [CLS] in the beginning and a separation token [SEP] at the end of every input text sample sequence. The BERT-Base model uses 12 hidden layers of transformer blocks with hidden dimension of 768 and 12 attention heads. For each BERT-Base model, we use the maximum sequence length of 512 tokens of texts including the [CLS] and [SEP] tokens. We add an output sigmoid layer at the end of last layer of the BERT-Base model in order to establish the rank of predictions indicating the likelihood of fraud. Our search space to find optimal hyperparameters are also based on Devlin et al. [2019]'s fine-tuning strategy. While we use a fixed learning rate of 1e-5 and adam optimizer, our optimal batch size and number of epochs are found from searching within the set {1,2,..,8}.

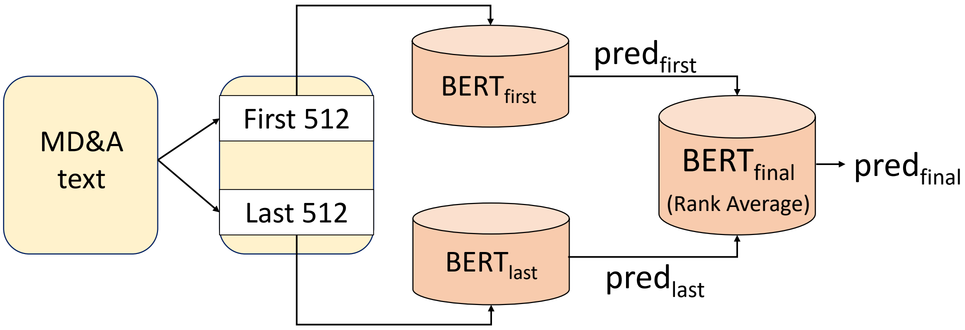


Figure 3.2: Model training from input text.

Our final model $BERT_{final}$ is the rank average of predictions from two separate fine-tuned BERT models: $BERT_{first}$ and $BERT_{last}$, where $BERT_{first}$ is trained on the first 512 tokens of each text samples and $BERT_{last}$ is trained on the last 512 tokens of each text sample. The process of training this model from two sources is illustrated in Figure 3.2. The choice of including the first and last tokens from each document is further supported by Sun et al. [2019], who show that including both the beginning and the end of an article results in

[5] https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4

lower error rates. Whereas initial tokens in the MD&A reports usually contain introductory comments, the final tokens contain concluding remarks of the MD&A section and mostly provide company's vision for the future and next steps. Since those two sources provide fundamentally different information, both of which are equally important, we ensemble predictions of these two models.

For prediction, we use rank average of predictions from $BERT_{first}$ and $BERT_{last}$ as described in Figure 3.2. For each of the 10-K reports in the set $\{Report_1, Report_2, \ldots, Report_n\}$, we first extract MD&A section. Then we use first 512 and last 512 tokens of MD&A sections as inputs to train $BERT_{first}$ and $BERT_{last}$ respectively. We obtain the $pred_{first}$ as output prediction from the sigmoid output layer of $BERT_{first}$ and obtain $pred_{last}$ from $BERT_{last}$. Our final prediction of $BERT_{final}$ is finally the rank average of $pred_{first}$ and $pred_{last}$, that is $pred_{final} = 0.5*rank(pred_{first})+0.5*rank(pred_{last})$.

3.4.2 Validation Strategy

We use rolling windows of consecutive 5 years to train our models, and the immediate following year as the test set, in order to evaluate the performance of the models. We use the period between the year 1994 and 1999 as our validation set. Specifically, to optimize parameters in the models, we initially train our models on the years between 1994 and 1998, to predict on the year 1999. Each model is trained with a different combination of parameters. After that, with the final set of parameters that produces the optimum prediction on 1999, we train our models on every 5-years data and predict accounting fraud on the immediate next year. This procedure is presented in Figure 3.3. From the Figure, it is visible that our first training period is the interval between the years 1995 and 1999, and the corresponding test set is for the year 2000. Similarly, our second training period is the interval between 1996 and 2000, and the corresponding test set is the year 2001. This procedure continues until the end of our sample, which results in 14 years of test period between the years 2000 and 2013. Importantly, this strategy is also in line with the previous research [Brown et al., 2020], which allows us to compare our model against the benchmarks established in the literature.

3.4.3 Evaluation Metrics

We use the area under the ROC curve (AUC) as our primary evaluation metric. Fraud detection models suffer from class imbalance problems, which makes AUC a reasonable metric choice, as it presents the probability that a randomly selected fraud sample would be ranked higher than a randomly selected non-fraud sample.

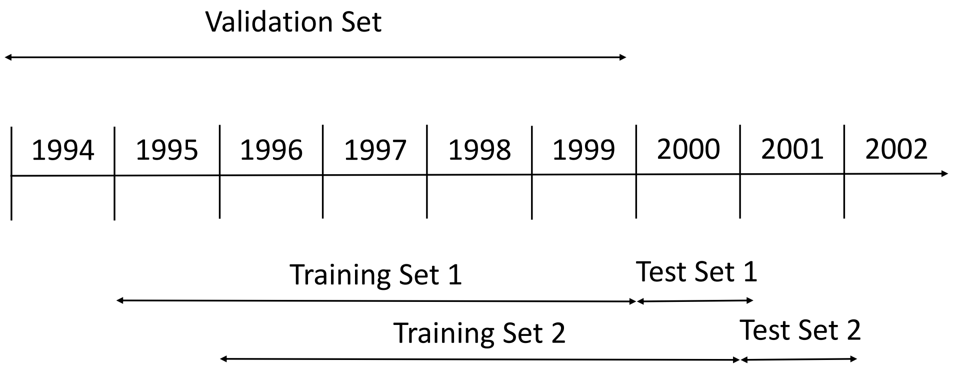


Figure 3.3: Validation strategy of our model.

As our second evaluation metric, we use Normalized Discounted Cumulative Gain at the position k (NDCG@k). Since the task of fraud prediction can also be postulated as a ranking problem, the NDCG@k provides insight into the structure of top k observations that have the highest probability of being fraudulent and that are in agreement with the original fraud samples. NDCG@k represents the ratio where a higher value represents better performance, and the measure ranges from 0 to 1. While the NDCG@k value of 1 represents that the first k observations with the highest prediction scores of being fraudulent are all true fraud samples, the NDCG@k value of 0 would indicate that none of the first k observations with the highest prediction scores of being fraudulent are true fraud samples. Throughout our study, we use 1% of firms in each test year to report the NDCG@k scores.

Because of time and financial constraints, it is unrealistic for regulators and corporate monitors to investigate all publicly traded firms for accounting fraud, we also measure in absolute terms how many fraud samples are being captured in the top 1% predicted firms (highest likelihood of being fraudulent). This metric, along with the NDCG@k measure, helps us evaluate the economic significance of the models, identifying if more fraudulent firms could be captured by investigating the same number of firms. We refer to this measure as *Capture* in Tables 3.3 and 3.4, where we present the performance results of our models.

Recent accounting fraud detection research [Bao et al., 2020, Brown et al., 2020] use AUC and NDCG@k in their studies. Hence, using those metrics as reference points provides a valid comparison of the models.

3.5 Results

To address our research questions and to provide support for our analysis, we use two models from the previous literature as benchmark models. We, therefore, compare the performance of our final models against the existing benchmark models. We use the Latent Dirichlet

Allocation (LDA) model as our textual benchmark model and the RUSBoost model as our quantitative benchmark model for accounting fraud prediction. We use LDA as our textual benchmark model since recent studies [Brown et al., 2020, Hoberg and Lewis, 2017] demonstrate that the LDA model outperforms the commonly used approach using textual style features to detect accounting frauds. On the other hand, we use the RUSBoost model as our quantitative benchmark model, since Bao et al. [2020] shows that the RUSBoost model outperforms commonly used logistic regression used by Dechow et al. [2011] and support vector machine from Cecchini et al. [2010a] to detect accounting frauds.

3.5.1 Addressing RQ1

To examine whether contextual learning from financial reports improves accounting fraud detection relative to the extant textual method, we compare our BERT models and the LDA benchmark model using the text data. The results of the calculations are presented in the Table 3.3.

First, we discuss the performance of the LDA model. Similar to Brown et al. [2020] and Hoberg and Lewis [2017], we adopted disintegrated topic features as vectors to use them in a logistic regression model as shown in Equation 1. We used Genism library [Řehůřek and Sojka, 2010] for LDA topic extraction.

$$log(\frac{fraud_i}{1 - fraud_i}) = \alpha + \sum_j \beta_j topic_{i,j} \tag{5}$$

In contrast to the BERT model, where we use the first and last 512 tokens of the MD&A text, for the LDA model we use the entire MD&A text. Before implementing the LDA model, we pre-processed the texts by performing lemmatization and removing stopwords. We optimize the *number of topics* parameter for the LDA model using the validation set. Hoberg and Lewis [2017] found 71 as the optimum number and Brown et al. [2020] found 31 as the optimum number of topics in their accounting fraud detection study. To accommodate these numbers, we searched within an interval of 10 and 150 topics and found that the optimum number of topics in our setting that maximizes the validation AUC is 78. We implement the LDA model with 78 topics and find that for the interval between the years 2000 and 2013, the average AUC is 0.720 and the average NDCG@k is 0.127. LDA model captures altogether 20 fraud samples in its top 1% predictions in 14 test years.

Next, we discuss the performance of the BERT model based on text data. We use NVIDIA TESLA P100 GPU[6] for fine-tuning the BERT models. As discussed in subsection

[6] We are thankful to the Kaggle community for providing free access to GPU. Details can be found under the following link: `https://www.kaggle.com/docs/efficient-gpu-usage`

AUC	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	Average
LDA	0.793	0.708	0.684	0.697	0.740	0.715	0.726	0.708	0.749	0.684	0.767	0.803	0.711	0.600	0.720
$BERT_{first}$	0.833	**0.810**	0.804	0.812	0.867	**0.887**	0.760	0.814	0.791	**0.736**	0.799	0.803	0.749	**0.787**	0.804
$BERT_{last}$	0.831	0.759	**0.879**	**0.896**	0.848	0.800	**0.849**	0.868	**0.858**	0.664	0.780	**0.881**	0.751	0.757	0.816
$BERT_{final}$	**0.845**	0.809	0.865	0.876	**0.871**	0.858	0.824	**0.874**	0.842	0.682	**0.818**	0.864	**0.769**	0.774	**0.826**
NDCG@k	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	Average
LDA	0.000	0.747	0.000	0.000	0.139	0.071	0.126	0.197	0.165	0.054	0.053	0.196	0.034	0.000	0.127
$BERT_{first}$	**1.000**	**1.000**	**1.000**	**1.000**	0.843	0.591	0.492	**0.616**	**0.456**	**0.539**	0.219	0.546	0.505	**0.432**	0.660
$BERT_{last}$	0.920	**1.000**	**1.000**	**1.000**	0.751	0.550	0.413	0.587	0.326	0.500	0.393	0.477	0.415	0.411	0.624
$BERT_{final}$	**1.000**	**1.000**	**1.000**	**1.000**	**0.852**	**0.637**	**0.504**	0.522	0.398	**0.539**	**0.482**	**0.592**	**0.522**	0.403	**0.675**
Capture	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	Sum
LDA	0	6	0	1	2	1	2	2	2	1	1	1	1	0	20
$BERT_{first}$	**8**	**9**	**11**	14	12	8	5	**5**	3	**6**	3	5	**6**	**5**	100
$BERT_{last}$	7	**9**	**11**	**15**	11	7	5	**5**	2	5	4	6	5	**5**	97
$BERT_{final}$	**8**	**9**	**11**	**15**	**13**	**10**	**6**	4	**4**	5	**5**	**7**	**6**	4	**107**

Table 3.3: Yearly performance of LDA and BERT models on the textual data. AUC - area under the ROC curve, LDA - Latent Dirichlet Allocation, NDCG@k - Normalized Discounted Cumulative Gain at the position k, BERT - Bidirectional Encoder Representations from Transformers, Average - average across 14 test years.

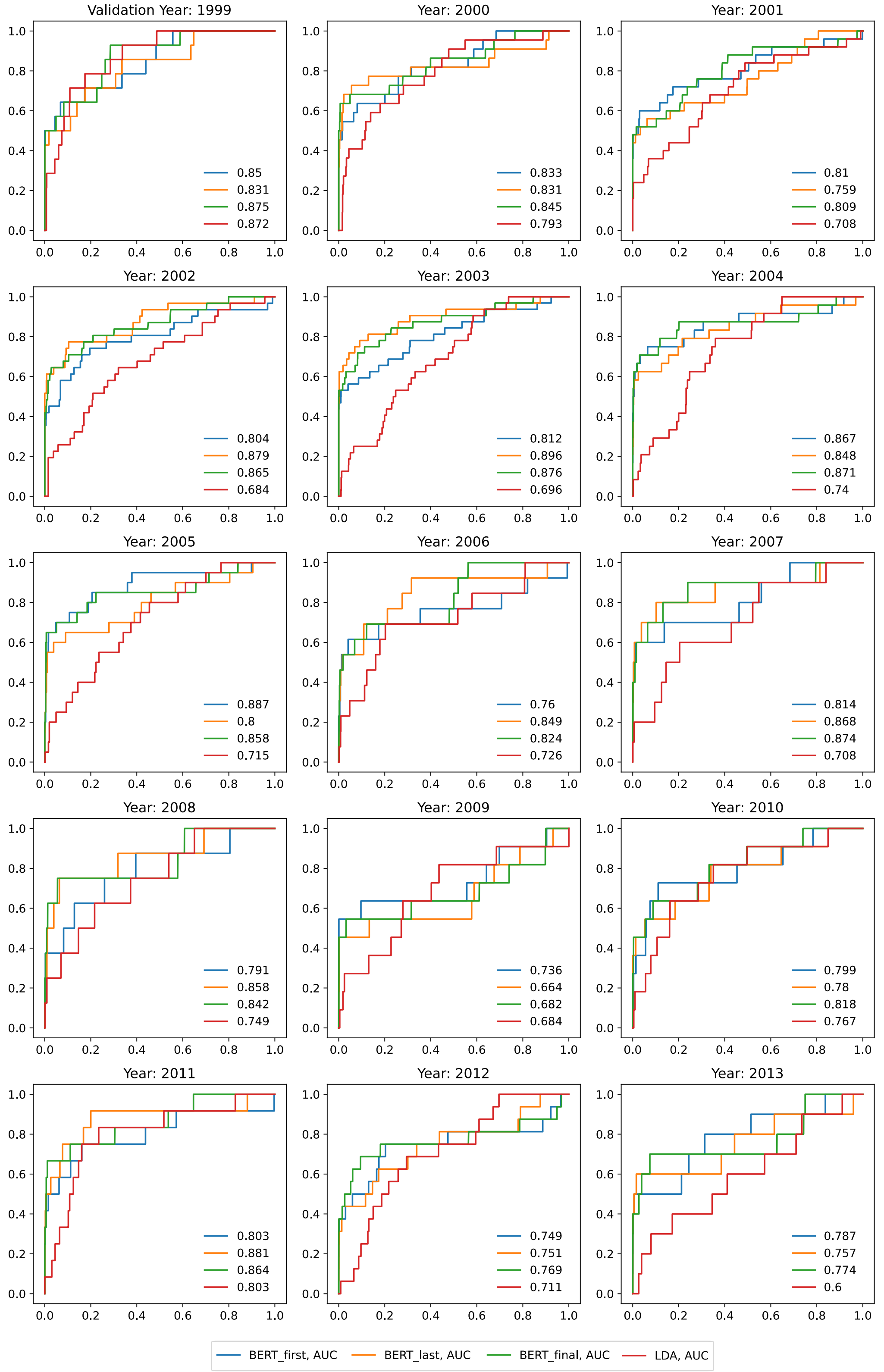


Figure 3.4: ROC curves of LDA and BERT model predictions (example year 2003). AUC - area under the ROC curve, BERT - Bidirectional Encoder Representations from Transformers, LDA - Latent Dirichlet Allocation.

3.4.1, we optimize batch size and the number of epochs using the validation set. Due to the computational constraints, we limit our search space for both the number of epochs and

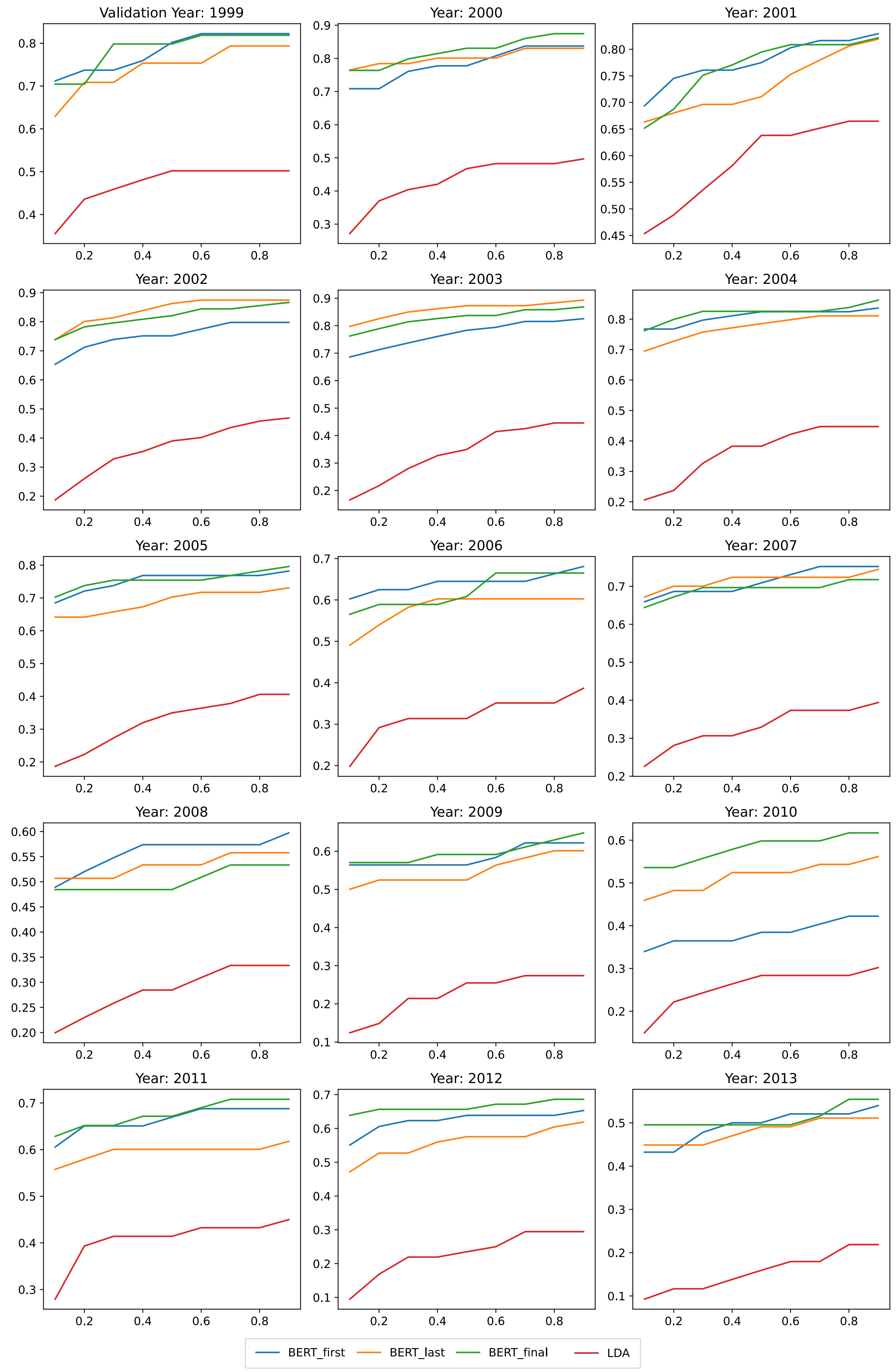


Figure 3.5: NDCG@K as a function of deciles for LDA and BERT model predictions from year 1999-2013. BERT - Bidirectional Encoder Representations from Transformers, LDA - Latent Dirichlet Allocation.

batch size within the set {1,2,..,8}. We find that a batch size of 8 and 3 epochs maximizes the validation AUC for $BERT_{final}$. In our validation set, we find that while $pred_{first}$ and

$pred_{last}$ produce AUC values of 0.85 and 0.831 respectively, their rank average $pred_{final}$ produces an AUC value of 0.875. Moreover, we find that the rank correlation between $pred_{first}$ and $pred_{last}$ in the validation set is as low as 0.191 which indicates the degree of information diversity obtained by learning from these two models. This also supports our claim in the previous section, where we conclude that the first and last parts of the MD&A section contain different information, and should therefore both be included.

We fine-tune both $BERT_{first}$ and $BERT_{last}$ with 3 epochs and a batch size of 8 to obtain predictions for each of our test years. We present the comparison of ROC curves for BERT and LDA models in the Figure 3.4, as well as comparison of NDCG as a function of deciles(k) in the Figure 3.5, which shows that LDA model does not catch up with BERT models. Generally, we find that NDCG scores for LDA model at different deciles are significantly smaller than those of BERT models. For example, from Table 3.1 for year 2002, we notice that all three BERT models capture 11 fraudulent firms (1% of total firms in the year 2002, see Table 3.1) within their first 11 highest predicted scores for the test year 2002, thus achieving an NDCG@k score of 1. On the other hand, the LDA model cannot capture any fraudulent firm in the same top 11 predictions, thereby obtaining an NDCG@k score of 0. Table 3.3 contains the results on the yearly performance of both models on the textual data. We find that the average AUC is the highest for $BERT_{final}$, and also 15% higher than the average AUC of the LDA model. $BERT_{final}$ also achieves an average NDCG@k score of 0.675 which is 5.3 times more than that of LDA. While $BERT_{first}$ and $BERT_{last}$ captured altogether 100 and 97 fraudulent firms, respectively, within their top 1% predictions in the 14 test years, $BERT_{final}$ captured 107 fraudulent firms, which is 5.3 times more than that of LDA model, thus providing evidence of the economic significance of our model over textual benchmark model from the literature.

Table 3.3 also shows the yearly performance of LDA, $BERT_{first}$, $BERT_{last}$ and $BERT_{final}$ model. We observe that the LDA model is significantly under-performing across all test years with respect to other models. Interestingly, we find that in some years $BERT_{first}$ or $BERT_{last}$ have higher AUC scores than $BERT_{final}$, such as in the year 2002, despite the improvement in the validation year 1999. However, we notice that for 11 out of 14 test years, $BERT_{final}$ produces higher NDCG@k scores and for 10 out of 14 test years it captures a higher number of fraudulent firms in the top 1% prediction. Hence, we proceed with $BERT_{final}$ as it generalizes more across the years.

To statistically test whether the differences in the average performance of other models compared to $BERT_{final}$, we conduct an analysis of variance. The test yields a p-value of less than 0.001, thereby also confirming the superiority of $BERT_{final}$ model in relation to other models.

3.5.2 Addressing RQ2

To examine how contextual information supplements the information obtained from existing quantitative methods, we first compare predictions of $BERT_{final}$ and RUSBoost model using the ensemble data. Then, we construct the Ensemble model of their predictions using rank average to understand the degree of complementarity.

Following Bao et al. [2020], we implement RUSBoost model from the imbalance-learn library [Lemaître et al., 2017] using 28 raw features as our independent variables. For the RUSBoost model, we optimize the number of trees using the validation set and find the following set of parameters that maximizes the RUSBoost AUC in the validation set. The final number of trees is set to be 2,500, the learning rate is 0.1, and we sample the same number of fraudulent and non-fraudulent observations during each iteration of the model. With this optimized set of parameters, we train the models using the same procedure as previously described in $BERT_{final}$ model using ensemble data to obtain predictions for 14 test years.

Finally, to produce the Ensemble model, we investigate the degree of complementarity that $BERT_{final}$ and the RUSBoost model share. We combine the predictions of those two models and check for overall improvement. We take the weighted rank average of these two models' prediction values for each year to obtain the Ensemble prediction. $Ensemble_{pred} = w * rank(pred_{final}) + (1 - w) * rank(pred_{RUSBoost})$. Searching from the set {0.1,0.2,...,0.9}, we find 0.5 to be the optimum value of w using the validation set that produces maximum AUC on 1999 based on $Ensemble_{pred}$. Implementing such rank average of these two models results in the Ensemble model.

We present the results on the yearly performance of RUSBoost, $BERT_{final}$ and the Ensemble model on the ensemble data in Table 3.4. RUSBoost model obtained an average AUC of 0.760 and an average NDCG@k score of 0.279 in the 14 test years. It captured in total 32 fraud samples in the top 1% predictions. The average AUC and NDCG@k of $BERT_{final}$ model in the ensemble data is 0.852 and 0.684 respectively, and it captures 96 fraud samples in the top 1% predictions altogether in 14 test years. This shows that the $BERT_{final}$ model outperforms the RUSBoost model by 12% when observing the AUC and captures three times more fraud samples in the top 1% predictions. $BERT_{final}$ also obtains 2.4 times more NDCG@k score than the RUSBoost model. The Ensemble model obtains an average AUC of 0.847 and an average NDCG@k of 0.550 and it captured 75 fraud samples in the top 1% predictions. We present the comparison of ROC curves for

AUC	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	Average
RUSBoost	0.784	0.780	0.808	0.830	0.823	0.802	0.671	0.683	0.768	0.735	0.750	0.865	0.723	0.620	0.760
$BERT_{final}$	0.840	0.804	0.863	0.874	0.868	0.853	**0.820**	**0.947**	**0.900**	**0.814**	0.822	0.867	0.765	**0.897**	**0.852**
Ensemble	**0.855**	**0.821**	**0.880**	**0.903**	**0.878**	**0.867**	0.781	0.882	0.857	0.813	**0.824**	**0.903**	**0.769**	0.827	0.847
NDCG@k	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	Average
RUSBoost	0.586	0.731	0.673	0.473	0.000	0.043	0.000	0.000	0.373	0.176	0.265	0.274	0.316	0.000	0.279
$BERT_{final}$	**1.000**	**1.000**	**1.000**	**1.000**	**0.846**	**0.663**	**0.512**	**0.575**	0.371	**0.611**	0.491	0.511	**0.522**	**0.468**	**0.684**
Ensemble	**1.000**	**1.000**	0.929	0.660	0.594	0.343	0.207	0.239	**0.586**	0.383	**0.557**	**0.607**	0.498	0.090	0.550
Capture	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	Sum
RUSBoost	3	6	5	5	0	1	0	0	2	2	2	3	3	0	32
$BERT_{final}$	**8**	**8**	**10**	**13**	**10**	**9**	**6**	**4**	**3**	**5**	**5**	5	**6**	**4**	**96**
Ensemble	7	**8**	9	7	9	5	3	**4**	**3**	2	**5**	**6**	**6**	1	75

Table 3.4: Yearly performance of RUSBoost, BERT and Ensemble model on the ensemble data. AUC - area under the ROC curve, NDCG@k - normalized discounted cumulative gain at the position k, BERT - Bidirectional Encoder Representations from Transformers, Ensemble - ensemble model based on both quantitative and textual data, Capture - the number of fraudulent firms that could be captured by investigating the same number of firms, Average - average across 14 test years.

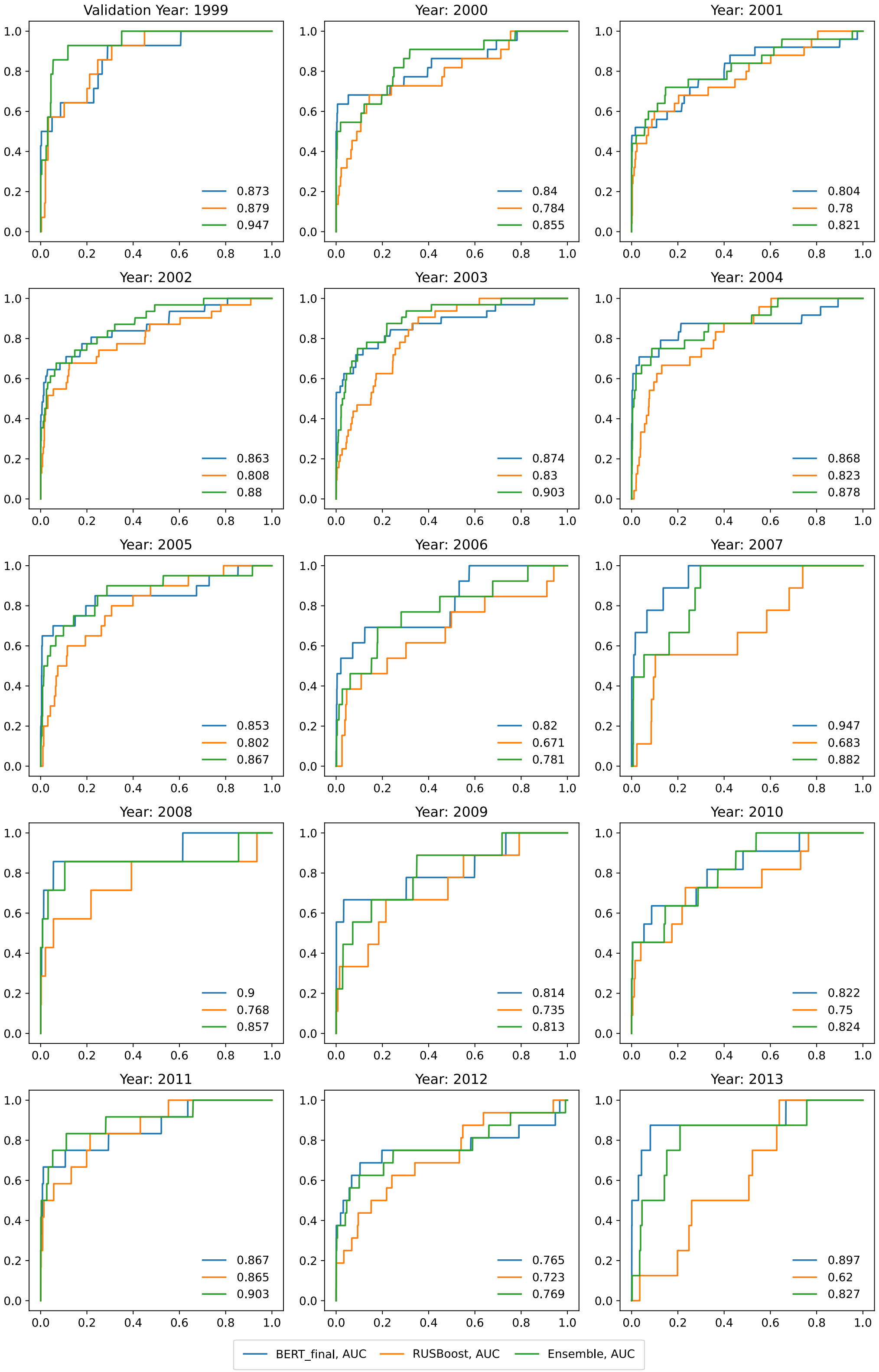


Figure 3.6: ROC curves of LDA and BERT model predictions (example year 2013). AUC - area under the ROC curve, BERT - Bidirectional Encoder Representations from Transformers, Ensemble - ensemble model based on both quantitative and textual data.

$BERT_{final}$, RUSBoost, and Ensemble models in the Figure 3.6, as well as comparison of NDCG as a function of deciles in the Figure 3.7, which shows that the RUSBoost model does

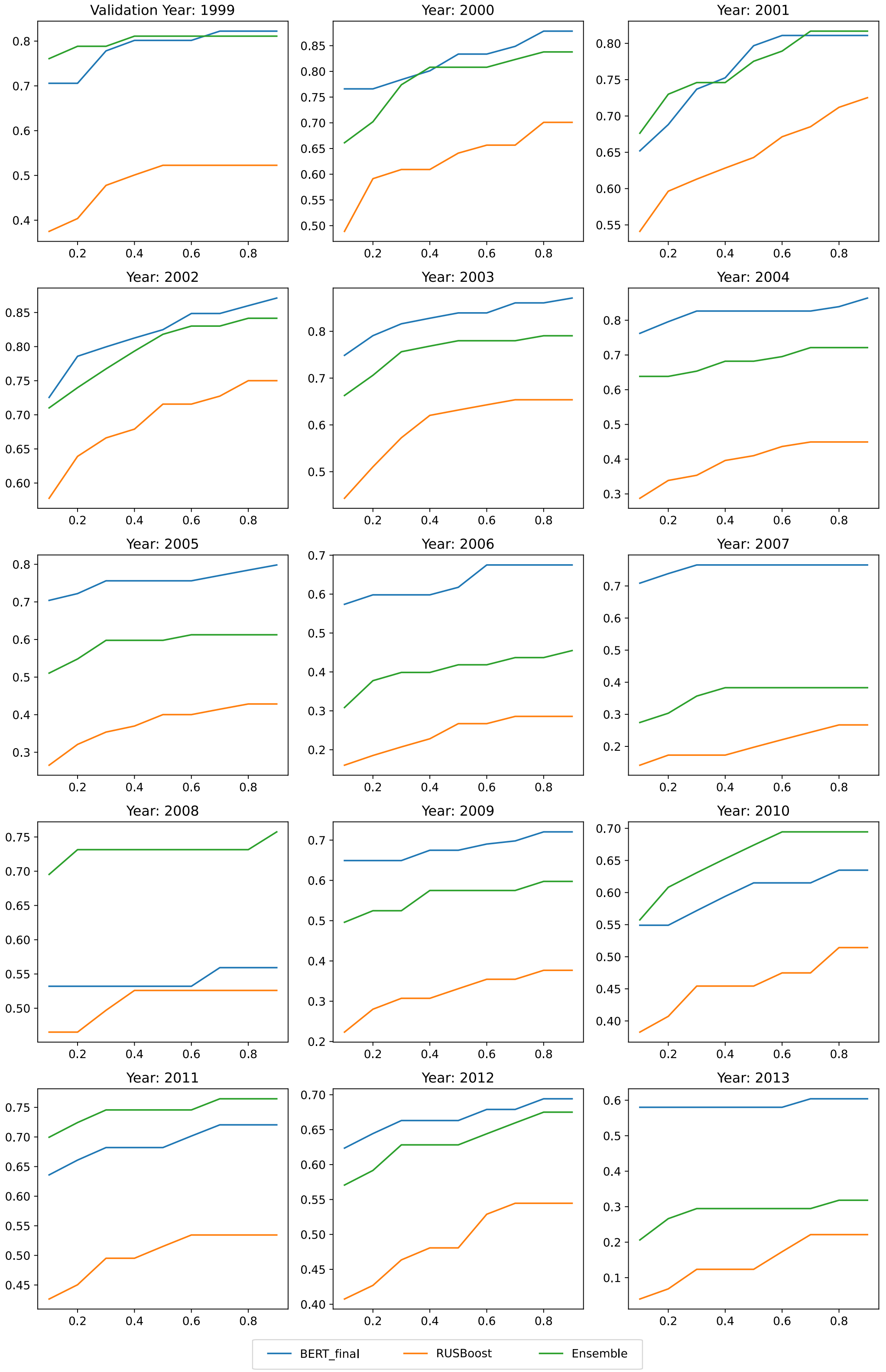


Figure 3.7: NDCG@K as a function of deciles for BERT, RUSBoost, and Ensemble model predictions from year 1999-2013. BERT - Bidirectional Encoder Representations from Transformers, Ensemble - ensemble model based on both quantitative and textual data.

not catch up with $BERT_{final}$ model. Generally, we find that NDCG scores for RUSBoost model at different deciles are significantly smaller than those of $BERT_{final}$ model.

Similarly, as in RQ1, we statistically test the differences in average performance of other models compared to Ensemble model by conducting an analysis of variance. The test yields a p-value of less than 0.001, thereby also confirming the superiority of the Ensemble model in relation to other models.

It is visible from the results that the Ensemble model obtains competitive AUC scores with respect to $BERT_{final}$, however, the economic significance of $BERT_{final}$ is higher than the significance of the Ensemble model. This shows that combining $BERT_{final}$ and RUSBoost improves the AUC score of the standalone RUSBoost model by 11%, NDCG@k score by 2 times, and captures 2.34 times more fraudulent samples in the top 1% prediction. Interestingly, the Ensemble model could not outperform the $BERT_{final}$'s performance. Although the Ensemble model achieves competitive AUC to that of $BERT_{final}$, it captures 22% less fraudulent firms and achieves a 24% lower NDCG@k score. Even though the combination of contextual and quantitative learning results in incremental improvements over the quantitative model alone, we finally conclude that the performance of the standalone contextual learning model is nevertheless higher.

3.6 Practical insights for financial investigators

In this section, we aim to provide further insights into how financial investigators can support their decisions based on the predictions obtained from $BERT_{final}$ model.

3.6.1 Insights using text data

We find that the MD&A writing style and the choice of words change dynamically over time. Furthermore, we find evidence that fraudulent firms tend to use more positive words and refrain from using negative words, possibly to disguise fraud. We further analyze the relative frequency of selection of positive and negative words from Loughran and McDonald's dictionary [Loughran and McDonald, 2011] in the MD&A reports separately for fraudulent and non-fraudulent firms across our training periods, and show that the use of positive words such as *gain*, *advances*, and *improvement* significantly increased over the years among the fraudulent firms, whereas the use of negative words such as *delays*, *excluding*, and *adverse* decreased over the years (Figure 3.8 and 3.9).

Additionally, we investigate how the $BERT_{final}$ model, developed on validation data, performs also on later test years in order to show whether the writing style evolved.[7] We find that performance decreased significantly over the years which indicates that the fraudulent firms adapt their writing style with time to produce misleading reports.

[7] $BERT_{final}$ model is developed on the period from 1994-1998, and the AUC for the test year 2000 is 0.738, for 2005 is 0.630, for 2010 is 0.609, and for 2013 is 0.522.

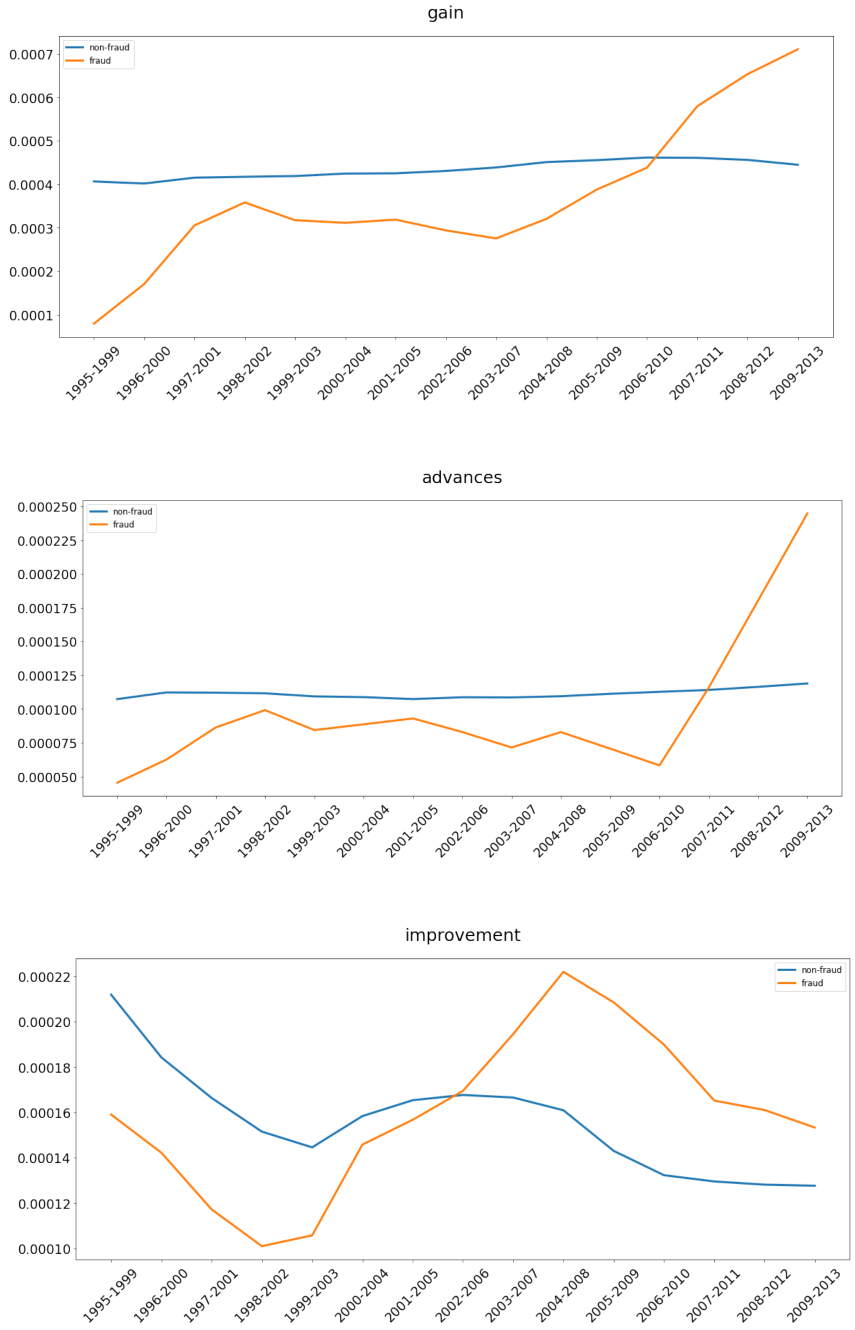


Figure 3.8: Relative frequency of selection of positive words from Loughran and McDonald's dictionary for fraudulent and non-fraudulent firms across our training periods.

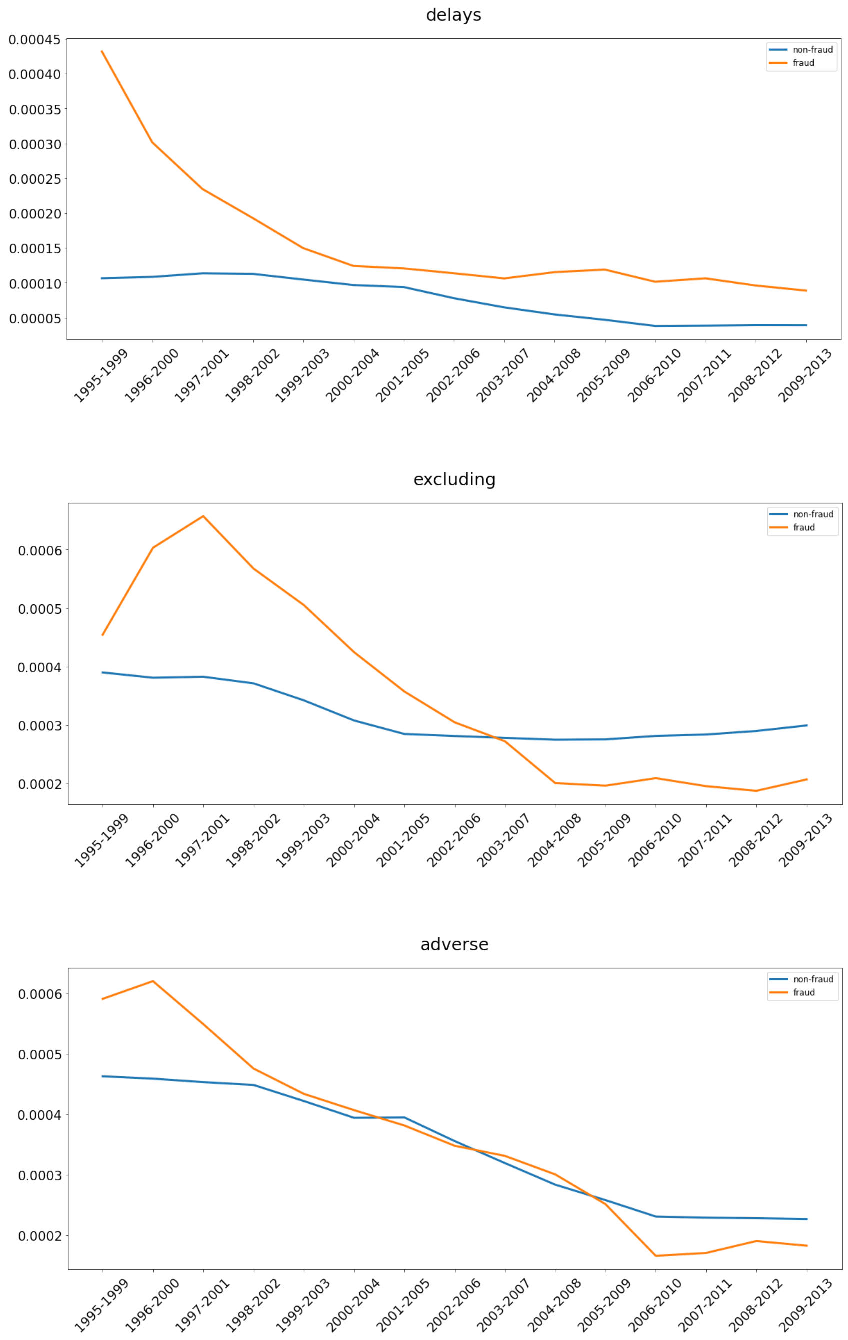
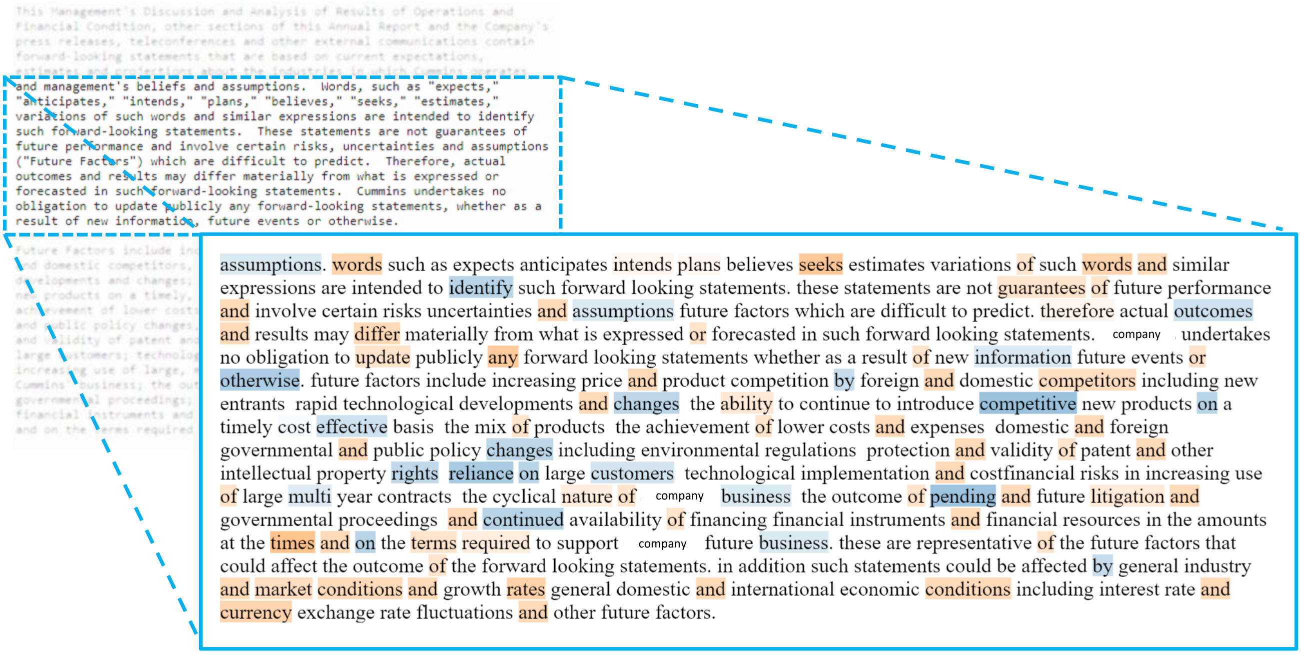


Figure 3.9: Relative frequency of selection of negative words from Loughran and McDonald's dictionary for fraudulent and non-fraudulent firms across our training periods.

This highlights the need for financial investigators to thoroughly scrutinize the signals that the model is picking to support their final decisions in prioritizing investigations. For that purpose, a model like LIME [Ribeiro et al., 2016] can be used, which can help the auditors to identify and visually present which words are playing a pivotal role for obtaining the final prediction. We present such a model on one fraudulent example in the Figure 3.10.



This Management's Discussion and Analysis of Results of Operations and Financial Condition, other sections of this Annual Report and the Company's press releases, teleconferences and other external communications contain forward-looking statements that are based on current expectations, estimates and projections about the industries in which Cummins operates and management's beliefs and assumptions. Words, such as "expects," "anticipates," "intends," "plans," "believes," "seeks," "estimates," variations of such words and similar expressions are intended to identify such forward-looking statements. These statements are not guarantees of future performance and involve certain risks, uncertainties and assumptions ("Future Factors") which are difficult to predict. Therefore, actual outcomes and results may differ materially from what is expressed or forecasted in such forward-looking statements. Cummins undertakes no obligation to update publicly any forward-looking statements, whether as a result of new information, future events or otherwise.

Future Factors include inc
and domestic competitors,
developments and changes;
new products on a timely,
achievement of lower costs
and public policy changes,
and validity of patent and
large customers; technolog
increasing use of large, m
Cummins' business; the out
governmental proceedings;
financial instruments and
and on the terms required

assumptions. words such as expects anticipates intends plans believes seeks estimates variations of such words and similar expressions are intended to identify such forward looking statements. these statements are not guarantees of future performance and involve certain risks uncertainties and assumptions future factors which are difficult to predict. therefore actual outcomes and results may differ materially from what is expressed or forecasted in such forward looking statements. company undertakes no obligation to update publicly any forward looking statements whether as a result of new information future events or otherwise. future factors include increasing price and product competition by foreign and domestic competitors including new entrants rapid technological developments and changes the ability to continue to introduce competitive new products on a timely cost effective basis the mix of products the achievement of lower costs and expenses domestic and foreign governmental and public policy changes including environmental regulations protection and validity of patent and other intellectual property rights reliance on large customers technological implementation and costfinancial risks in increasing use of large multi year contracts the cyclical nature of company business the outcome of pending and future litigation and governmental proceedings and continued availability of financing financial instruments and financial resources in the amounts at the times and on the terms required to support company future business. these are representative of the future factors that could affect the outcome of the forward looking statements. in addition such statements could be affected by general industry and market conditions and growth rates general domestic and international economic conditions including interest rate and currency exchange rate fluctuations and other future factors.

Figure 3.10: Example of implementation of LIME model on one fraudulent MD&A report. This model can help the auditors to identify and visually present words that are important for the final prediction: blue words signal less riskiness, and red signal more riskiness. The name of the company is replaced by the word *company*.

3.6.2 Insights from financial data

In order to provide insights from financial data that can be used by financial investigators, we conduct two analyses. First, we analyse which firms are easy and difficult to identify for our $BERT_{final}$ model by uncovering the factors that are driving the probability of (mis)classifications. We consider the fraudulent firms which are correctly identified as within the top 1% predictions of the model and the non-fraudulent firms which are not in the top 1% predictions to be easy to identify. Reversely, the fraudulent firms that are not in the top 1% prediction and the non-fraudulent firms which are in the top 1% predictions, we consider to be difficult to identify. In our sample, we found altogether 21,691 firms that are easy to identify and 271 firms that are difficult to identify. Next, we use a decision tree classifier, along with the financial features from the ensemble data, to extract the rules that help to improve the understanding when a firm is difficult for $BERT_{final}$ to identify. We find that firms with higher inventories and higher annual sales are difficult for our model to identify, as well as the firms with higher inventories, lower annual sales, and lower interest related expenses.

In the second analysis, we examine what factors drive a fraudulent firm to be erroneously identified by the $BERT_{final}$ model as a non-fraudulent firm and vice versa by performing a similar analysis on the set of firms that are difficult to identify. For that purpose, we consider firms which are erroneously identified by the model as fraudulent firms (firms identified by the $BERT_{final}$ model as belonging to top 1% predictions, but are in reality non-fraudulent), and the firms which are erroneously identified by the model as non-fraudulent (not in the top 1% prediction, but in reality fraudulent). We concatenate these two subsets for all 14 years of the test period and produce the data that have been difficult to identify by $BERT_{final}$ model. Set of all difficult firm-year observations produces altogether 143 fraud samples and 128 non-fraud samples. Next, we use a decision tree classifier and the financial features in order to understand what factors influence the erroneous classifications. We identify that fraudulent firms with higher annual sales, lower net income, and higher annual close price have erroneously been identified by $BERT_{final}$ model as not fraudulent, whereas non-fraudulent firms with low sales have been wrongly identified by the model as high-risk firms. These inferences can be further applied in making decisions by the financial investigators to prioritize their investigations. Investigators can use the first analysis to understand which firms could be difficult for the model to identify, and then on that set that is difficult to identify, the second analysis can be performed to better understand how to prioritize the investigations. Both analyses described above are presented in Figure 3.11.

3.6.3 Attention mechanism

Recent literature has started to explore how attention mechanisms can be presented visually and interpreted in NLP tasks. For that purpose, a model like BertViz [Vig, 2019] can be used, which visualizes attention weights and the color of the connection represents a stronger or weaker focus on the relevant words. One important aspect of attention is that every time the model tries to predict a missing word in a sentence, it focuses attention to some specific words which are contextually relevant in that sentence rather than concentrating on the entire sentence. In Figure 3.12, we present a practical example from our 10-K sample in order to demonstrate how BERT operates in financial text and where it directs its attention. We use BertViz [Vig, 2019] to present the attention visualization, where darker connections express a stronger focus on the relevant words. In the first sentence of the sample, we observe that in

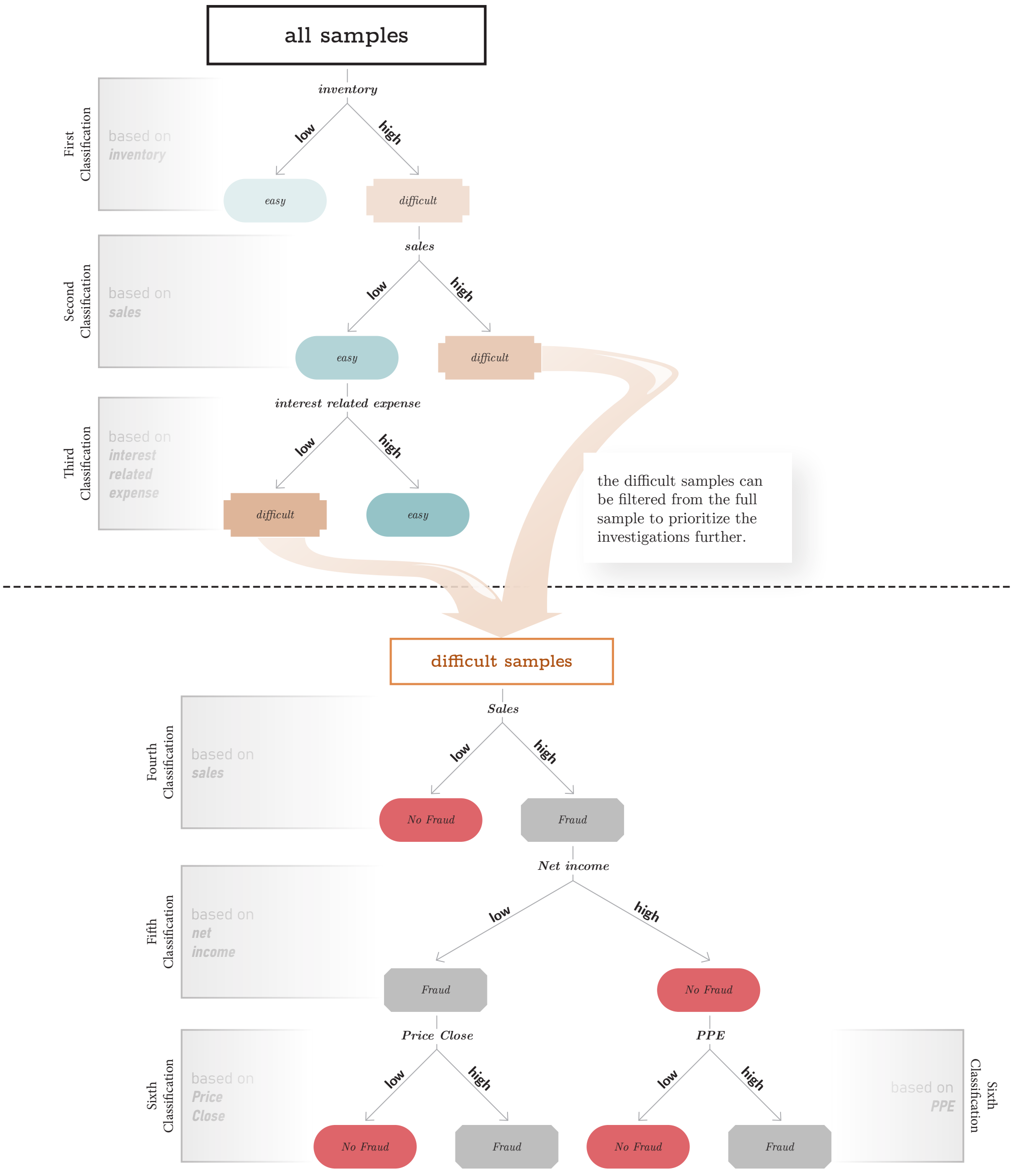
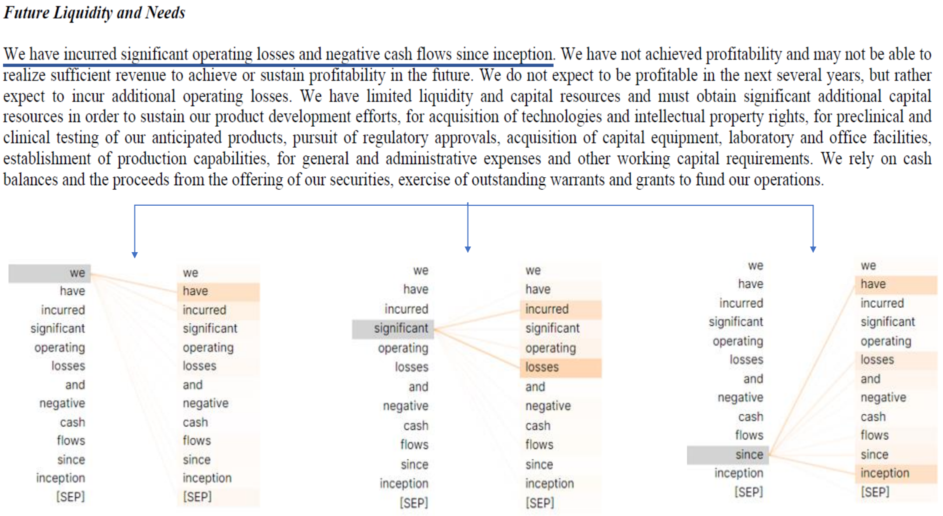


Figure 3.11: Two analyses that are providing insights from financial data. First, we analyze which firms are difficult to identify for our BERT model. Next, we examine what drives fraudulent firms to be erroneously identified by the BERT model as non-fraudulent firms and vice-versa by a similar analysis on the set of difficult firms.

the case of the word “we” the model focuses attention mostly to the word “have”, the word “significant” attends to words “incurred” and “losses”, and “since” attends to “inception”. Besides providing insight into specific patterns of attention, we also observe that the model is able to capture linguistic notions, such as adjectives attending to corresponding nouns, and prepositions attending to their objects.



Future Liquidity and Needs

We have incurred significant operating losses and negative cash flows since inception. We have not achieved profitability and may not be able to realize sufficient revenue to achieve or sustain profitability in the future. We do not expect to be profitable in the next several years, but rather expect to incur additional operating losses. We have limited liquidity and capital resources and must obtain significant additional capital resources in order to sustain our product development efforts, for acquisition of technologies and intellectual property rights, for preclinical and clinical testing of our anticipated products, pursuit of regulatory approvals, acquisition of capital equipment, laboratory and office facilities, establishment of production capabilities, for general and administrative expenses and other working capital requirements. We rely on cash balances and the proceeds from the offering of our securities, exercise of outstanding warrants and grants to fund our operations.

we have incurred significant operating losses and negative cash flows since inception [SEP]
we have incurred significant operating losses and negative cash flows since inception [SEP]
we have incurred significant operating losses and negative cash flows since inception [SEP]
we have incurred significant operating losses and negative cash flows since inception [SEP]
we have incurred significant operating losses and negative cash flows since inception [SEP]
we have incurred significant operating losses and negative cash flows since inception [SEP]

Figure 3.12: Attention example.

3.7 Supplementary Analysis

To further increase the confidence in our findings, we conduct the following supplementary analysis.[8]

3.7.1 Tackling Class Imbalance

The accounting fraud detection problem suffers from class imbalance since the average percentage of fraudulent firms in our text data is only 1.30%. Therefore, we consider whether accounting for class imbalance would help in improving the baseline $BERT_{final}$ model. We attempt to tackle class imbalance by specifying class weights in our fine-tuning procedure using validation set, which would cause the model to pay more attention to the fraud samples. We observe that assigning class weights by obtaining them from the training set does not improve predictions on the validation set. We use different combinations of batch size and number of epochs from the set {1,2,...,8} and specify class weights of fraud and no-fraud samples in the validation set. We find that the $BERT_{final}$ model obtains the highest validation AUC of 0.856 with 4 epochs and batch size 4 after specifying class weights. This score is higher without accounting for class imbalance (0.875), as described in Section 3.5.1.

3.7.2 Robustness Check

We perform an additional robustness check following research conducted by Siano and Wysocki [2021]. For every test year, we first identify the 30 most frequent words in the corresponding training set and replace these words from the test set with a random word

[8] Detailed results available upon request.

$wxyz$ in our text data. We find that replacing these words does not affect the model extensively as it produces an average AUC of 0.824, is able to produce an average NDCG@k score of 0.637, and captures 94 fraud samples altogether in the 14 test years. This is lower than the values obtained with $BERT_{final}$ model, where AUC is 0.826, NDCG@k score is 0.675, and it captures 107 fraud samples. It shows that our model is not extensively dependent on the most frequent words and even after replacing them with a random word, it can retain its contextual learning from the texts.

3.7.3 Excluding Serial Frauds

In some firms serial fraudulent behavior is detected, where fraud spans over multiple consecutive years. On the other hand, it is also likely that the same professional body is responsible for filling the 10-K reports. Therefore, it is important to investigate whether our model is gaining knowledge about fraudulent behavior, or merely learning the writing pattern from serial fraud firms. For each test year, we first identify fraudulent firms in the training set which are appearing multiple times. Next, we keep only one fraudulent observation of such a serial fraud firm and remove other observations. Specifically, we keep the first fraudulent year's observation and remove the consecutive fraudulent observations.[9]

This results in a drop of 51% of total fraud observations. However, even after such a significant drop of total fraudulent observations in the training set, retraining the $BERT_{final}$ upon excluding serial fraud did not reduce its performance significantly. The retrained model finally produces an average AUC of 0.763 (compared to 0.826 for $BERT_{final}$), captures 72 fraud samples altogether in the 14 test years (compared to 107 for $BERT_{final}$), and an average NDCG@k of 0.508 (comparing to 0.675 for $BERT_{final}$). The drop in performance can be accounted for by the considerable drop in fraudulent observations in the training data. This analysis shows that the model is in general learning about the inherent nature of fraud and not only picking up the style and language of serial fraud firms, which also demonstrates the possibility of generalizing the approach to a broader population.

3.7.4 Ensembling All Models

We investigate whether ensembling $BERT_{final}$, LDA, and the RUSBoost model would result in further improvement. For this experiment, we first compute weighted average of predictions from these 3 models as $Ensemble_{all} = w_1 * rank(pred_{final}) + w_2 * rank(pred_{LDA}) + (1 - w_1 - w_2) * rank(pred_{RUSBoost})$, where $w_1, w_2 \in (0, 1)$. We search the optimum values

[9] For example, for a specific firm and for test year t, if fraud is detected in three years in the training period ($t-4$, $t-3$, and $t-2$). We then keep only the first fraudulent observation (in $t-4$), and exclude the following fraudulent observations (in years $t-3$ and $t-2$).

of w_1 and w_2 using validation set from the search space of {0.1,0.2,...,0.9}, and find that $(w_1, w_2) = (0.3, 0.4)$ maximizes the validation AUC on the year 1999. With these values of w_1 and w_2, we compute the $Ensemble_{all}$ for all the 14 years using ensemble data. We find that $Ensemble_{all}$ produces an average AUC of 0.836, an average NDCG@K score of 0.502, and it captures altogether 69 fraudulent firms in the 14 test years. Although we show that ensembling all three models with a simple weighted average does not seem to improve the performance of the model, future research could use more sophisticated methods such as bagging and boosting that could potentially further improve performance.

3.8 Discussion and Future Research

Because of the progress in the natural language processing models in the past years, the simple financial information extracted from the balance sheets is no longer enough to supplement the textual models. This is also visible from our results, presented in Table 3.4, where the performance of the final BERT model is similar to the performance of the Ensemble model, and even outperforms it in terms of economic significance. Instead of using raw financial information, more complex financial measures could be used to improve the prediction.

In the overview tables of our sample, namely Tables 3.1 and 3.2, it is visible that the percentage of detected fraudulent firms is declining over time. This potentially indicates that an increasing number of fraudulent companies are being undetected. Incidentally, SEC indicated that their focus shifted during the financial crisis in 2008, focusing more on the collateralized debt obligation (CDOs), residential mortgage-backed securities (RMBS), and Ponzi schemes [Ceresney, 2013]. The change of focus could also explain the declining number of detected fraudulent firms in our sample. Moreover, some studies suggest that the number of fraudulent firms is significantly higher than what is actually detected, some citing as much as 11% of the large U.S. public corporations allegedly committing fraud [Dyck et al., 2013]. Working with regulators and incorporating state-of-the-art tools from natural language processing could help detect more fraudulent companies, beyond the currently detected ones.

We believe that the potential of contextual language learning in detecting accounting frauds is vast and that a lot of facets are still left unexplored. In the following, we provide some ideas for future research. For example, future research can be carried out to explore if extracting a deeper sense of the business text (from financial reports, conference calls, or corporate social responsibility reports) can help in forecasting companies' earnings or in predicting audit quality. Another interesting direction for future research could be contextual learning based on academic and professional publications, such as earnings announcements, earnings call transcripts, analysts' reports, and journals. Similar to SciBERT [Beltagy et al., 2019], BioBERT [Lee et al., 2020], researchers can develop an accounting BERT model by pre-training on publications from accounting literature and further directing it in order to tackle domain-specific tasks. Finally, the research has demonstrated that BERT performs well in different languages. It would be interesting to explore how the contextual complexity and topics in accounting reports vary in different languages or in different geographical locations.

3.9 Conclusion

The problem of accounting fraud detection sparks interest among auditors, investors, and researchers. However, solving this problem is not easy, and detecting fraud is neither easy nor free. Previous literature mostly explored the potential of using different quantitative features (such as information from financial statements or the stock market) to detect the likelihood of fraud, and recent literature started investigating the use of textual analysis to detect fraud. We build on this research and show how including context from financial reports helps in detecting accounting fraud. We apply the BERT model to the accounting field to learn the contexts of the MD&A section of annual 10-K reports and further direct that contextual knowledge to detect accounting fraud. We find that the BERT model significantly outperforms previously used textual and quantitative models. Moreover, we find that our final model identifies five times more fraudulent firms than the textual benchmark by investigating the same number of firms, and three times more than the quantitative benchmark.

4 What Makes Earnings Predictable?

4.1 Introduction

In this chapter, we theorize and empirically identify what makes earnings predictable. Forecasting firms' future earnings is essential for maintaining an efficient capital market and accurate earnings forecasts are essential for making informed investment decisions [Clement and Tse, 2003, Hope, 2003, Barron et al., 2009]. Hence an effective and accurate earnings forecasting model has been of great interest to investors, shareholders, and corporate regulators.

Given its importance, a large body of research has been dedicated to developing a robust forecasting model. However, the majority of earnings forecasting research has relied on a restricted design, such as the linear regression framework (OLS), and most models have displayed very little outperformance compared a simple random walk model [Hou et al., 2012, So, 2013, Li and Mohanram, 2014]. One potential reason for this result may be due to the limitations of OLS regressions, which make earnings forecasting models in the extant literature over-restrictive. Moreover, the forecasting models suffer from different biases generated from design choices, such as bias generated from model selection and training data [Hyndman and Athanasopoulos, 2018, Petropoulos et al., 2020]. Because of such restrictions, observations at the time (t+1) may become no longer predictable by a model using observations from $\{t, t-1, t-2, ...\}$. Statistical restrictions and research design biases impair the development of a comprehensive forecasting model that is robust.

To obtain better forecasting performance from any model, recent advances in machine learning have been adopted to reduce the biases that are generated from design choices, producing models that outperform traditional OLS models [Binz et al., 2020, Cao and You, 2020, Easton et al., 2020b, Hendriock, 2021]. As Monahan [2018, p. 166] argues, statistical learning allows the researcher to "let the data speak", without the need for an explicit theory regarding for instance the functional form and the selection of explanatory variables. At the same time, the lack of theoretical foundation may also be detrimental to forecasting, as there is a substantial risk of overfitting the data. In addition, theory informs the research design toward optimization in a machine learning environment.

Our theoretical framework is based on the premise that accounting data is multidimensional, consisting of hard information, that is subject to verification of consistency with mandatory generally accepted accounting principles, and soft information, which is the unaudited and unregulated disclosure of information through for instance press releases, conference calls and the management discussion and analysis (MD&A) [Bertomeu and Marinovic, 2016, Bertomeu et al., 2019, 2021, Versano, 2021]. Both hard information and soft information are subject to managerial discretion with regard to the production of this information. The production of any earnings forecast is determined by the probability density function of information at time *t*, subject to the interaction of hard and soft information at time *t*, and the probability density function of information at time *t+n*, subject to the interaction of hard and soft information at time *t+n*. In our setting, earnings are considered more predictable when the distribution of forecast earnings *t+1* resembles the distribution of current earnings *t* more.

Next, we explore our theoretical framework empirically by examining whether combining multiple and diverse machine learning models makes earnings more predictable. We argue that strong generalization in an unobserved framework is possible if we diversify our design choices and combine them into a single framework [Wolpert, 1992, Rogova, 1994, Sharkey, 1996]. Finally, we apply our theory on particular sets of firms that are hard-to-predict, more specifically, loss firms and non-surviving firms.

We introduce a hybrid machine learning framework that is *Stacking* [LeDell, 2015, Michailidis, 2017] that can accommodate multiple and typically diverse machine learning models into a single structure to forecast earnings. Within the stacking framework, we incorporate three diverse models. First, we use a linear first-order Auto-Regressive (AR) model that is a generalized random-walk model which is the most comprehensively used model in earnings forecasting literature. Second, we use the non-linear LightGBM model [Ke et al., 2017] that uses Compustat features. Third, we use a state-of-the-art textual deep learning model RoBERTa [Liu et al., 2019] that uses forward-looking MD&A texts from the annual 10-K reports collected from the Securities and Exchange Commission's (SEC) EDGAR database. We investigate if the stacking framework with these three diverse models helps in earnings forecasting.

Additionally, unlike other domains of forecasting (weather, population, etc.) it is still unclear if earnings forecasting errors are unbiased with scale, or biased with scale due to managerial discretion [Cheong and Thomas, 2011, 2018]. We exploit this empirical observation to determine if a certain model's performance deterioration is attributable to unforeseen natural volatility or to intrinsic flaws in the model design or due to managerial discretion. Since commonly used metrics (such as Mean Absolute Error MAE) to measure forecast accuracy are scale-dependent, we address this by introducing a new scale-independent metric. Following [Hyndman and Koehler, 2006], we propose a scale-independent metric Scale-independent Absolute Forecast Error (SAFE) to determine model performance. Being scale-independent, this metric can be used to compare the performance (forecasting accuracy) across models or across different time frames.

We find that our final model based on stacking outperforms the AR model by 8.12% on average. Moreover, we find that while our proposed model is 8.24% better than the AR model in the first ten years (1994-2003) of the test data, it becomes stronger with time and outperforms the AR model by 11.78% in the final ten years (2010-2019) of the test set.

The restrictions of traditional statistical forecasting are particularly challenging in a volatile environment. For instance, loss firms have proven to cause difficulties in forecasting earnings [Hwang et al., 1996, Brown, 2001, Kothari et al., 2009]. While Li [2011] has improved forecasting models by incorporating losses in a non-linear OLS model, improvements in forecasting using traditional models have been minimal.

We, therefore, test if our empirical approach also makes future earnings predictable in a particular set of firms that are considered "*hard to predict*". We find that the average performance of our final model is 12.11% better for than the AR model loss-making firms, and 7.62% for non-surviving firms.

Our study joins a growing body of earnings forecasting literature. While the majority of extant earnings forecasting models relied on simple linear regression, we introduce a novel method to forecast earnings by combining three diverse machine learning models. Our study further contributes to introducing a new scale-independent metric to measure forecasting errors in the literature so that we can have a fair comparison of models' performance across time. We find that our final model not only significantly outperforms the benchmark model, it also improves the forecasting in a "*hard to predict*" sets of firms previously identified in the literature. Finally, our study is the first study that uses a textual model to forecast earnings.

4.2 Background and Prior literature

Over the last six decades, forecasting future earnings has been of great interest to researchers, investors, and analysts. While analysts' forecasting has been studied extensively (see the comprehensive reviews in Ramnath et al. [2008], Givoly and Biddle [2018]), we review studies that are based on statistical methods developed on publicly available data. We do not include the analysts' forecasting literature in our review because of two reasons. First, analysts' forecasting is applicable only for a specific set of firms. Second, the forecasting methods are not publicly available for investors and the researchers to replicate. The set of studies that focus on statistical approaches to forecast future earnings can be divided into three sets.

The first set of studies is based on a simple random walk model, where we assume the expected value of earnings at time t + 1 is only dependent on earnings at time t, More specifically, $E(E_{t+1}) = \alpha + E_t$. Researchers [Ball and Watts, 1972, Albrecht et al., 1977, Watts and Leftwich, 1977] have developed the foundation of earnings forecasting models based on this fundamental random walk property. Few studies also established that simple random walk models are also better than advanced ARIMA models [Brown, 1993, Kothari, 2001]. However, simple random walk models are too restrictive and it does not exploit other predictors in forecasting.

The second set of studies is based on simple linear time series regression using cross-sectional data to explore the predictive potential of other historical accounting variables. Hou et al. [2012] investigated whether current assets, dividends, and accruals can help forecast earnings. So [2013] studied the predictive potential of book-to-market ratio and stock price for forecasting earnings per share. Li and Mohanram [2014] used book value of equity and total accruals as predictors to develop two separate models for loss firms and profitable firms. This set of models allows for selection bias in their design choice as they assume that the future earnings are linearly dependent on the time series property of current earnings and other specific historical accounting predictors.

The third set of studies includes the recent machine learning models that explore the non-linear relationship between future earnings and other historical accounting numbers. Easton et al. [2020b] used k-nearest neighbor regression for forecasting earnings. Cao and You [2020] and Binz et al. [2020] compared several machine learning algorithms for earnings forecasting using historical numbers. Hendriock [2021] used probability density functions to forecast earnings.

4.3 Theoretical Framework

Despite the recent developments in the literature, recent studies still find that a simple random-walk-based model is more effective than other models [Monahan, 2018, Easton et al., 2020a]. We theorize three potential issues that explain why the existing extant models are underperforming.

4.3.1 Problem 1: Selection Bias and Design Bias

While extant literature helps us understand the useful determinants of statistical earnings forecasting models, all these approaches potentially suffer from selection bias due to design choice. Either the model assumes too restrictive assumptions such as linear dependency, or it uses only a set of historical accounting numbers. Moreover, the historical accounting numbers are backward-looking, which also restricts the forecasting ability.

When we aim to estimate the expected future earnings for year $t + 1$, using a model f

and set of predictors X_t which are observed till the year t, that is $E(E_{t+1}) = f(X_t)$, we assume that the distribution of X_t would be perpetual over time t. Such strong assumption leads to selection bias. For example, studies find how distributions of accruals relative to cash flows [Bushman et al., 2016, Green et al., 2021], revenues relative to expenses [Dichev and Tang, 2008, Srivastava, 2014] or accounting losses [Givoly and Hayn, 2000, Klein and Marquardt, 2006] change over time. Additionally, such historical accounting variables are not forward-looking. Hence forecasting models that use these variables as predictors would be inefficient to forecast future earnings and the models become ineffective in subsets where distributions of predictors alter. In other words, assume $E(E^*_{t+1}) = f^*(X_t)$ after changes in the distribution of accounting data. A linear forecasting models is then biased as $f(X_t) \neq f^*(X_t)$.

Moreover, there is no general framework to develop a robust forecasting model. Rather, the majority of earnings forecasting research has relied on a rather general method that is simple linear regression. Due to such a restricted general approach, earnings forecasting models in the extant literature are over-restrictive and therefore suffer from design bias. Such models almost always have limitations arising out of sets of firms which are "*hard to predict*". For instance, loss firms and firms with poor performance cause difficulties in forecasting earnings [e.g. Li, 2011].

4.3.2 Problem 2: The interaction between hard and soft accounting data

Accounting data is multidimensional. On the one hand, there is hard accounting data, such as reported earnings, of which the production is governed by regulation (e.g. US GAAP or IFRS), and which is subject to mandatory audit. On the other hand, firms also produce soft information. This data is often voluntary (e.g. NON-GAAP information and management forecasts), unregulated (e.g. conference calls), or unaudited (e.g.the management discussion and analysis (MD&A)).

Hard accounting data is limited in its ability to provide information on future earnings due to accounting principles such as the revenue recognition principle and the matching principle, as well as the conservatism principle. To the extent that managers are limited in their ability to provide forward-looking information in hard data, they may provide forward-looking information in soft data, either as a supplement or as a substitute [Bertomeu et al., 2021]. For instance, managers may provide NON-GAAP information to help investors predict future cash flows for loss firms [e.g. Leung and Veenman, 2018]. However, both hard information and soft information are subject to managerial discretion [e.g. Versano, 2021]. Discretion is necessary to optimize the quality of the information signals

[Bertomeu et al., 2019]. This discretion may be used informatively or opportunistically. For instance, Bertomeu and Marinovic [2016] suggests that misreporting is more likely when soft information is issued jointly with hard information. Linear prediction models are biased to the extent that they are not able to model the interaction between hard information and soft information. In other words, the dynamic interaction between hard and soft accounting information may cause $f(X_t) \neq f^*(X_t)$.

4.3.3 Problem 3: Scale-dependent empirical evaluation

Extant models in forecasting literature relied on evaluation metrics such as Mean absolute error (MAE) or Root Mean Squared Error (RMSE), which are scale-dependent. However, previous research identified that it is not obvious whether earnings forecasting errors are biased with scale [Cheong and Thomas, 2011, 2018]. That makes it difficult for models to be compared with time. We illustrate this by the following two examples, where scale factors are obvious.

Example A: Let's say that the daily temperatures of city A and city B at a certain point of time are T and $5T$. Then, a temperature forecasting model would predict the temperatures between $(T-\epsilon, T+\epsilon)$ and $(5T-\epsilon, 5T+\epsilon)$ for city A and city B respectively. [10] Example B: Let's say that the population for city A and city B are P and $5P$ respectively. Then, a population forecasting model would predict the populations between $(P-\epsilon, P+\epsilon)$ and $(5P-5\epsilon, 5P+5\epsilon)$ for city A and city B respectively. So, we observe how forecasting error is scale-independent in example A and scale-dependent in example B. While the behavior of forecasting error concerning scale is obvious in some other domains, earnings forecasting is an exception. Therefore, if earnings at year t and $t+1$ are respectively E and $5E$, we can not determine if a good forecasting model should produce forecasts between $(5E-\epsilon, 5E+\epsilon)$ or $(5E-5\epsilon, 5E+5\epsilon)$ for the year $t+1$.

4.4 Method

One theoretical approach to reduce the bias generated from design choice is to create multiple subsamples (F_i) of the entire data (X) such that $X = \bigcup F_i$. Next, employ the characteristics of each domain (F_i) as it pertains to forecasting optimization to forecast earnings. However, this is difficult to implement as finding such optimum and diverse mutually exclusive and exhaustive F_i is impractical. Moreover, it is also important to optimize the interaction between soft and hard information for each F_i.

[10] This example based on differences in temperature as it relates to scale is taken from Cheong and Thomas [2011]

In this study, we put forward a machine learning framework that prunes the partialities originating from model design choice. Our framework effectively combines three diverse machine learning models and explores beyond the hard information that is historical accounting numbers. We also let our framework learn from forward-looking soft information that is MD&A texts from the annual 10-K reports through a textual model. With each model, we split the data randomly and expect the model to perform well in one subset and let the other models generalize the bias created by the previous model in other subsets. Finally, we investigate if our proposed model outperforms a generalized random-walk model that is AR(1).

In this section, we describe how we design the stacking framework combining three diverse models to forecast future earnings. We consider the following time series data $\{(X_1, E_1), (X_2, E_2), \cdots, (X_t, E_t), (X_{t+1}, E_{t+1})\}$, where E_i's are the future earnings at year i and X_i's are the predictors observed till year i. We use training set $X_{train} = \{(X_1, E_1), (X_2, E_2), \cdots, (X_t, E_t)\}$ and test set $X_{test} = (X_{t+1}, E_{t+1})$. Our final objective is to train a forecasting model M based on X_{train} and predict future earnings at year $t+1$, $\hat{E}_{t+1} = M(X_{t+1})$.

Our final model M is based on three diverse models $\{M_1, M_2, M_3\}$. For each M_j, we split X_{train} into 5 random disjoint folds: $\{F_1, F_2, \cdots, F_5\}$. We train M_j five times (leaving one fold out) on $X_{train} \backslash Fi$ and predict on F_i and X_{t+1} to obtain the out of fold predictions $oof_{i,j}$ and predictions $\hat{E}_{t+1,i,j}$. Our final out of fold predictions from model M_j corresponding to X_{train} is defined as $oof_j = \bigcup_{i=1}^{5} \{oof_{i,j}\}$. Final test prediction from model M_j is defined as $\hat{E}_{t+1,j} = \sum_{i=1}^{5} \hat{E}_{t+1,i,j}/5$.

Our final model M combining $\{M_1, M_2, M_3\}$ is developed by training a linear regression model. We train a linear regression model using oof_j as our independent variable and y_{t+1} as our dependent variable. Finally, our stacking prediction is based on the linear regression model applied on $\hat{E}_{t+1,j}$s. Hence the final prediction is defined as $\hat{E}_{t+1} = \alpha_0 + \sum_{j=1}^{3} \hat{E}_{t+1,j}$. Next, we discuss the three diverse models that we combine in our stacking framework.

4.4.1 AR Model

The first order auto regressive model or the AR(1) model assumes that the earnings at time $t+1$ can be linearly explained by the earnings at time t. Hence, $\hat{E}_{t+1} = \alpha + \beta E_t$

4.4.2 LightGBM Model

LightGBM is a gradient boosting decision tree algorithm developed by Ke et al. [2017]. In the gradient boosting decision tree algorithm, a sequence of small decision trees (weak regressors) are developed iteratively. The decision tree at every iteration tries to minimize the error produced by the decision trees in the previous iterations. Finally, a strong regressor is developed by computing the weighted average of these weak decision tree regressors. Unlike other gradient boosting decision tree algorithms, LightGBM grows decision trees leaf-wise instead of depth-wise. Table 4.1 presents the main parameters of the LightGBM regressor. We have used *LGBMRegressor* from Python's *LightGBM* library.

LightGBM Parameters	Description
num_leaves	Maximum number of leaves in a tree in each iteration
learning_rate	Step parameter that manages the speed of model training
max_depth	Maximum depth of tree in each iteration (<0 means no limit)
colsample_bytree	Fraction of features to be used in each iteration
subsample	Fraction of observations to be used in each iteration
n_estimators	Number of trees/iterations

Table 4.1: Main parameters of LightGBM and their descriptions

4.4.3 RoBERTa Model

Robustly Optimised BERT Pre-training Approach (RoBERTa) is a deep neural network-based semi-supervised model that improves on the Bidirectional Encoder Representations from Transformers (BERT) model [Devlin et al., 2018]. Similar to BERT, RoBERTa is also trained to learn the deeper sense of language contexts present in text data. Application of BERT is processed through transfer learning. [11] This essentially comprises of two primary steps: (1) pre-training and (2) fine-tuning. In our study, we use a pre-trained BERT model (RoBERTa base, uncased[12]) developed by Liu et al. [2019]. Following [Sun et al., 2019], We further pre-train and fine-tune it on MD&A texts in order to predict the future earnings.

4.5 Performance Evaluation

To discuss how we evaluate the models' performance, we first discuss the opted validation strategy, and then we introduce a new evaluation metric.

4.5.1 Validation Strategy

We use rolling windows of consecutive five years to train our models and the immediate next year to test the performance of our model. We use 1994-1999 as our validation set to obtain the optimized set of models' parameters. We train our final models on every 5 years of data to forecast earnings for the next year. We use 1999-2003 as our first training set and 2004

[11] Transfer Learning: Transfer learning is a method of training a deep neural network on a large dataset and use it to accomplish related work by transferring the knowledge gained in the former training, see [Bozinovski, 2020] for more details on Transfer Learning.

[12] uncased model does not differentiate between lower case and upper case characters.

as our first test set, 2000-2004 as our second training set and 2005 as our second test set, and so on. This procedure produces 16 test years ranging from 2004-2019 [Figure 4.1].

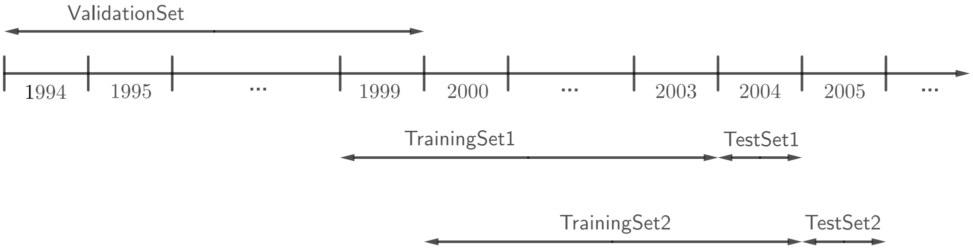


Figure 4.1: Validation strategy of our model

4.5.2 Evaluation Metric

We introduce a new performance evaluation metric for forecasting earnings, referred to as Scale-independent Absolute Forecast Error (SAFE), based on Mean Absolute Scaled Error (MASE) from Hyndman and Koehler [2006]. Commonly used metrics in earnings forecasting literature such as MAE, RMSE, etc. may lead to spurious results in the context of empirical comparisons since it is still unclear if firms' earnings are unbiased with scale Cheong and Thomas [2011, 2018].

While absolute measures do not remove the scale of the data, SAFE is based on relative error and focused on removing the scale factor of the observations by comparing the predictions obtained from a naive forecasting method. To evaluate the performance of the forecasting model at year $t+1$, we use the mean value of firms' earnings ($\bar{E}_t$) from the previous year (t) as the naive forecast of earning estimate of the current year ($t+1$). We scale the Mean absolute error (MAE) of E_{t+1} and $\hat{E}_{t+1}$ by MAE of E_{t+1} and $\bar{E}_t$ to obtain the SAFE scores.

$$SAFE = \frac{MAE(E_{t+1}, \hat{E}_{t+1})}{MAE(E_{t+1}, \bar{E}_t)} = \frac{\sum_{j=1}^{N_{t+1}} |e_{t+1,j} - \hat{e}_{t+1,j}|}{\sum_{j=1}^{N_{t+1}} |e_{t+1,j} - \bar{E}_t|}$$

where N_t and $e_{t,j}$ and denotes the total number of firms and earnings of firm j in the year t respectively. From the definition itself, SAFE is a scale-independent metric. If SAFE is more than 1, that would indicate that the predictions for the current year ($t+1$) are even under-performing than the naive estimate using average earnings from the previous year (t). Hence, a good forecasting model must produce an SAFE value of less than 1. The lower the value of SAFE, the better the performance of the model.

4.6 Data and Sampling Design

Our study is based on the earnings of publicly traded U.S. firms and our data ranges from 1994 to 2019. We use 1994 as the starting year as 10-K reports are available from that year on the SEC website. We use 1994-2003 to fine-tune models' parameters and 2004-2019 (16 years) as our test set. We adopt AR(1) model as our benchmark model because AR(1) is a generalized random walk model and extant literature strongly argues that random walk is still the most effective earnings forecasting model so far [Monahan, 2018, Easton et al., 2020a].

To remove the data selection bias, we consider two different sources of data to develop our final modeling architecture so that we capture the interaction between hard and soft information. First, we use numerical features based on accounting numbers and second we extract raw text from annual 10-K reports. We merge the numerical data set and the text data set to obtain our final data set that contains altogether 103,006 firm-year observations. Table 4.2 shows the year-wise distribution of the number of firm-year observations. In the next two sections, we discuss the details of preparing the numerical data set and the text data set.

Year	No. of firms	Year	No. of firms
1994	98	2007	1095
1995	133	2008	1104
1996	201	2009	1238
1997	324	2010	1281
1998	361	2011	1329
1999	404	2012	1378
2000	432	2013	1471
2001	512	2014	1913
2002	587	2015	1939
2003	856	2016	1933
2004	922	2017	1979
2005	971	2018	1928
2006	1036	2019	1897

Table 4.2: Distribution of the year-wise number of firms

4.6.1 Text Data Construction

We extract texts from Item 7, which is the Management Discussion and Analysis (MD&A) section of annual 10-K reports from the Securities and Exchange Commission's (SEC) EDGAR database. Since the MD&A section contains forward-looking information Muslu et al. [2015] and also investors use the MD&A section for strategic investments Bryan [1997], Durnev and Mangen [2020], we use texts from the MD&A section to develop our textual model in the forecasting framework.

We use Python to extract potential MD&A sections from 10-K reports by following Berns et al. [2021a]. We collect the list of all CIKs (central index key: unique for each publicly traded U.S. firm) from the SEC's website. Thereafter, for each unique CIK, we collect year-wise annual 10-K report filing dates from 1994 to 2019 and the corresponding accession numbers (accession numbers are unique for each 10-K report). For each 10-K filings, we create the URL using CIK and the accession number that lands to the corresponding 10-K reports. Following Berns et al. [2021a], the text parsing algorithm searches "Item 7. Management Discussion and Analysis" and any one of the phrases "the following discussion", "this discussion and analysis", "should be read in conjunction", "should be read together with", "the following management's discussion and analysis" in the following 5 sentences to identify the beginning of the MD&A section of 10-K reports. The end of the MD&A section is determined by searching the variations of "Item8. Consolidated Financial Statements".

Variables	Sources
Accounts Payable - Trade	Chen et al. [2015], Cao and You [2020]
Accruals	HVZ 2012 Cao and You [2020]
Negative Accruals per share, and zero otherwise	So [2013], Cao and You [2020]
Positive Accruals per share, and zero otherwise	So [2013], Cao and You [2020]
Advertising Expense	Chen et al. [2015], Cao and You [2020]
Assets - Total	Hou et al. [2012], Chen et al. [2015], Cao and You [2020]
Book-to-market ratio	So [2013], Cao and You [2020]
Cash and Short-Term Investments	Chen et al. [2015], Cao and You [2020]
Cash flow from operating activities	Chen et al. [2015], Cao and You [2020]
Common Shares Outstanding	
Common/Ordinary Equity - Total	Li and Mohanram [2014], Chen et al. [2015], Cao and You [2020]
Cost of Goods Sold	Chen et al. [2015], Cao and You [2020]
Current Assets - Total	Chen et al. [2015], Cao and You [2020]
Current Liabilities - Total	Chen et al. [2015], Cao and You [2020]
Debt in Current Liabilities - Total	Chen et al. [2015], Cao and You [2020]
Depreciation and Amortization	Chen et al. [2015], Cao and You [2020]
Dividends Common/Ordinary	Hou et al. [2012], Chen et al. [2015], Cao and You [2020]
Dividends per Share - Ex-Date - Fiscal	So [2013], Cao and You [2020]
Dummy variable indiacating divident payers	Hou et al. [2012], Cao and You [2020]
Dummy variable indicating negative earnings	Hou et al. [2012], So [2013], Li and Mohanram [2014], Cao and You [2020]
Dummy variable indicating zero dividend per share	So [2013], Cao and You [2020]
Earnings at year t	Hou et al. [2012], Li and Mohanram [2014], Chen et al. [2015], Cao and You [2020]
Positive Earnings per share, and zero otherwise	So [2013], Cao and You [2020]
Extraordinary Items and Discontinued Operations	Chen et al. [2015], Cao and You [2020]
Income Before Extraordinary Items	
Income Taxes - Total	Chen et al. [2015], Cao and You [2020]
Income Taxes Payable	Chen et al. [2015], Cao and You [2020]
Intangible Assets - Total	Chen et al. [2015], Cao and You [2020]
Interest and Related Expense - Total	Chen et al. [2015], Cao and You [2020]
Inventories - Total	Chen et al. [2015], Cao and You [2020]
Investment and Advances Other	Chen et al. [2015], Cao and You [2020]
Liabilities - Total	Chen et al. [2015], Cao and You [2020]
Long-Term Debt - Total	Chen et al. [2015], Cao and You [2020]
Nonoperating Income (Expense) Other	Chen et al. [2015], Cao and You [2020]
percentage change in total assets	Chen et al. [2015], Cao and You [2020]
Close - Annual - Fiscal	So [2013], Cao and You [2020]
Property, Plant and Equipment - Total (Net)	Chen et al. [2015], Cao and You [2020]
Receivables Total	Chen et al. [2015], Cao and You [2020]
Research and Development Expense	Chen et al. [2015], Cao and You [2020]
Sales/Turnover (Net)	Chen et al. [2015], Cao and You [2020]
Selling, General and Administrative Expense	Chen et al. [2015], Cao and You [2020]
Special Items	
Total accruals defined in Richardson et al. (2005)	Li and Mohanram [2014], Cao and You [2020]

Table 4.3: Features used for LightGBM model and their sources in extant literature

4.6.2 Compustat Data Construction

Our numerical dataset is obtained from the fundamental annual data from the merged database of CRSP and Compustat. We select a comprehensive list of 48 variables from the extant literature [Table 4.3] and their one-year lag variables to prepare the final numerical data. We define earnings as Income Before extraordinary Items less the Special Items).

We use five exclusion criteria to obtain the final numerical data. First, we remove firm-year observations that have missing values in any of the following variables: total assets, sales revenue, income before extraordinary items, and common shares outstanding. Second, we removed firms whose stocks are not ordinary common shares listed on the NYSE, AMEX, or NASDAQ. Third, we remove firms that are in the financial or regulated industry (SIC: 6000-6999 and SIC: 4900-4999). Fourth, we remove firms with annual fiscal prices is more than 1 USD. Finally, we remove firm-year observations whose earnings or future earnings are missing.

We scale all the predictors and the Earnings by common shares outstanding for each year to ensure our evaluation measure SAFE is not dominated by a small amount of firms with outlier earnings values. Next, we compute the lag variables of the selected 48 variables. Finally, we impute the other missing values by the corresponding mean value of the variables.

4.7 Results and Discussion

First, we discuss the performance of three meta-models that is AR(1), LightGBM and RoBERTa individually, next we discuss the performance of stacking.

First, we use the validation year 1999 to train AR(1) model from 1994-1998. $oof_{AR(1)}$ produces an MAE of 0.663 on 1994-1998 and the final prediction on the validation year 1999 obtains a SAFE score of 0.547. Next, we train AR(1) on every consecutive 5 years rolling window keeping 2004 as the first test year as discussed in section 4.5.1. We find that the AR(1) model finally obtains an average SAFE score of 0.530 in the 16 years of test data [Table 4.5]. We also report the performance of the AR(1) model using the traditional evaluation metric MAE [Table 4.6]. We find the average MAE of AR(1) over the test period is 1.045.

Second, we discuss the LightGBM model. We optimize the LightGBM parameters using the validation set as described in section 4.5.1. Our search space included {0.3, 0.5, 0.7} to find the optimum values for colsample_bytree and subsample. We use default values for other parameters, that are: num_leaves = 31, learning_rate = 0.1, max_depth = -1 and we use the optimum n_estimators found from the early stopping from training phase. The final set of parameters that minimizes the validation SAFE score in the year 1999 by $oof_{LightGBM}$ is {colsample_bytree, subsample} = {0.7, 0.7} and it obtains a SAFE of 0.539. Next, we train the LightGBM model with the optimized set of parameters and it obtains an average SAFE of 0.502 in the 16 years of test data [Table 4.5]. We also report the performance of the LightGBM model using the traditional evaluation metric MAE [Table 4.6]. We find the average MAE of LightGBM over the test period is 0.991.

Next, we discuss the third meta-model. RoBERTa. We first pre-train and fine-tune the RoBERTa model on 1994-1998 and then we forecast earnings on 1999. Pre-training RoBERTa on the validation set produces a perplexity score of 3.428 after 5 epochs. $oof_{RoBERTa}$ obtains a SAFE score of 0.628 in the year 1999. To forecast earnings every year, we pre-train the RoBERTa model using the MD&A text from the year 1994 to the last year of the training sample. For example, to predict future earnings for the test year 2014, we pre-train RoBERTa using the MD&A text from the year 1994 to 2013 and so on for 5 epochs. We observe that pre-training RoBERTa improves the perplexity score over time and that indicates that the model improves with more data [Table 4.4]. For each test year, we fine-tune the pre-trained RoBERTa model with optimized parameters. On average RoBERTa obtains an average SAFE score of 0.664 over 16 test years [Table 4.5]. We also report the performance of the RoBERTa model using the traditional evaluation metric MAE [Table 4.6]. We find the average MAE of RoBERTa over the test period is 1.311.

Before combining the models, we compare the performance of the RoBERTa model against two other textual benchmark models using the bag of words method. The first model uses the top thirty most frequently used positive and negative words from Loughran and McDonald [2011]'s dictionary. The second model uses the forward-looking words proposed by Muslu et al. [2015][13]. For both models, we count the term frequency of words in the MD&As and use them as features of the regression model to predict future earnings. The Loughran & McDonald model produces an average MAE of 1.782 and a SAFE score of 0.899. The Forward-Looking model produces an average MAE of 1.896 and a SAFE score of 0.954. This indicates that RoBERTa significantly outperforms these benchmark models.

Next, we combine all three metamodels with a linear regression stack. We use oof_{AR}, $oof_{LightGBM}$, and $oof_{RoBERTa}$ as a linear regressor to predict future earnings on the training set. Finally, we use the linear regression model on $pred_{AR}$, $pred_{LightGBM}$ and $pred_{RoBERTa}$

[13] Thirty most frequent positive words from Loughran and McDonald [2011]'s dictionary are: ['gain', 'gains', 'able', 'advances', 'improvements', 'best', 'successful', 'opportunities', 'good', 'favorable', 'exclusive', 'achieve', 'profitability', 'successfully', 'success', 'improved', 'opportunity', 'satisfy', 'improve', 'improvement', 'positive', 'strong', 'profitable', 'progress', 'achieved', 'satisfaction', 'enable', 'beneficially', 'better', 'leading'].

The thirty most frequent negative words are: ['loss', 'losses', 'against', 'impairment', 'disclosed', 'deficit', 'termination', 'adversely', 'adverse', 'litigation', 'restated', 'discontinued', 'default', 'concern', 'restructuring', 'failure', 'decline', 'deficiencies', 'weaknesses', 'fraud', 'misleading', 'unable', 'omit', 'defaults', 'bankruptcy', 'damages', 'terminated', 'liquidation', 'omitted', 'negative'].

Forward looking words from Muslu et al. [2015] are: [”will”, ”future”, ”next fiscal”, ”next month”, ”next period”, ”next quarter”, ”next year”, ”incoming”, ”coming fiscal”, ”coming month”, ”coming period”, ”coming quarter”, ”coming year”, ”upcoming fiscal”, ”upcoming month”, ”upcoming period”, ”upcoming quarter”, ”upcoming year”, ”subsequent fiscal”, ”subsequent month”, ”subsequent period”, ”subsequent quarter”, ”subsequent year”, ”following fiscal”, ”following month”, ”following period”, ”following quarter”, ”following year”, ”aim”, ”anticipate”, ”assume”, ”commit”, ”estimate”, ”expect”, ”forecast”, ”foresee”, ”hope”, ”intend”, ”plan”, ”project”, ”seek”, ”target”

Pre-training RoBERTa	Perplexity Score	Pre-training RoBERTa	Perplexity Score
Train: 1994-1998, Test: 1999	2.839	Train: 1994-2011, Test: 2012	2.556
Train: 1994-2003, Test: 2004	2.689	Train: 1994-2012, Test: 2013	2.557
Train: 1994-2004, Test: 2005	2.660	Train: 1994-2013, Test: 2014	2.537
Train: 1994-2005, Test: 2006	2.652	Train: 1994-2014, Test: 2015	2.527
Train: 1994-2006, Test: 2007	2.656	Train: 1994-2015, Test: 2016	2.507
Train: 1994-2007, Test: 2008	2.656	Train: 1994-2016, Test: 2017	2.501
Train: 1994-2008, Test: 2009	2.658	Train: 1994-2017, Test: 2018	2.489
Train: 1994-2009, Test: 2010	2.607	Train: 1994-2018, Test: 2019	2.480
Train: 1994-2010, Test: 2011	2.609		

Table 4.4: Year-wise perplexity Scores from Pre-training RoBERTa for five epochs

to forecast the future earnings on every test set. We find that the stack ensemble model obtains an average SAFE score of 0.495 over 16 test years. This results in an 8.12% improvement over the benchmark AR(1) model. Moreover, we find that the improvement of the Stack ensemble model is more evident with the recent times' data. While the stack ensemble model is 8.24% better than the AR(1) model for the first 10 years of test data (2004-2013), it is 11.78% better for the last 10 years of test data (2010-2019). However, we find that the improvement of the Stack model over AR(1) is not statistically significant in the first 10 years of test data (2004-2013). An analysis of the variance test produces a p-value of 0.118 for the first 10 years of test data. However, we find the improvement of Stack is statistically significant in the last 10 years of test data (2010-2019). Analysis of variance yields a p-value of 0.04 in the last 10 years of test data. This shows that the model learns better with more data and promises to perform better in future tests.

Test Year	AR(1)	LightGBM	RoBERTa	Stack
2004	0.434	0.539	0.761	0.439
2005	0.526	0.523	0.691	0.545
2006	0.578	0.600	0.688	0.586
2007	0.548	0.577	0.705	0.528
2008	0.737	0.633	0.774	0.754
2009	0.612	0.587	0.706	0.577
2010	0.582	0.487	0.654	0.475
2011	0.539	0.457	0.636	0.431
2012	0.492	0.396	0.574	0.379
2013	0.467	0.415	0.597	0.398
2014	0.519	0.483	0.669	0.474
2015	0.451	0.401	0.612	0.399
2016	0.526	0.503	0.640	0.505
2017	0.498	0.447	0.622	0.448
2018	0.367	0.389	0.584	0.362
2019	0.611	0.601	0.705	0.614
Average	0.530	0.502	0.664	0.495

Table 4.5: Master Table showing SAFE performance comparison

Test Year	AR(1)	LightGBM	RoBERTa	Stack
2004	0.646	0.801	1.132	0.653
2005	0.762	0.758	1.000	0.79
2006	0.852	0.885	1.015	0.864
2007	0.912	0.959	1.172	0.879
2008	1.098	0.943	1.153	1.123
2009	0.89	0.54	1.026	0.839
2010	0.959	0.803	1.077	0.782
2011	0.942	0.8	1.113	0.754
2012	0.892	0.717	1.04	0.686
2013	0.898	0.798	1.148	0.766
2014	1.11	1.033	1.43	1.014
2015	0.937	0.834	1.273	0.83
2016	1.24	1.186	1.508	1.192
2017	1.316	1.182	1.644	1.185
2018	1.06	1.125	1.689	1.047
2019	2.211	2.177	2.551	2.223
Average	1.045	0.991	1.311	0.977

Table 4.6: Master Table showing performance comparison using MAE

We present the time trends of our models in Figure A.5 and Figure A.6. We find the superiority of the RoBERTa model among the other textual benchmarks. However, stand alone textual models are not as predictive as the models developed from the financial features. However, upon combining the RoBERTa within the stack framework improves the prediction with time. We also present the pairwise comparison of our models comparing both MAE and SAFE [Table A.3 - A.32].

4.8 Supplementary Analysis

We investigate the difference of predictive potentials of all the models in different deciles of current earnings. We split the earnings into ten deciles for the entire data and find that the models find it difficult to predict the future earnings for firms with earnings at the first and tenth deciles. We find that the stack ensemble model performs best out of all the models even in these two deciles [Figure A.3 and Figure A.4] and that indicates the usefulness of the stack ensemble model for “hard to predict” firms. In these next two sections, we discuss the performance of our stack ensemble model on different subsets of firms that prior literature identifies as “hard to predict” firms.

4.8.1 Surviving and Non-surviving firms

We split our entire sample into two subsets. One is with the set of surviving firms that stayed throughout the entire test sample period of 16 years and two is the set of firms that are not present throughout the entire time frame of our test period. We found altogether 381 surviving firms in our sample. We find that for the set of non-surviving firms, the benchmark AR(1) model produces an average SAFE score of 0.556, and or stack ensemble model produces an average SAFE of 0.513 which is 7.62% [Table A.40]. We find that the out-performance in the stack over AR(1) is also statistically significant in the last 10 years of test data for non-surviving firms. On the other set with surviving firms, AR(1) model produces an average SAFE score of 0.499 and the stack ensemble model obtains an average SAFE score of 0.479 which results in a 4.01% improvement [Table A.39]. However, analysis of variance test indicates that this improvement of the stack over AR(1) is not statistically significant. This shows that the surviving firms are not hard to predict and due to their steady growth, AR(1) is sufficient for their earnings forecasting.

4.8.2 Loss and profit firms

We also test our models' performance with the set of loss firms that is whose earnings are negative in the corresponding fiscal year. We find that with these loss firms, our stack ensemble model outperforms the AR(1) model by 12.11%. While the AR(1) model produces an average SAFE score of 0.836, the stack ensemble model obtains an average SAFE score of 0.735 across 16 test years [Table A.33]. An analysis of the variance test also confirms that the out-performance of the stack model is statistically significant over AR(1) model. We also test our model with the set of profit firms that is the set of firms with positive earnings in the corresponding fiscal year. We find that for-profit firms, the AR(1) model and stack ensemble model obtains average SAFE of 0.533 and 0.502 respectively resulting in a 5.78% better performance [Table A.34]. However, we find that the improvement of the stack model over AR(1) is not statistically significant for profit firms. However, the improvement for profit firms is statistically significant over the last 10 years of test data.

Next, we also explore how our stack model captures the interaction between soft and hard information by evaluating the performance in the following 2X2 settings of loss firms and Non-GAAP firms. We use 4 subsets: 1. loss and Non-GAAP firms, 2. Loss and no Non-GAAP firms, 3. Profit and Non-GAAP firms, 4. profit and no Non-GAAP firms. We find a statistically significant performance improvement of 9.48% and 12.58% of the stack model over AR(1) model in the set 1 and set 2 respectively [table A.35, A.36]. For set 3, the stack model outperforms the AR(1) model by 6.78% and the improvement is also statistically significant in the last 10 years of test data [table A.37]. However, for set 4, we find that although the stack model is out-performing the AR(1) model by 4.10% the improvement is not statistically significant [Table A.38].

4.9 Conclusion

Predicting future earnings has always been of great interest to investors, shareholders, and researchers. However, statistical time-series forecasting suffers from different biases generated from modeling design choices. In this study, we examine how to combine several machine learning models so that the modeling design itself reduces partialities to produce more accurate earnings forecasts.

In this study, we introduce a stacked ensemble modeling framework that combines three machine learning models: AR(1), LightGBM, and RoBERTa. We also introduce a new metric to the literature that can be used to compare forecasting accuracy across models in different time frames. We find that our model significantly outperforms the benchmark model from the extant literature. We also demonstrate our models' superiority on different sets of "hard to predict" firms that the literature previously identified.

5 General Discussion and Future Perspectives

5.1 General Discussion

Financial accounting helps assess the business performance and guides investors to set goals, thereby maintaining a healthy economic state of affairs. Shareholders and other stakeholders rely on financial reports to evaluate organizations' financial health. Financial accounting is the instrument to collate the day-to-day transaction level data to produce these financial reports and beyond.

Because of its large-scale economic dependencies, the existing methods in the financial accounting literature and practice are unsurprisingly conservative in nature. Practitioners such as auditors analyze big accounting data to find patterns to investigate the reliability of financial reports. These methods tend to involve extensive human interventions, which renders the process error-prone and time consuming.

There has been a huge technological shift in the last few decades. Specifically, the machine learning literature introduced state-of-the-art algorithms which shifted the paradigm of data analysis and pattern recognition. Researchers from the financial accounting domain have also started implementing these machine learning algorithms to solve practical problems [Loughran and McDonald, 2016, Bertomeu, 2020, Liu, 2022]. However, because of the generally conservative approach, researchers also argue that machine learning is yet to be fully unleashed to its full potential in the financial accounting domain [Lev and Gu, 2016, Dickey et al., 2019, Bertomeu, 2020].

Moreover, the state-of-the-art algorithms are black boxes in nature i.e. they lack explainability. Therefore, it is difficult to deploy these models at the production level to draw inferences. Additionally, it is also convoluted how to combine the machine learning algorithms optimally with financial accounting to solve real-world problems. Hence, we have to delicately ensemble these two literatures to find working solutions.

In this thesis, I put forward three chapters that lie in the intersection of both machine learning literature and financial accounting literature. I introduce advanced machine learning frameworks and methods that can be implemented in an optimized way to solve real-world problems in the financial accounting domain.

The methods and the frameworks are heavily dependent on current accounting practices as the input features were collected from the extant literature. These methods are also designed and motivated to mimic how practitioners deal with these real-world problems in a more efficient way.

In the next sections, we discuss the contributions of the research following the limitations and the directions to future research.

5.2 Contribution to Research and Practice

In order to assess if the financial reports are free from material misstatements, domain experts investigate the accounting data that is used to prepare the reports. Such accounting data consists of day-to-day transactions and hence they are found to be substantially large in volume. Therefore, it becomes impossible to sift through the data to find anomalous observations. Moreover, a small number of anomalous observations in the accounting data can result in inaccurate financial reports. Auditors generally work with a small sample of the data to investigate if anomalies are present.

In chapter 2, a semi-supervised framework has been proposed which is motivated by such sampling design. Our framework first produces a representative sample of the entire big data and then detects anomalies in that sample using an unsupervised method. Next, the anomalies are thoroughly checked by the auditors and domain experts. Next, a supervised algorithm is deployed to produce the anomalies in the entire sample using the sampled data as the training set. This method ensures that the learning from the representative sample can be transferred to the big data. In our experiment, we use data with 32 million records and the proposed framework could capture 90% and 96% anomalies upon investigating only 5% and 10% of the entire data.

Our study also introduces pseudo-labeling in the accounting practice. Since big accounting data are generally unlabelled, pseudo-labeling can help in producing data that can be used to train supervised models. While pseudo-labeling is also manually possible i.e. by sifting through the data and identifying the anomaly labels, we employ an unsupervised method to obtain these pseudo-labels.

Upon detecting anomalies by the unsupervised method in a small representative subset of the data, the overall framework becomes scalable. Training an unsupervised algorithm on large data can be computationally expensive hence depending on the memory capacity, it is possible to opt for an adequate sample size to implement the unsupervised algorithm. The proposed framework also goes through expert validation after producing the anomalies in the small sample. Moreover, the classification rules can be further examined by experts to understand the nature of the anomalies. Our anomaly detection framework is independent of any accounting assumptions and hence can also be generalized in detecting anomalies in any big data.

Although financial reports are vastly relied upon by investors to make investment decisions, fraudulent firms tend to misreport to hide their deceptive activities. Managers of the fraudulent firms deliberately misguide the investors, which can potentially cause significant disruption to the stock market and the economy. Therefore accounting fraud detection has become an important subject to study in the last few decades.

While extensive research has been already done using financial numbers [Dechow et al., 2011, Perols et al., 2017, Bao et al., 2020], recent studies found clues in the business text from financial reports to detect accounting fraud [Loughran and McDonald, 2016, Bushee et al., 2018, Brown et al., 2020]. However, most of these studies are based on the bag-of-words method which is finding features from discrete words present in the text to detect fraud. In chapter 3, we put forward a textual model that captures the context from the business texts of financial reports to detect accounting fraud.

We apply and fine-tune the BERT model [Devlin et al., 2019] using texts from the annual financial reports of publicly traded U.S. firms to detect accounting frauds. The proposed method in chapter 2 is compared to two benchmark models, which are both textual [Brown et al., 2020] and financial [Bao et al., 2020] based approaches. Our model outperforms these benchmark models by 15% and 12% respectively.

Financial investigators investigate publicly traded U.S. firms to detect fraudulent activities. The investigation process is not only expensive but also takes a significant amount of time. Previous research found that the average time gap between the misreporting and the initial declaration of fraud is around two years. While it is practically impossible for the regulators to investigate all the publicly traded firms, SEC also indicated that the focus on detecting accounting fraud has been diluted because of prioritizing the focus on investigating Residential mortgage-backed securities, Collateralized debt obligations, and Ponzi schemes [Ceresney, 2013].

Our proposed fraud detection model finds five times more fraudulent firms and three times more fraudulent firms than the textual benchmark and the quantitative benchmark respectively upon investigating only 1% of the firms. In other words, we believe that our model can help in identifying the same number of frauds upon investigating less number of firms. Therefore, the adoption of the model can be economically significant for regulators, auditors, and financial investigators.

Investors tend to forecast how a company is going to perform in the future to decide its investment strategies. Forecasting future earnings helps in maintaining an efficient capital market. Given the importance of earnings forecasting, a large body of literature has developed in the last five-six decades. However, the majority of this literature has relied on restricted designs such as OLS regression. Although there is a vast literature, the models still find it difficult to forecast future earnings of loss firms or non-surviving firms [Hwang et al., 1996, Brown, 2001, Kothari et al., 2009].

In chapter 4, we theorize and empirically identify what makes earnings predictable. The majority of the proposed models rely on over-restrictive assumptions thereby suffering from selection and design bias. For example, models assume that the set of predictors would be perpetual. On the other side, the extant methods are also based on historical accounting numbers which are not forward-looking in nature. We theorize that earnings predictability can be improved by designing a framework that can combine several independent frameworks thereby achieving more generalization. Such a framework would also result in reducing the selection and design bias and therefore being more robust. Since accounting information is multi-dimensional, we combine both hard information such as historical accounting numbers, and soft information such as raw business text from the annual financial reports to design our framework.

We introduce a novel hybrid method to forecast future earnings using stacking [LeDell, 2015, Michailidis, 2017] that can accommodate several and typically diverse machine learning models. We combine one Auto Regressive model, one LightGBM model [Ke et al., 2017], and one RoBERTa [Liu et al., 2019] model into one framework. We use an exhaustive set of predictors provided by the extant literature to train the LightGBM model and use MD&A sections from annual 10-K reports to train the RoBERTa model. Ours is also the first method to adopt a textual model to forecast future earnings.

Moreover, the majority of the extant literature uses a scale-dependent metric to evaluate the forecasting models while it is still not clear if forecasting errors are biased with scale [Cheong and Thomas, 2011, 2018]. We put forward a new scale-independent evaluation metric (SAFE) to evaluate the model. We find that our stack ensemble model significantly outperforms the benchmark auto-regressive model. Our proposed method is found to be significantly outperforming the benchmark model, particularly for the *hard to predict* firms such as loss firms and non-surviving firms. Our proposed framework is also scalable in nature as it can accommodate other forecasting models.

5.3 Limitations

In this section, we discuss the limitations of my thesis work. In chapter 2, we discuss how a semi-supervised machine learning framework can be used to detect anomalies in big accounting data. One limitation of our proposed method is the inclusion of domain experts. The performance of the supervised method is conditional on how domain experts are validating the anomalies detected by the unsupervised method. The business validation is also tricky because we should also find a set of non-anomalous observations to reduce the true negative errors.

The proposed framework in chapter 2 also involves the decision of selecting the subset size to employ the unsupervised algorithm. While the size of the subset should be optimized according to the hardware availability, it also would affect the validation work of the domain experts. A bigger subset size can produce more accurate anomaly detection but would also require more extensive manual labor and vice-versa.

We use publicly available data in chapter 3 and chapter 4 for accounting fraud detection and earnings forecasting respectively. Our experiment in these two chapters is limited by the set of publicly traded U.S. firms. We find empirical evidence that our proposed method is superior for these firms. However, we believe that it can potentially be generalized to other geographies as well.

Generally, when a machine learning framework is developed an enormous set of important steps are taken, such as which data to use, how to create a validation method, which models to try, how to optimize parameters, and how to evaluate the performance, etc. However, once a framework is finalized and found to be working after several validation tests and robustness checks, it is important to also create a deployment framework for the finalized method. Another limitation of my thesis is that we do not include the deployment pipeline for our proposed methods.

5.4 Directions to future research

Due to its extensive reliance, methods in financial accounting literature are overly conservative. Although machine learning literature has grown significantly, the application of advanced machine learning algorithms in accounting literature is still growing. This dissertation is joining that growing body of literature that introduces state-of-the-art machine learning algorithms in financial accounting practice.

In chapter 2, we implemented pseudo-labeling which gained popularity in the computer vision and deep learning literature [Aroyehun and Gelbukh, 2018, Ding et al., 2019]. Creating synthetic labels where the labels are difficult to obtain can be very useful. For instance, fraud detection or bankruptcy detection deals with highly imbalanced classes. Moreover, the investigation of bankruptcy or accounting fraud can take several years and that restricts us to obtain the true label of these firms in the recent data. Pseudo-labeling in these data can be very useful to obtain a working training sample.

In Chapters 3 and 4, we use deep neural networks-based advanced architectures to find contextual patterns in business texts to detect accounting frauds and forecast earnings. While we use MD&A texts from the annual 10-K reports, other text sources can be explored to extract the deeper meaning of business texts. Researchers found clues in the conference calls, and corporate social responsibility reports and extracted patterns from that [Bowen et al., 2002, Brown et al., 2004, Larcker and Zakolyukina, 2012]. Deep neural network-based models can also be employed on these different sources of data to improve the extant methods in practice. These models can further be employed to develop methods related to audit quality prediction, bankruptcy prediction, etc.

Similar to BioBERT [Lee et al., 2020] and SciBERT [Beltagy et al., 2019], future research can also be carried out to produce AuditBERT or AccountingBERT which can be fine-tuned to carry out a lot of accounting research, for instance, question answering related to the Audit industry, etc. With the advancement of machine learning, it is also possible to find patterns beyond quantitative and text data. Future research can be carried out to find patterns in audio or video data. For example, conference, and audio calls can be used to find earnings forecasts of publicly traded companies. Since a lot of quantitative accounting research depends on interview and survey data, advanced machine learning models can be employed to find patterns in such data.

At the same time, future research should be carried out regarding the deployment of machine learning models at the production level. Once a model is found to be empirically successful on the data, it is important to understand how these models can be deployed at the organizations. This can include guidelines concerning how to implement the model, what specific hardware is required to store the model, how to monitor the performance of the model, how to retrain these models, etc. Future research related to the productionalization of these advanced models would be of huge importance to the accounting practice.

A Appendices

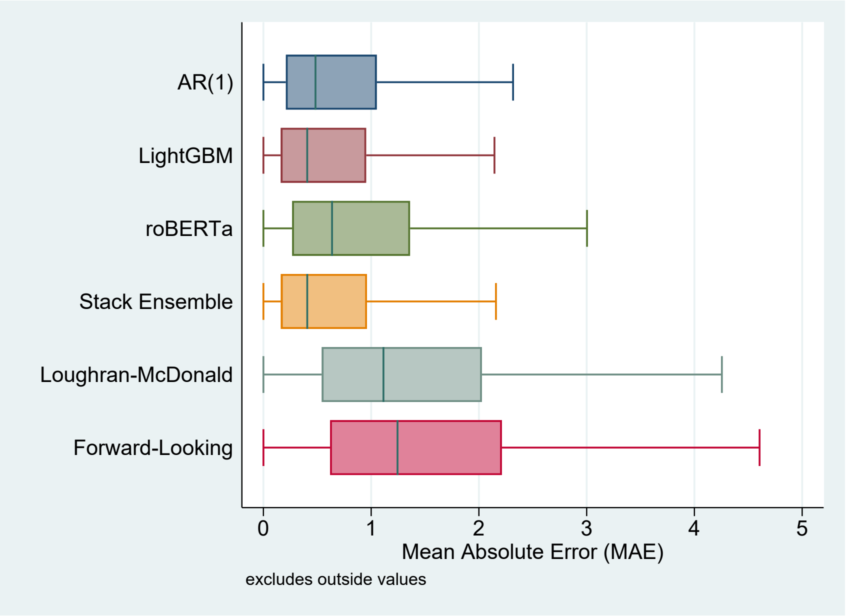


Figure A.1: Box Plots of Mean Absolute Error (MAE) per model

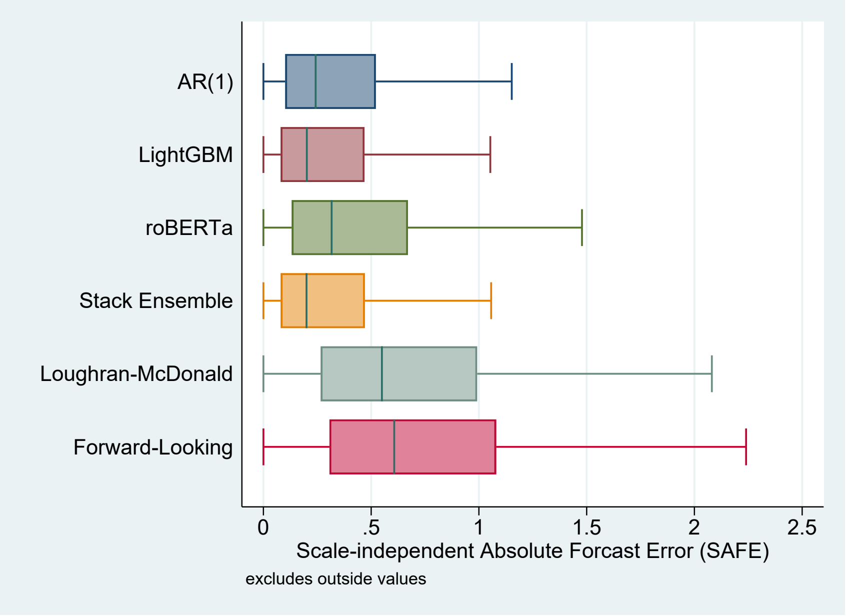


Figure A.2: Box Plots of Scale-Independent Absolute Forecast Error (SAFE) per model

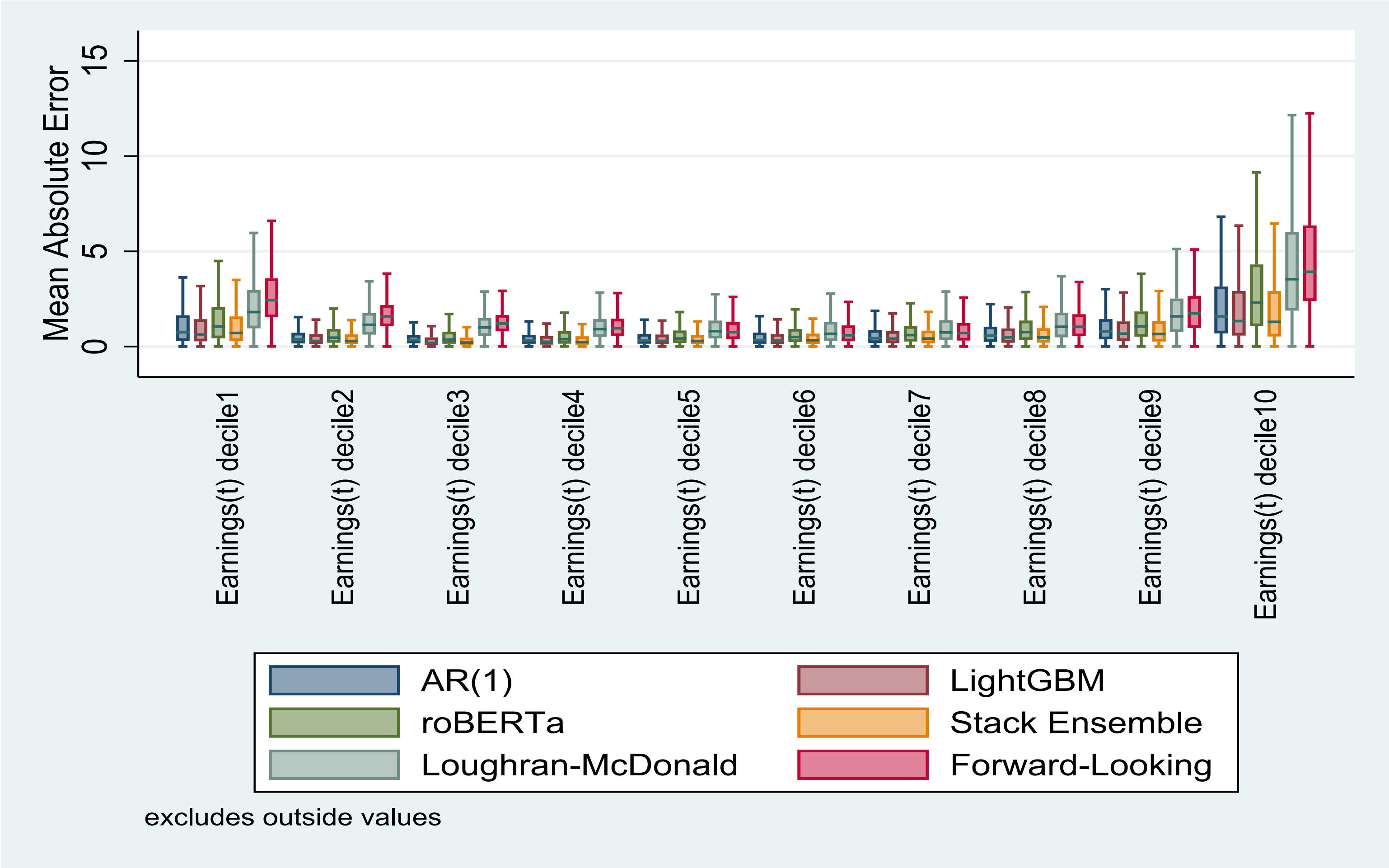


Figure A.3: Differences in Earnings Prediction Error based on Mean Absolute Error (MAE) per Earnings Decile

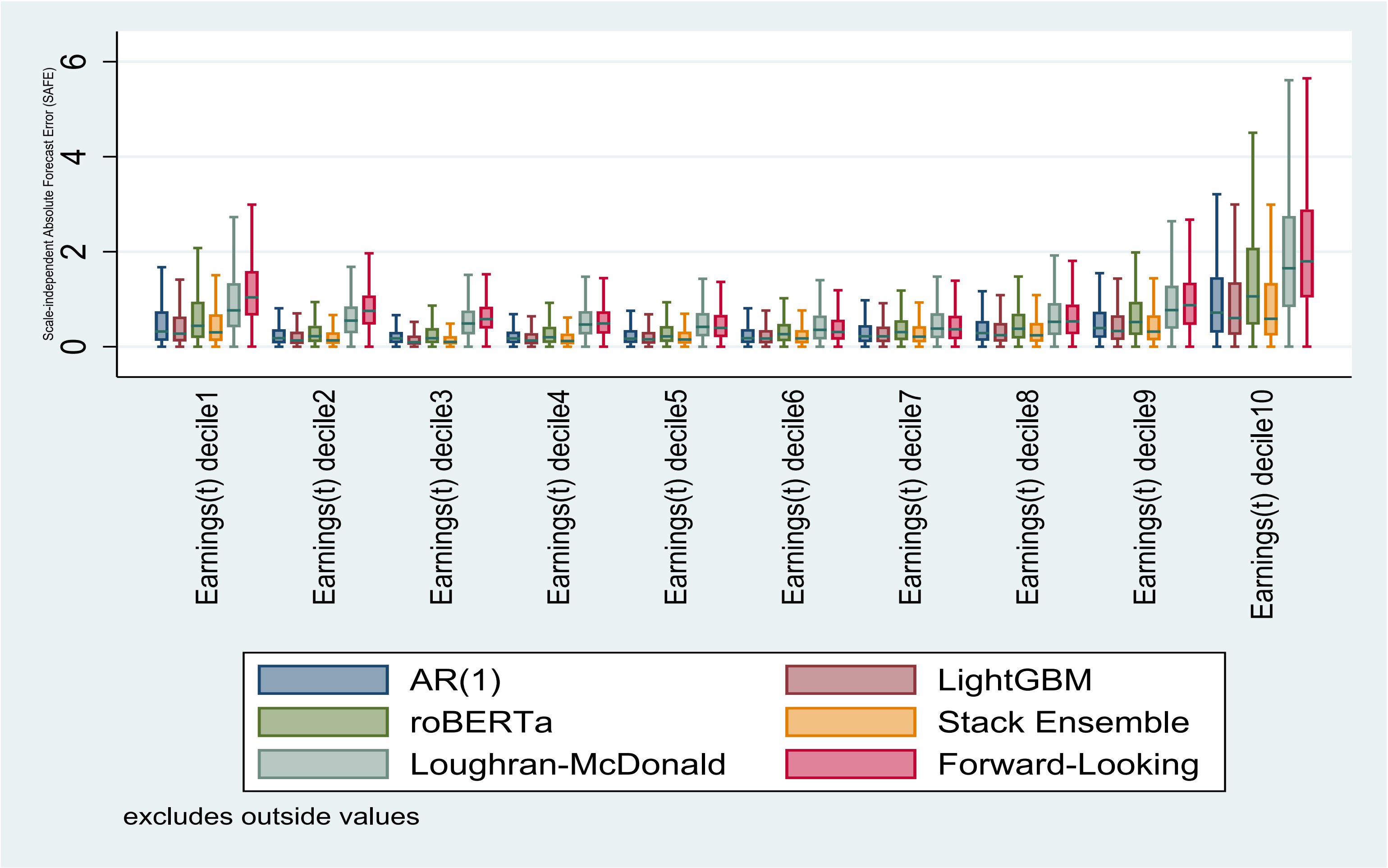


Figure A.4: Differences in Earnings Prediction Error based on Scale-Independent Absolute Forecast Error (SAFE) per Earnings Decile

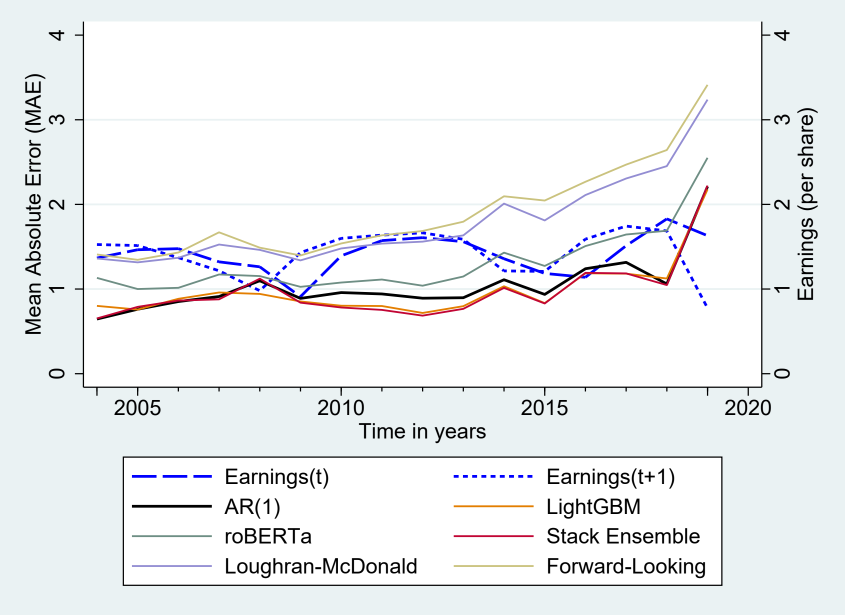


Figure A.5: Timetrends per model based on Mean Absolute Error (MAE)

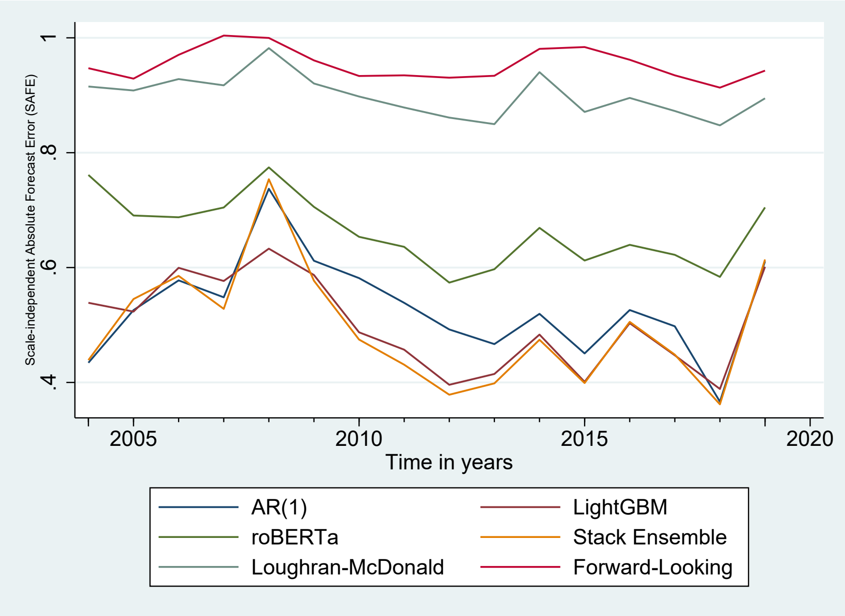


Figure A.6: Timetrends per model based on Scale-Independent Absolute Forecast Error (SAFE)

Table A.1: Time trend analysis of the Mean Absolute Error (MAE)

	AR(1)	LightGBM	RoBERTa	Stack Ensemble	Loughran_McDonald	Forward_Looking
Timetrend	0.052***	0.045**	0.064***	0.047**	0.095***	0.106***
	(3.219)	(2.261)	(3.084)	(2.327)	(4.292)	(4.789)
Constant	0.606***	0.606***	0.771***	0.577***	0.972***	0.997***
	(5.529)	(4.099)	(4.754)	(3.801)	(5.319)	(5.265)
Observations	16	16	16	16	16	16

Note: this table shows the timetrend for the period 2004-2019, where the time trend is determined as Year-2003. T-statistics based on Newey-West standard errors corrected for autocorrelation with three period lag. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.2: Time trend analysis of the Scale-Independent Absolute Forecast Error(SAFE)

	AR(1)	LightGBM	RoBERTa	Stack Ensemble	Loughran_McDonald	Forward_Looking
Timetrend	-0.005	-0.008*	-0.007***	-0.007	-0.004***	-0.002
	(-0.997)	(-2.087)	(-3.385)	(-1.366)	(-3.719)	(-1.085)
Constant	0.572***	0.572***	0.727***	0.554***	0.933***	0.967***
	(9.829)	(15.289)	(36.952)	(9.678)	(67.438)	(61.942)
Observations	16	16	16	16	16	16

Note: this table shows the timetrend for the period 2004-2019, where the time trend is determined as Year-2003. T-statistics based on Newey-West standard errors corrected for autocorrelation with three period lag. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.3: Comparing Mean Absolute Error (MAE) for AR(1) with LightGBM

	AR(1)			LightGBM			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.646	0.301	1.938	0.801	0.314	3.743	-0.155	-0.013
2005	0.762	0.327	1.986	0.758	0.344	2.776	0.004	-0.016
2006	0.852	0.340	2.235	0.885	0.353	2.099	-0.032	-0.014
2007	0.912	0.426	1.930	0.959	0.391	1.928	-0.047	0.034
2008	1.098	0.535	1.960	0.943	0.469	1.559	0.155**	0.066**
2009	0.890	0.440	1.407	0.854	0.408	1.403	0.036	0.032
2010	0.959	0.550	1.992	0.803	0.368	1.968	0.155**	0.181***
2011	0.942	0.545	2.190	0.800	0.340	2.144	0.143*	0.205***
2012	0.892	0.529	1.564	0.717	0.330	1.658	0.175***	0.198***
2013	0.898	0.516	1.890	0.798	0.332	2.141	0.100	0.183***
2014	1.110	0.447	2.958	1.033	0.370	3.125	0.077	0.077***
2015	0.937	0.435	1.964	0.834	0.389	2.352	0.103	0.047**
2016	1.240	0.526	2.688	1.186	0.502	3.007	0.054	0.025
2017	1.316	0.611	3.126	1.182	0.551	4.390	0.134	0.060**
2018	1.060	0.445	2.235	1.125	0.445	4.696	-0.064	-0.000
2019	2.211	0.637	10.283	2.177	0.569	10.586	0.035	0.067*
ALL	1.099	0.483	3.663	1.039	0.407	4.128	0.061**	0.077***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the AR(1) prediction model and LightGBM model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.4: Comparing Scale-Independent Absolute Forecast Error (SAFE) for AR(1) with LightGBM

	AR(1)			LightGBM			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.434	0.202	1.303	0.539	0.211	2.517	-0.104	-0.009
2005	0.526	0.226	1.371	0.523	0.238	1.917	0.003	-0.011
2006	0.578	0.230	1.515	0.600	0.239	1.423	-0.022	-0.009
2007	0.548	0.256	1.160	0.577	0.235	1.159	-0.028	0.020
2008	0.737	0.360	1.316	0.633	0.315	1.047	0.104**	0.045**
2009	0.612	0.303	0.968	0.587	0.280	0.965	0.025	0.022
2010	0.582	0.334	1.209	0.487	0.224	1.194	0.094**	0.110***
2011	0.539	0.311	1.252	0.457	0.194	1.226	0.082*	0.117***
2012	0.492	0.292	0.863	0.396	0.182	0.915	0.096***	0.109***
2013	0.467	0.268	0.983	0.415	0.172	1.114	0.052	0.095***
2014	0.519	0.209	1.384	0.483	0.173	1.462	0.036	0.036***
2015	0.451	0.209	0.945	0.401	0.187	1.131	0.049	0.022**
2016	0.526	0.223	1.140	0.503	0.213	1.276	0.023	0.011
2017	0.498	0.231	1.183	0.447	0.209	1.661	0.051	0.023**
2018	0.367	0.154	0.773	0.389	0.154	1.623	-0.022	-0.000
2019	0.611	0.176	2.841	0.601	0.157	2.925	0.010	0.019*
ALL	0.523	0.242	1.371	0.492	0.202	1.584	0.031***	0.041***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the AR(1) prediction model and LightGBM model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.5: Comparing Mean Absolute Error (MAE) for AR(1) with RoBERTa

	AR(1)			RoBERTa			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.646	0.301	1.938	1.132	0.557	4.387	-0.486***	-0.256***
2005	0.762	0.327	1.986	1.000	0.534	3.448	-0.239*	-0.208***
2006	0.852	0.340	2.235	1.015	0.562	2.390	-0.162	-0.222***
2007	0.912	0.426	1.930	1.172	0.626	1.987	-0.260***	-0.200***
2008	1.098	0.535	1.960	1.153	0.653	1.774	-0.055	-0.118***
2009	0.890	0.440	1.407	1.026	0.561	1.739	-0.136**	-0.120***
2010	0.959	0.550	1.992	1.077	0.563	2.427	-0.118	-0.013
2011	0.942	0.545	2.190	1.113	0.532	2.654	-0.171*	0.014
2012	0.892	0.529	1.564	1.040	0.570	2.068	-0.148**	-0.041
2013	0.898	0.516	1.890	1.148	0.562	2.834	-0.251***	-0.047*
2014	1.110	0.447	2.958	1.430	0.641	3.701	-0.320***	-0.194***
2015	0.937	0.435	1.964	1.273	0.638	3.406	-0.336***	-0.202***
2016	1.240	0.526	2.688	1.508	0.708	4.143	-0.268**	-0.181***
2017	1.316	0.611	3.126	1.644	0.812	5.889	-0.329**	-0.200***
2018	1.060	0.445	2.235	1.689	0.727	6.671	-0.629***	-0.282***
2019	2.211	0.637	10.283	2.551	0.863	11.523	-0.339	-0.226***
ALL	1.099	0.483	3.663	1.381	0.637	4.949	-0.282***	-0.154***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the AR(1) prediction model and RoBERTa model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.6: Comparing Scale-Independent Absolute Forecast Error (SAFE) for AR(1) with RoBERTa

	AR(1)			RoBERTa			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.434	0.202	1.303	0.761	0.374	2.950	-0.327***	-0.172***
2005	0.526	0.226	1.371	0.691	0.368	2.381	-0.165*	-0.143***
2006	0.578	0.230	1.515	0.688	0.381	1.620	-0.110	-0.150***
2007	0.548	0.256	1.160	0.705	0.376	1.194	-0.156***	-0.120***
2008	0.737	0.360	1.316	0.774	0.439	1.191	-0.037	-0.079***
2009	0.612	0.303	0.968	0.706	0.386	1.196	-0.094**	-0.083***
2010	0.582	0.334	1.209	0.654	0.342	1.472	-0.072	-0.008
2011	0.539	0.311	1.252	0.636	0.304	1.517	-0.097*	0.008
2012	0.492	0.292	0.863	0.574	0.314	1.141	-0.082**	-0.023
2013	0.467	0.268	0.983	0.597	0.292	1.473	-0.130***	-0.024*
2014	0.519	0.209	1.384	0.669	0.300	1.732	-0.150***	-0.091***
2015	0.451	0.209	0.945	0.612	0.307	1.638	-0.162***	-0.097***
2016	0.526	0.223	1.140	0.640	0.300	1.758	-0.114**	-0.077***
2017	0.498	0.231	1.183	0.622	0.307	2.229	-0.124**	-0.076***
2018	0.367	0.154	0.773	0.584	0.251	2.306	-0.217***	-0.097***
2019	0.611	0.176	2.841	0.705	0.239	3.184	-0.094	-0.063***
ALL	0.523	0.242	1.371	0.655	0.317	1.941	-0.132***	-0.074***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the AR(1) prediction model and RoBERTa model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.7: Comparing Mean Absolute Error (MAE) for AR(1) with Stack Ensemble

	AR(1)			Stack Ensemble			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.646	0.301	1.938	0.653	0.288	2.049	-0.007	0.013
2005	0.762	0.327	1.986	0.790	0.332	2.270	-0.028	-0.007
2006	0.852	0.340	2.235	0.864	0.328	2.102	-0.012	0.012
2007	0.912	0.426	1.930	0.879	0.376	1.957	0.033	0.050*
2008	1.098	0.535	1.960	1.123	0.521	2.023	-0.025	0.015
2009	0.890	0.440	1.407	0.839	0.411	1.371	0.050	0.029
2010	0.959	0.550	1.992	0.782	0.376	1.932	0.176**	0.175***
2011	0.942	0.545	2.190	0.754	0.320	2.097	0.188**	0.225***
2012	0.892	0.529	1.564	0.686	0.324	1.506	0.206***	0.204***
2013	0.898	0.516	1.890	0.766	0.321	2.025	0.131*	0.195***
2014	1.110	0.447	2.958	1.014	0.361	2.996	0.096	0.086***
2015	0.937	0.435	1.964	0.830	0.372	1.946	0.107*	0.063***
2016	1.240	0.526	2.688	1.192	0.519	2.606	0.048	0.008
2017	1.316	0.611	3.126	1.185	0.563	3.370	0.130	0.049
2018	1.060	0.445	2.235	1.047	0.437	2.540	0.013	0.007
2019	2.211	0.637	10.283	2.223	0.661	10.205	-0.011	-0.024
ALL	1.099	0.483	3.663	1.026	0.407	3.685	0.074***	0.077***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the AR(1) prediction model and the Stack Ensemble model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.8: Comparing Scale-Independent Absolute Forecast Error (SAFE) for AR(1) with Stack Ensemble

	AR(1)			Stack Ensemble			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.434	0.202	1.303	0.439	0.194	1.377	-0.005	0.009
2005	0.526	0.226	1.371	0.545	0.229	1.568	-0.019	-0.005
2006	0.578	0.230	1.515	0.586	0.222	1.424	-0.008	0.008
2007	0.548	0.256	1.160	0.528	0.226	1.176	0.020	0.030*
2008	0.737	0.360	1.316	0.754	0.350	1.358	-0.017	0.010
2009	0.612	0.303	0.968	0.577	0.283	0.943	0.035	0.020
2010	0.582	0.334	1.209	0.475	0.228	1.172	0.107**	0.106***
2011	0.539	0.311	1.252	0.431	0.183	1.199	0.108**	0.129***
2012	0.492	0.292	0.863	0.379	0.179	0.831	0.114***	0.113***
2013	0.467	0.268	0.983	0.398	0.167	1.053	0.068*	0.102***
2014	0.519	0.209	1.384	0.474	0.169	1.402	0.045	0.040***
2015	0.451	0.209	0.945	0.399	0.179	0.936	0.052*	0.030***
2016	0.526	0.223	1.140	0.505	0.220	1.105	0.021	0.003
2017	0.498	0.231	1.183	0.448	0.213	1.275	0.049	0.019
2018	0.367	0.154	0.773	0.362	0.151	0.878	0.005	0.002
2019	0.611	0.176	2.841	0.614	0.183	2.820	-0.003	-0.007
ALL	0.523	0.242	1.371	0.485	0.200	1.385	0.038***	0.042***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the AR(1) prediction model and the Stack Ensemble model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.9: Comparing Mean Absolute Error (MAE) for AR(1) with the bag-of-words model of Loughran-McDonald

	AR(1)			Loughran-McDonald			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.646	0.301	1.938	1.361	0.788	4.469	-0.715***	-0.488***
2005	0.762	0.327	1.986	1.315	0.847	3.637	-0.554***	-0.520***
2006	0.852	0.340	2.235	1.370	0.974	2.450	-0.517***	-0.634***
2007	0.912	0.426	1.930	1.526	1.032	2.046	-0.614***	-0.607***
2008	1.098	0.535	1.960	1.463	1.046	1.768	-0.365***	-0.511***
2009	0.890	0.440	1.407	1.338	0.860	1.876	-0.449***	-0.420***
2010	0.959	0.550	1.992	1.480	0.934	2.545	-0.521***	-0.383***
2011	0.942	0.545	2.190	1.537	0.970	2.717	-0.595***	-0.425***
2012	0.892	0.529	1.564	1.560	0.993	2.734	-0.668***	-0.465***
2013	0.898	0.516	1.890	1.634	1.052	2.925	-0.736***	-0.536***
2014	1.110	0.447	2.958	2.009	1.160	5.154	-0.899***	-0.713***
2015	0.937	0.435	1.964	1.811	1.196	3.598	-0.874***	-0.761***
2016	1.240	0.526	2.688	2.111	1.287	4.401	-0.871***	-0.762***
2017	1.316	0.611	3.126	2.306	1.397	6.161	-0.990***	-0.786***
2018	1.060	0.445	2.235	2.452	1.434	6.913	-1.392***	-0.991***
2019	2.211	0.637	10.283	3.238	1.494	11.689	-1.027***	-0.857***
ALL	1.099	0.483	3.663	1.890	1.114	5.217	-1.027***	-0.631***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the AR(1) prediction model and the bag-of-words model of Loughran-McDonald. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.10: Comparing Scale-Independent Absolute Forecast Error (SAFE) for AR(1) with the bag-of-words model of Loughran-McDonald

	AR(1)			Loughran-McDonald			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.434	0.202	1.303	0.915	0.530	3.005	-0.481***	-0.328***
2005	0.526	0.226	1.371	0.908	0.585	2.511	-0.382***	-0.359***
2006	0.578	0.230	1.515	0.928	0.660	1.660	-0.350***	-0.430***
2007	0.548	0.256	1.160	0.917	0.620	1.230	-0.369***	-0.365***
2008	0.737	0.360	1.316	0.982	0.702	1.187	-0.245***	-0.343***
2009	0.612	0.303	0.968	0.920	0.592	1.290	-0.309***	-0.289***
2010	0.582	0.334	1.209	0.898	0.567	1.544	-0.316***	-0.233***
2011	0.539	0.311	1.252	0.879	0.554	1.553	-0.340***	-0.243***
2012	0.492	0.292	0.863	0.861	0.548	1.509	-0.369***	-0.257***
2013	0.467	0.268	0.983	0.850	0.547	1.521	-0.383***	-0.279***
2014	0.519	0.209	1.384	0.940	0.543	2.411	-0.421***	-0.334***
2015	0.451	0.209	0.945	0.871	0.575	1.731	-0.420***	-0.366***
2016	0.526	0.223	1.140	0.895	0.546	1.867	-0.369***	-0.323***
2017	0.498	0.231	1.183	0.873	0.529	2.331	-0.375***	-0.297***
2018	0.367	0.154	0.773	0.848	0.496	2.389	-0.481***	-0.342***
2019	0.611	0.176	2.841	0.895	0.413	3.230	-0.284***	-0.237***
ALL	0.523	0.242	1.371	0.895	0.550	2.072	-0.284***	-0.308***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the AR(1) prediction model and the bag-of-words model of Loughran-McDonald. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.11: Comparing Mean Absolute Error (MAE) for AR(1) with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	AR(1)			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.646	0.301	1.938	1.409	0.865	4.468	-0.763***	-0.565***
2005	0.762	0.327	1.986	1.345	0.847	3.628	-0.584***	-0.520***
2006	0.852	0.340	2.235	1.432	0.982	2.445	-0.579***	-0.642***
2007	0.912	0.426	1.930	1.670	1.118	2.108	-0.758***	-0.693***
2008	1.098	0.535	1.960	1.489	1.083	1.795	-0.391***	-0.547***
2009	0.890	0.440	1.407	1.397	0.929	1.898	-0.507***	-0.489***
2010	0.959	0.550	1.992	1.538	0.976	2.600	-0.580***	-0.426***
2011	0.942	0.545	2.190	1.635	1.081	2.771	-0.693***	-0.536***
2012	0.892	0.529	1.564	1.686	1.146	2.859	-0.794***	-0.618***
2013	0.898	0.516	1.890	1.796	1.247	3.050	-0.898***	-0.733***
2014	1.110	0.447	2.958	2.096	1.393	3.837	-0.986***	-0.947***
2015	0.937	0.435	1.964	2.046	1.392	3.694	-1.109***	-0.957***
2016	1.240	0.526	2.688	2.267	1.432	4.461	-1.027***	-0.906***
2017	1.316	0.611	3.126	2.470	1.525	6.234	-1.155***	-0.913***
2018	1.060	0.445	2.235	2.642	1.539	7.027	-1.582***	-1.095***
2019	2.211	0.637	10.283	3.413	1.679	11.707	-1.201***	-1.043***
ALL	1.099	0.483	3.663	2.017	1.245	5.174	-0.917***	-0.761***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the AR(1) prediction model and the Forward-Looking bag-of-words model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.12: Comparing Scale-Independent Absolute Forecast Error (SAFE) for AR(1) with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	AR(1)			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.434	0.202	1.303	0.947	0.582	3.004	-0.513***	-0.380***
2005	0.526	0.226	1.371	0.929	0.585	2.505	-0.403***	-0.359***
2006	0.578	0.230	1.515	0.970	0.666	1.657	-0.393***	-0.435***
2007	0.548	0.256	1.160	1.004	0.672	1.267	-0.456***	-0.416***
2008	0.737	0.360	1.316	1.000	0.727	1.205	-0.263***	-0.367***
2009	0.612	0.303	0.968	0.961	0.639	1.305	-0.349***	-0.336***
2010	0.582	0.334	1.209	0.934	0.592	1.578	-0.352***	-0.258***
2011	0.539	0.311	1.252	0.935	0.618	1.584	-0.396***	-0.306***
2012	0.492	0.292	0.863	0.931	0.633	1.578	-0.438***	-0.341***
2013	0.467	0.268	0.983	0.934	0.648	1.586	-0.467***	-0.381***
2014	0.519	0.209	1.384	0.981	0.652	1.796	-0.461***	-0.443***
2015	0.451	0.209	0.945	0.984	0.669	1.777	-0.533***	-0.460***
2016	0.526	0.223	1.140	0.962	0.607	1.892	-0.436***	-0.384***
2017	0.498	0.231	1.183	0.935	0.577	2.359	-0.437***	-0.346***
2018	0.367	0.154	0.773	0.913	0.532	2.429	-0.547***	-0.378***
2019	0.611	0.176	2.841	0.943	0.464	3.234	-0.332***	-0.288***
ALL	0.523	0.242	1.371	0.953	0.607	2.044	-0.430***	-0.365***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the AR(1) prediction model and the Forward-Looking bag-of-words model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.13: Comparing Mean Absolute Error (MAE) for LightGBM with RoBERTa

	LightGBM			RoBERTa			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.801	0.314	3.743	1.132	0.557	4.387	-0.331*	-0.242***
2005	0.758	0.344	2.776	1.000	0.534	3.448	-0.242*	-0.191***
2006	0.885	0.353	2.099	1.015	0.562	2.390	-0.130	-0.208***
2007	0.959	0.391	1.928	1.172	0.626	1.987	-0.213**	-0.233***
2008	0.943	0.469	1.559	1.153	0.653	1.774	-0.211***	-0.184***
2009	0.854	0.408	1.403	1.026	0.561	1.739	-0.173***	-0.153***
2010	0.803	0.368	1.968	1.077	0.563	2.427	-0.274***	-0.194***
2011	0.800	0.340	2.144	1.113	0.532	2.654	-0.313***	-0.192***
2012	0.717	0.330	1.658	1.040	0.570	2.068	-0.322***	-0.239***
2013	0.798	0.332	2.141	1.148	0.562	2.834	-0.351***	-0.230***
2014	1.033	0.370	3.125	1.430	0.641	3.701	-0.397***	-0.271***
2015	0.834	0.389	2.352	1.273	0.638	3.406	-0.439***	-0.249***
2016	1.186	0.502	3.007	1.508	0.708	4.143	-0.322***	-0.206***
2017	1.182	0.551	4.390	1.644	0.812	5.889	-0.462***	-0.260***
2018	1.125	0.445	4.696	1.689	0.727	6.671	-0.565***	-0.282***
2019	2.177	0.569	10.586	2.551	0.863	11.523	-0.374	-0.293***
ALL	1.039	0.407	4.128	1.381	0.637	4.949	-0.343***	-0.230***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the LightGBM prediction model and the RoBERTa model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.14: Comparing Scale-Independent Absolute Forecast Error (SAFE) for LightGBM with RoBERTa

	LightGBM			RoBERTa			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.539	0.211	2.517	0.761	0.374	2.950	-0.223*	-0.163***
2005	0.523	0.238	1.917	0.691	0.368	2.381	-0.167*	-0.132***
2006	0.600	0.239	1.423	0.688	0.381	1.620	-0.088	-0.141***
2007	0.577	0.235	1.159	0.705	0.376	1.194	-0.128**	-0.140***
2008	0.633	0.315	1.047	0.774	0.439	1.191	-0.142***	-0.124***
2009	0.587	0.280	0.965	0.706	0.386	1.196	-0.119***	-0.105***
2010	0.487	0.224	1.194	0.654	0.342	1.472	-0.166***	-0.118***
2011	0.457	0.194	1.226	0.636	0.304	1.517	-0.179***	-0.110***
2012	0.396	0.182	0.915	0.574	0.314	1.141	-0.178***	-0.132***
2013	0.415	0.172	1.114	0.597	0.292	1.473	-0.182***	-0.119***
2014	0.483	0.173	1.462	0.669	0.300	1.732	-0.186***	-0.127***
2015	0.401	0.187	1.131	0.612	0.307	1.638	-0.211***	-0.120***
2016	0.503	0.213	1.276	0.640	0.300	1.758	-0.136***	-0.087***
2017	0.447	0.209	1.661	0.622	0.307	2.229	-0.175***	-0.099***
2018	0.389	0.154	1.623	0.584	0.251	2.306	-0.195***	-0.097***
2019	0.601	0.157	2.925	0.705	0.239	3.184	-0.103	-0.081***
ALL	0.492	0.202	1.584	0.655	0.317	1.941	-0.163***	-0.115***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the LightGBM prediction model and the RoBERTa model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.15: Comparing Mean Absolute Error (MAE) for LightGBM with Stack Ensemble

	LightGBM			Stack Ensemble			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.801	0.314	3.743	0.653	0.288	2.049	0.148	0.026
2005	0.758	0.344	2.776	0.790	0.332	2.270	-0.032	0.009
2006	0.885	0.353	2.099	0.864	0.328	2.102	0.021	0.025
2007	0.959	0.391	1.928	0.879	0.376	1.957	0.081	0.017
2008	0.943	0.469	1.559	1.123	0.521	2.023	-0.180**	-0.052
2009	0.854	0.408	1.403	0.839	0.411	1.371	0.014	-0.003
2010	0.803	0.368	1.968	0.782	0.376	1.932	0.021	-0.006
2011	0.800	0.340	2.144	0.754	0.320	2.097	0.046	0.020
2012	0.717	0.330	1.658	0.686	0.324	1.506	0.031	0.006
2013	0.798	0.332	2.141	0.766	0.321	2.025	0.031	0.012
2014	1.033	0.370	3.125	1.014	0.361	2.996	0.019	0.009
2015	0.834	0.389	2.352	0.830	0.372	1.946	0.004	0.016
2016	1.186	0.502	3.007	1.192	0.519	2.606	-0.005	-0.017
2017	1.182	0.551	4.390	1.185	0.563	3.370	-0.003	-0.011
2018	1.125	0.445	4.696	1.047	0.437	2.540	0.077	0.007
2019	2.177	0.569	10.586	2.223	0.661	10.205	-0.046	-0.091***
ALL	1.039	0.407	4.128	1.026	0.407	3.685	0.013	-0.000

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the LightGBM prediction model and the Stack Ensemble model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.16: Comparing Scale-Independent Absolute Forecast Error (SAFE) for LightGBM with Stack Enesmble

	LightGBM			Stack Ensemble			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.539	0.211	2.517	0.439	0.194	1.377	0.100	0.018
2005	0.523	0.238	1.917	0.545	0.229	1.568	-0.022	0.006
2006	0.600	0.239	1.423	0.586	0.222	1.424	0.014	0.017
2007	0.577	0.235	1.159	0.528	0.226	1.176	0.048	0.010
2008	0.633	0.315	1.047	0.754	0.350	1.358	-0.121**	-0.035
2009	0.587	0.280	0.965	0.577	0.283	0.943	0.010	-0.002
2010	0.487	0.224	1.194	0.475	0.228	1.172	0.013	-0.004
2011	0.457	0.194	1.226	0.431	0.183	1.199	0.026	0.012
2012	0.396	0.182	0.915	0.379	0.179	0.831	0.017	0.003
2013	0.415	0.172	1.114	0.398	0.167	1.053	0.016	0.006
2014	0.483	0.173	1.462	0.474	0.169	1.402	0.009	0.004
2015	0.401	0.187	1.131	0.399	0.179	0.936	0.002	0.008
2016	0.503	0.213	1.276	0.505	0.220	1.105	-0.002	-0.007
2017	0.447	0.209	1.661	0.448	0.213	1.275	-0.001	-0.004
2018	0.389	0.154	1.623	0.362	0.151	0.878	0.027	0.003
2019	0.601	0.157	2.925	0.614	0.183	2.820	-0.013	-0.025***
ALL	0.492	0.202	1.584	0.485	0.200	1.385	0.007	0.001

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the LightGBM prediction model and the Stack Ensemble model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.17: Comparing Mean Absolute Error (MAE) for LightGBM with the bag-of-words model of Loughran-McDonald

	LightGBM			Loughran-McDonald			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.801	0.314	3.743	1.361	0.788	4.469	-0.560***	-0.474***
2005	0.758	0.344	2.776	1.315	0.847	3.637	-0.558***	-0.503***
2006	0.885	0.353	2.099	1.370	0.974	2.450	-0.485***	-0.621***
2007	0.959	0.391	1.928	1.526	1.032	2.046	-0.567***	-0.641***
2008	0.943	0.469	1.559	1.463	1.046	1.768	-0.520***	-0.577***
2009	0.854	0.408	1.403	1.338	0.860	1.876	-0.485***	-0.453***
2010	0.803	0.368	1.968	1.480	0.934	2.545	-0.677***	-0.564***
2011	0.800	0.340	2.144	1.537	0.970	2.717	-0.737***	-0.630***
2012	0.717	0.330	1.658	1.560	0.993	2.734	-0.843***	-0.663***
2013	0.798	0.332	2.141	1.634	1.052	2.925	-0.836***	-0.719***
2014	1.033	0.370	3.125	2.009	1.160	5.154	-0.976***	-0.790***
2015	0.834	0.389	2.352	1.811	1.196	3.598	-0.977***	-0.807***
2016	1.186	0.502	3.007	2.111	1.287	4.401	-0.925***	-0.787***
2017	1.182	0.551	4.390	2.306	1.397	6.161	-1.124***	-0.847***
2018	1.125	0.445	4.696	2.452	1.434	6.913	-1.328***	-0.990***
2019	2.177	0.569	10.586	3.238	1.494	11.689	-1.061***	-0.924***
ALL	1.039	0.407	4.128	1.890	1.114	5.217	-0.852***	-0.708***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the LightGBM prediction model and the bag-of-words model of Loughran-McDonald. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.18: Comparing Scale-Independent Absolute Forecast Error (SAFE) for LightGBM with the bag-of-words model of Loughran-McDonald

	LightGBM			Loughran-McDonald			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.539	0.211	2.517	0.915	0.530	3.005	-0.377***	-0.319***
2005	0.523	0.238	1.917	0.908	0.585	2.511	-0.385***	-0.347***
2006	0.600	0.239	1.423	0.928	0.660	1.660	-0.329***	-0.421***
2007	0.577	0.235	1.159	0.917	0.620	1.230	-0.341***	-0.385***
2008	0.633	0.315	1.047	0.982	0.702	1.187	-0.349***	-0.388***
2009	0.587	0.280	0.965	0.920	0.592	1.290	-0.333***	-0.311***
2010	0.487	0.224	1.194	0.898	0.567	1.544	-0.411***	-0.342***
2011	0.457	0.194	1.226	0.879	0.554	1.553	-0.422***	-0.360***
2012	0.396	0.182	0.915	0.861	0.548	1.509	-0.465***	-0.366***
2013	0.415	0.172	1.114	0.850	0.547	1.521	-0.435***	-0.374***
2014	0.483	0.173	1.462	0.940	0.543	2.411	-0.457***	-0.370***
2015	0.401	0.187	1.131	0.871	0.575	1.731	-0.470***	-0.388***
2016	0.503	0.213	1.276	0.895	0.546	1.867	-0.392***	-0.334***
2017	0.447	0.209	1.661	0.873	0.529	2.331	-0.425***	-0.320***
2018	0.389	0.154	1.623	0.848	0.496	2.389	-0.459***	-0.342***
2019	0.601	0.157	2.925	0.895	0.413	3.230	-0.293***	-0.255***
ALL	0.492	0.202	1.584	0.895	0.550	2.072	-0.403***	-0.348***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the LightGBM prediction model and the bag-of-words model of Loughran-McDonald. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.19: Comparing Mean Absolute Error (MAE) for LightGBM with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	LightGBM			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.801	0.314	3.743	1.409	0.865	4.468	-0.607***	-0.551***
2005	0.758	0.344	2.776	1.345	0.847	3.628	-0.587***	-0.503***
2006	0.885	0.353	2.099	1.432	0.982	2.445	-0.547***	-0.629***
2007	0.959	0.391	1.928	1.670	1.118	2.108	-0.711***	-0.726***
2008	0.943	0.469	1.559	1.489	1.083	1.795	-0.546***	-0.614***
2009	0.854	0.408	1.403	1.397	0.929	1.898	-0.543***	-0.521***
2010	0.803	0.368	1.968	1.538	0.976	2.600	-0.735***	-0.607***
2011	0.800	0.340	2.144	1.635	1.081	2.771	-0.835***	-0.741***
2012	0.717	0.330	1.658	1.686	1.146	2.859	-0.969***	-0.816***
2013	0.798	0.332	2.141	1.796	1.247	3.050	-0.998***	-0.916***
2014	1.033	0.370	3.125	2.096	1.393	3.837	-1.063***	-1.024***
2015	0.834	0.389	2.352	2.046	1.392	3.694	-1.212***	-1.003***
2016	1.186	0.502	3.007	2.267	1.432	4.461	-1.081***	-0.931***
2017	1.182	0.551	4.390	2.470	1.525	6.234	-1.288***	-0.974***
2018	1.125	0.445	4.696	2.642	1.539	7.027	-1.518***	-1.094***
2019	2.177	0.569	10.586	3.413	1.679	11.707	-1.236***	-1.110***
ALL	1.039	0.407	4.128	2.017	1.245	5.174	-1.236***	-0.838***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the LightGBM prediction model and the Forward-Looking bag-of-words model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.20: Comparing Scale-Independent Absolute Forecast Error (SAFE) for LightGBM with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	LightGBM			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.539	0.211	2.517	0.947	0.582	3.004	-0.408***	-0.371***
2005	0.523	0.238	1.917	0.929	0.585	2.505	-0.406***	-0.348***
2006	0.600	0.239	1.423	0.970	0.666	1.657	-0.371***	-0.426***
2007	0.577	0.235	1.159	1.004	0.672	1.267	-0.427***	-0.437***
2008	0.633	0.315	1.047	1.000	0.727	1.205	-0.367***	-0.412***
2009	0.587	0.280	0.965	0.961	0.639	1.305	-0.374***	-0.359***
2010	0.487	0.224	1.194	0.934	0.592	1.578	-0.446***	-0.368***
2011	0.457	0.194	1.226	0.935	0.618	1.584	-0.478***	-0.424***
2012	0.396	0.182	0.915	0.931	0.633	1.578	-0.535***	-0.450***
2013	0.415	0.172	1.114	0.934	0.648	1.586	-0.519***	-0.476***
2014	0.483	0.173	1.462	0.981	0.652	1.796	-0.497***	-0.479***
2015	0.401	0.187	1.131	0.984	0.669	1.777	-0.583***	-0.483***
2016	0.503	0.213	1.276	0.962	0.607	1.892	-0.459***	-0.395***
2017	0.447	0.209	1.661	0.935	0.577	2.359	-0.487***	-0.368***
2018	0.389	0.154	1.623	0.913	0.532	2.429	-0.525***	-0.378***
2019	0.601	0.157	2.925	0.943	0.464	3.234	-0.341***	-0.307***
ALL	0.492	0.202	1.584	0.953	0.607	2.044	-0.341***	-0.406***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the LightGBM prediction model and the Forward-Looking bag-of-words model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.21: Comparing Mean Absolute Error (MAE) for RoBERTa with Stack Ensemble

	RoBERTa			Stack Ensemble			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	1.132	0.557	4.387	0.653	0.288	2.049	0.479***	0.269***
2005	1.000	0.534	3.448	0.790	0.332	2.270	0.211	0.200***
2006	1.015	0.562	2.390	0.864	0.328	2.102	0.151	0.233***
2007	1.172	0.626	1.987	0.879	0.376	1.957	0.294***	0.250***
2008	1.153	0.653	1.774	1.123	0.521	2.023	0.031	0.132***
2009	1.026	0.561	1.739	0.839	0.411	1.371	0.187***	0.150***
2010	1.077	0.563	2.427	0.782	0.376	1.932	0.295***	0.188***
2011	1.113	0.532	2.654	0.754	0.320	2.097	0.359***	0.212***
2012	1.040	0.570	2.068	0.686	0.324	1.506	0.354***	0.245***
2013	1.148	0.562	2.834	0.766	0.321	2.025	0.382***	0.242***
2014	1.430	0.641	3.701	1.014	0.361	2.996	0.416***	0.280***
2015	1.273	0.638	3.406	0.830	0.372	1.946	0.443***	0.265***
2016	1.508	0.708	4.143	1.192	0.519	2.606	0.316***	0.189***
2017	1.644	0.812	5.889	1.185	0.563	3.370	0.459***	0.249***
2018	1.689	0.727	6.671	1.047	0.437	2.540	0.642***	0.289***
2019	2.551	0.863	11.523	2.223	0.661	10.205	0.328	0.203***
ALL	1.381	0.637	4.949	1.026	0.407	3.685	0.356***	0.230***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the RoBERTa prediction model and the Stack Ensemble model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.22: Comparing Scale-Independent Absolute Forecast Error (SAFE) for RoBERTa with Stack Ensemble

	RoBERTa			Stack Ensemble			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.761	0.374	2.950	0.439	0.194	1.377	0.322***	0.181***
2005	0.691	0.368	2.381	0.545	0.229	1.568	0.145	0.138***
2006	0.688	0.381	1.620	0.586	0.222	1.424	0.102	0.158***
2007	0.705	0.376	1.194	0.528	0.226	1.176	0.176***	0.150***
2008	0.774	0.439	1.191	0.754	0.350	1.358	0.021	0.089***
2009	0.706	0.386	1.196	0.577	0.283	0.943	0.129***	0.103***
2010	0.654	0.342	1.472	0.475	0.228	1.172	0.179***	0.114***
2011	0.636	0.304	1.517	0.431	0.183	1.199	0.205***	0.121***
2012	0.574	0.314	1.141	0.379	0.179	0.831	0.195***	0.135***
2013	0.597	0.292	1.473	0.398	0.167	1.053	0.199***	0.126***
2014	0.669	0.300	1.732	0.474	0.169	1.402	0.195***	0.131***
2015	0.612	0.307	1.638	0.399	0.179	0.936	0.213***	0.127***
2016	0.640	0.300	1.758	0.505	0.220	1.105	0.134***	0.080***
2017	0.622	0.307	2.229	0.448	0.213	1.275	0.174***	0.094***
2018	0.584	0.251	2.306	0.362	0.151	0.878	0.222***	0.100***
2019	0.705	0.239	3.184	0.614	0.183	2.820	0.091	0.056***
ALL	0.655	0.317	1.941	0.485	0.200	1.385	0.170***	0.116***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the RoBERTa prediction model and the Stack Ensemble model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.23: Comparing Mean Absolute Error (MAE) for RoBERTa with the bag-of-words model of LoughranMcDonald

	RoBERTa			LoughranMcDonald			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	1.132	0.557	4.387	1.361	0.788	4.469	-0.229	-0.232***
2005	1.000	0.534	3.448	1.315	0.847	3.637	-0.315*	-0.312***
2006	1.015	0.562	2.390	1.370	0.974	2.450	-0.355***	-0.413***
2007	1.172	0.626	1.987	1.526	1.032	2.046	-0.354***	-0.408***
2008	1.153	0.653	1.774	1.463	1.046	1.768	-0.309***	-0.393***
2009	1.026	0.561	1.739	1.338	0.860	1.876	-0.312***	-0.300***
2010	1.077	0.563	2.427	1.480	0.934	2.545	-0.403***	-0.370***
2011	1.113	0.532	2.654	1.537	0.970	2.717	-0.424***	-0.438***
2012	1.040	0.570	2.068	1.560	0.993	2.734	-0.521***	-0.424***
2013	1.148	0.562	2.834	1.634	1.052	2.925	-0.486***	-0.489***
2014	1.430	0.641	3.701	2.009	1.160	5.154	-0.579***	-0.519***
2015	1.273	0.638	3.406	1.811	1.196	3.598	-0.538***	-0.558***
2016	1.508	0.708	4.143	2.111	1.287	4.401	-0.603***	-0.581***
2017	1.644	0.812	5.889	2.306	1.397	6.161	-0.662***	-0.586***
2018	1.689	0.727	6.671	2.452	1.434	6.913	-0.763***	-0.709***
2019	2.551	0.863	11.523	3.238	1.494	11.689	-0.687*	-0.631***
ALL	1.381	0.637	4.949	1.890	1.114	5.217	-0.509***	-0.477***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the RoBERTa prediction model and the bag-of-words model of Loughran-McDonald. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.24: Comparing Scale-Independent Absolute Forecast Error (SAFE) for RoBERTa with the bag-of-words model of LoughranMcDonald

	RoBERTa			LoughranMcDonald			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.761	0.374	2.950	0.915	0.530	3.005	-0.154	-0.156***
2005	0.691	0.368	2.381	0.908	0.585	2.511	-0.218*	-0.215***
2006	0.688	0.381	1.620	0.928	0.660	1.660	-0.241***	-0.280***
2007	0.705	0.376	1.194	0.917	0.620	1.230	-0.213***	-0.245***
2008	0.774	0.439	1.191	0.982	0.702	1.187	-0.208***	-0.264***
2009	0.706	0.386	1.196	0.920	0.592	1.290	-0.215***	-0.206***
2010	0.654	0.342	1.472	0.898	0.567	1.544	-0.244***	-0.225***
2011	0.636	0.304	1.517	0.879	0.554	1.553	-0.243***	-0.250***
2012	0.574	0.314	1.141	0.861	0.548	1.509	-0.287***	-0.234***
2013	0.597	0.292	1.473	0.850	0.547	1.521	-0.253***	-0.254***
2014	0.669	0.300	1.732	0.940	0.543	2.411	-0.271***	-0.243***
2015	0.612	0.307	1.638	0.871	0.575	1.731	-0.259***	-0.269***
2016	0.640	0.300	1.758	0.895	0.546	1.867	-0.256***	-0.246***
2017	0.622	0.307	2.229	0.873	0.529	2.331	-0.250***	-0.222***
2018	0.584	0.251	2.306	0.848	0.496	2.389	-0.264***	-0.245***
2019	0.705	0.239	3.184	0.895	0.413	3.230	-0.190*	-0.174***
ALL	0.655	0.317	1.941	0.895	0.550	2.072	-0.240***	-0.233***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the RoBERTa prediction model and the bag-of-words model of Loughran-McDonald. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.25: Comparing Mean Absolute Error (MAE) for RoBERTa with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	RoBERTa			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	1.132	0.557	4.387	1.409	0.865	4.468	-0.276	-0.309***
2005	1.000	0.534	3.448	1.345	0.847	3.628	-0.345**	-0.312***
2006	1.015	0.562	2.390	1.432	0.982	2.445	-0.417***	-0.420***
2007	1.172	0.626	1.987	1.670	1.118	2.108	-0.498***	-0.493***
2008	1.153	0.653	1.774	1.489	1.083	1.795	-0.336***	-0.429***
2009	1.026	0.561	1.739	1.397	0.929	1.898	-0.371***	-0.368***
2010	1.077	0.563	2.427	1.538	0.976	2.600	-0.461***	-0.413***
2011	1.113	0.532	2.654	1.635	1.081	2.771	-0.522***	-0.550***
2012	1.040	0.570	2.068	1.686	1.146	2.859	-0.646***	-0.576***
2013	1.148	0.562	2.834	1.796	1.247	3.050	-0.648***	-0.687***
2014	1.430	0.641	3.701	2.096	1.393	3.837	-0.666***	-0.753***
2015	1.273	0.638	3.406	2.046	1.392	3.694	-0.773***	-0.754***
2016	1.508	0.708	4.143	2.267	1.432	4.461	-0.759***	-0.724***
2017	1.644	0.812	5.889	2.470	1.525	6.234	-0.826***	-0.713***
2018	1.689	0.727	6.671	2.642	1.539	7.027	-0.953***	-0.813***
2019	2.551	0.863	11.523	3.413	1.679	11.707	-0.862**	-0.816***
ALL	1.381	0.637	4.949	2.017	1.245	5.174	-0.635***	-0.608***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the RoBERTa prediction model and the Forward-Looking bag-of-words model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.26: Comparing Scale-Independent Absolute Forecast Error (SAFE) for RoBERTa with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	RoBERTa			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.761	0.374	2.950	0.947	0.582	3.004	-0.186	-0.208***
2005	0.691	0.368	2.381	0.929	0.585	2.505	-0.238**	-0.216***
2006	0.688	0.381	1.620	0.970	0.666	1.657	-0.283***	-0.285***
2007	0.705	0.376	1.194	1.004	0.672	1.267	-0.299***	-0.296***
2008	0.774	0.439	1.191	1.000	0.727	1.205	-0.225***	-0.288***
2009	0.706	0.386	1.196	0.961	0.639	1.305	-0.255***	-0.253***
2010	0.654	0.342	1.472	0.934	0.592	1.578	-0.280***	-0.251***
2011	0.636	0.304	1.517	0.935	0.618	1.584	-0.299***	-0.314***
2012	0.574	0.314	1.141	0.931	0.633	1.578	-0.357***	-0.318***
2013	0.597	0.292	1.473	0.934	0.648	1.586	-0.337***	-0.357***
2014	0.669	0.300	1.732	0.981	0.652	1.796	-0.312***	-0.352***
2015	0.612	0.307	1.638	0.984	0.669	1.777	-0.372***	-0.363***
2016	0.640	0.300	1.758	0.962	0.607	1.892	-0.322***	-0.307***
2017	0.622	0.307	2.229	0.935	0.577	2.359	-0.312***	-0.270***
2018	0.584	0.251	2.306	0.913	0.532	2.429	-0.329***	-0.281***
2019	0.705	0.239	3.184	0.943	0.464	3.234	-0.238**	-0.226***
ALL	0.655	0.317	1.941	0.953	0.607	2.044	-0.298***	-0.291***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the RoBERTa prediction model and the Forward-Looking bag-of-words model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.27: Comparing Mean Absolute Error (MAE) for Stack Ensemble with the bag-of-words model of LoughranMcDonald

	RoBERTa			Stack Ensemble			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.653	0.288	2.049	1.361	0.788	4.469	-0.708***	-0.500***
2005	0.790	0.332	2.270	1.315	0.847	3.637	-0.526***	-0.512***
2006	0.864	0.328	2.102	1.370	0.974	2.450	-0.506***	-0.646***
2007	0.879	0.376	1.957	1.526	1.032	2.046	-0.647***	-0.658***
2008	1.123	0.521	2.023	1.463	1.046	1.768	-0.340***	-0.525***
2009	0.839	0.411	1.371	1.338	0.860	1.876	-0.499***	-0.450***
2010	0.782	0.376	1.932	1.480	0.934	2.545	-0.697***	-0.558***
2011	0.754	0.320	2.097	1.537	0.970	2.717	-0.783***	-0.650***
2012	0.686	0.324	1.506	1.560	0.993	2.734	-0.874***	-0.669***
2013	0.766	0.321	2.025	1.634	1.052	2.925	-0.868***	-0.731***
2014	1.014	0.361	2.996	2.009	1.160	5.154	-0.996***	-0.799***
2015	0.830	0.372	1.946	1.811	1.196	3.598	-0.981***	-0.823***
2016	1.192	0.519	2.606	2.111	1.287	4.401	-0.919***	-0.770***
2017	1.185	0.563	3.370	2.306	1.397	6.161	-1.121***	-0.835***
2018	1.047	0.437	2.540	2.452	1.434	6.913	-1.405***	-0.998***
2019	2.223	0.661	10.205	3.238	1.494	11.689	-1.015***	-0.833***
ALL	1.026	0.407	3.685	1.890	1.114	5.217	-0.865***	-0.707***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the RoBERTa prediction model and the Stack Ensemble model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.28: Comparing Scale-Independent Absolute Forecast Error (SAFE) for Stack Ensemble with the bag-of-words model of LoughranMcDonald

	RoBERTa			Stack Ensemble			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.439	0.194	1.377	0.915	0.530	3.005	-0.476***	-0.337***
2005	0.545	0.229	1.568	0.908	0.585	2.511	-0.363***	-0.354***
2006	0.586	0.222	1.424	0.928	0.660	1.660	-0.343***	-0.438***
2007	0.528	0.226	1.176	0.917	0.620	1.230	-0.389***	-0.395***
2008	0.754	0.350	1.358	0.982	0.702	1.187	-0.228***	-0.353***
2009	0.577	0.283	0.943	0.920	0.592	1.290	-0.343***	-0.309***
2010	0.475	0.228	1.172	0.898	0.567	1.544	-0.423***	-0.339***
2011	0.431	0.183	1.199	0.879	0.554	1.553	-0.448***	-0.372***
2012	0.379	0.179	0.831	0.861	0.548	1.509	-0.482***	-0.369***
2013	0.398	0.167	1.053	0.850	0.547	1.521	-0.451***	-0.380***
2014	0.474	0.169	1.402	0.940	0.543	2.411	-0.466***	-0.374***
2015	0.399	0.179	0.936	0.871	0.575	1.731	-0.472***	-0.396***
2016	0.505	0.220	1.105	0.895	0.546	1.867	-0.390***	-0.326***
2017	0.448	0.213	1.275	0.873	0.529	2.331	-0.424***	-0.316***
2018	0.362	0.151	0.878	0.848	0.496	2.389	-0.486***	-0.345***
2019	0.614	0.183	2.820	0.895	0.413	3.230	-0.281***	-0.230***
ALL	0.485	0.200	1.385	0.895	0.550	2.072	-0.409***	-0.350***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the RoBERTa prediction model and the Stack Ensemble model. T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.29: Comparing Mean Absolute Error (MAE) for Stack Ensemble with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	Stack Ensemble			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.653	0.288	2.049	1.409	0.865	4.468	-0.756***	-0.578***
2005	0.790	0.332	2.270	1.345	0.847	3.628	-0.556***	-0.512***
2006	0.864	0.328	2.102	1.432	0.982	2.445	-0.568***	-0.654***
2007	0.879	0.376	1.957	1.670	1.118	2.108	-0.792***	-0.743***
2008	1.123	0.521	2.023	1.489	1.083	1.795	-0.366***	-0.562***
2009	0.839	0.411	1.371	1.397	0.929	1.898	-0.558***	-0.518***
2010	0.782	0.376	1.932	1.538	0.976	2.600	-0.756***	-0.601***
2011	0.754	0.320	2.097	1.635	1.081	2.771	-0.881***	-0.762***
2012	0.686	0.324	1.506	1.686	1.146	2.859	-1.000***	-0.822***
2013	0.766	0.321	2.025	1.796	1.247	3.050	-1.030***	-0.928***
2014	1.014	0.361	2.996	2.096	1.393	3.837	-1.082***	-1.033***
2015	0.830	0.372	1.946	2.046	1.392	3.694	-1.216***	-1.019***
2016	1.192	0.519	2.606	2.267	1.432	4.461	-1.076***	-0.913***
2017	1.185	0.563	3.370	2.470	1.525	6.234	-1.285***	-0.962***
2018	1.047	0.437	2.540	2.642	1.539	7.027	-1.595***	-1.102***
2019	2.223	0.661	10.205	3.413	1.679	11.707	-1.190***	-1.019***
ALL	1.026	0.407	3.685	2.017	1.245	5.174	-0.991***	-0.838***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the Stack Ensemble prediction model and the Forward-Looking bag-of-words model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.30: Comparing Scale-Independent Absolute Forecast Error (SAFE) for Stack Ensemble with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	Stack Ensemble			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.439	0.194	1.377	0.947	0.582	3.004	-0.508***	-0.388***
2005	0.545	0.229	1.568	0.929	0.585	2.505	-0.384***	-0.354***
2006	0.586	0.222	1.424	0.970	0.666	1.657	-0.385***	-0.443***
2007	0.528	0.226	1.176	1.004	0.672	1.267	-0.476***	-0.447***
2008	0.754	0.350	1.358	1.000	0.727	1.205	-0.246***	-0.377***
2009	0.577	0.283	0.943	0.961	0.639	1.305	-0.384***	-0.356***
2010	0.475	0.228	1.172	0.934	0.592	1.578	-0.459***	-0.364***
2011	0.431	0.183	1.199	0.935	0.618	1.584	-0.504***	-0.435***
2012	0.379	0.179	0.831	0.931	0.633	1.578	-0.552***	-0.454***
2013	0.398	0.167	1.053	0.934	0.648	1.586	-0.535***	-0.483***
2014	0.474	0.169	1.402	0.981	0.652	1.796	-0.506***	-0.483***
2015	0.399	0.179	0.936	0.984	0.669	1.777	-0.585***	-0.490***
2016	0.505	0.220	1.105	0.962	0.607	1.892	-0.456***	-0.388***
2017	0.448	0.213	1.275	0.935	0.577	2.359	-0.486***	-0.364***
2018	0.362	0.151	0.878	0.913	0.532	2.429	-0.551***	-0.381***
2019	0.614	0.183	2.820	0.943	0.464	3.234	-0.329***	-0.282***
ALL	0.485	0.200	1.385	0.953	0.607	2.044	-0.468***	-0.407***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the Stack Ensemble prediction model and the Forward-Looking bag-of-words model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.31: Comparing Mean Absolute Error (MAE) for LoughranMcDonald with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	LoughranMcDonald			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	1.361	0.788	4.469	1.409	0.865	4.468	-0.047	-0.077
2005	1.315	0.847	3.637	1.345	0.847	3.628	-0.030	-0.000
2006	1.370	0.974	2.450	1.432	0.982	2.445	-0.062	-0.008
2007	1.526	1.032	2.046	1.670	1.118	2.108	-0.144	-0.085
2008	1.463	1.046	1.768	1.489	1.083	1.795	-0.026	-0.036
2009	1.338	0.860	1.876	1.397	0.929	1.898	-0.059	-0.069*
2010	1.480	0.934	2.545	1.538	0.976	2.600	-0.059	-0.043
2011	1.537	0.970	2.717	1.635	1.081	2.771	-0.098	-0.111**
2012	1.560	0.993	2.734	1.686	1.146	2.859	-0.126	-0.153***
2013	1.634	1.052	2.925	1.796	1.247	3.050	-0.162	-0.197***
2014	2.009	1.160	5.154	2.096	1.393	3.837	-0.087	-0.234***
2015	1.811	1.196	3.598	2.046	1.392	3.694	-0.235**	-0.196***
2016	2.111	1.287	4.401	2.267	1.432	4.461	-0.157	-0.144***
2017	2.306	1.397	6.161	2.470	1.525	6.234	-0.164	-0.127**
2018	2.452	1.434	6.913	2.642	1.539	7.027	-0.190	-0.104**
2019	3.238	1.494	11.689	3.413	1.679	11.707	-0.175	-0.186***
ALL	1.890	1.114	5.217	2.017	1.245	5.174	-0.126***	-0.130***

Note: this table shows the differences in mean and median Mean Absolute Error (MAE) based on the LoughranMcDonald bag-of-words prediction model and the Forward-Looking bag-of-words prediction model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Table A.32: Comparing Scale-Independent Absolute Forecast Error (SAFE) for LoughranMcDonald with the Forward-Looking bag-of-words model of Muslu et al. (2015)

	LoughranMcDonald			Forward-Looking			Test of Difference	
	Mean	Median	s.d.	Mean	Median	s.d.	Mean	Median
2004	0.915	0.530	3.005	0.947	0.582	3.004	-0.032	-0.052
2005	0.908	0.585	2.511	0.929	0.585	2.505	-0.021	-0.000
2006	0.928	0.660	1.660	0.970	0.666	1.657	-0.042	-0.005
2007	0.917	0.620	1.230	1.004	0.672	1.267	-0.087	-0.051
2008	0.982	0.702	1.187	1.000	0.727	1.205	-0.018	-0.024
2009	0.920	0.592	1.290	0.961	0.639	1.305	-0.040	-0.047*
2010	0.898	0.567	1.544	0.934	0.592	1.578	-0.036	-0.026
2011	0.879	0.554	1.553	0.935	0.618	1.584	-0.056	-0.064**
2012	0.861	0.548	1.509	0.931	0.633	1.578	-0.070	-0.084***
2013	0.850	0.547	1.521	0.934	0.648	1.586	-0.084	-0.103***
2014	0.940	0.543	2.411	0.981	0.652	1.796	-0.041	-0.109***
2015	0.871	0.575	1.731	0.984	0.669	1.777	-0.113**	-0.094***
2016	0.895	0.546	1.867	0.962	0.607	1.892	-0.066	-0.061***
2017	0.873	0.529	2.331	0.935	0.577	2.359	-0.062	-0.048**
2018	0.848	0.496	2.389	0.913	0.532	2.429	-0.066	-0.036**
2019	0.895	0.413	3.230	0.943	0.464	3.234	-0.048	-0.051***
ALL	0.895	0.550	2.072	0.953	0.607	2.044	-0.058***	-0.057***

Note: this table shows the differences in mean and median Scale-Independent Absolute Forecast Error (SAFE) based on the LoughranMcDonald bag-of-words prediction model and the Forward-Looking bag-of-words prediction model of Muslu et al (2015). T-statistics based on robust standard errors corrected for heteroskedasticity for annual comparisons. T-statistics based on standard errors clustered by firms to correct for correlations within groups in the pooled regression. ***, **, and * correspond to 1 percent, 5 percent, and 10 percent significance levels, respectively (two-tailed).

Loss Firms

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
AR(1)	0.622	0.917	0.827	0.705	0.846	0.768	1.002	0.983	0.964	0.931	0.972	0.793	0.755	0.723	0.678	0.889	0.836
Stack Ensemble	0.622	0.961	0.763	0.649	0.835	0.805	0.788	0.656	0.61	0.731	0.764	0.642	0.713	0.653	0.674	0.89	0.735

Table A.33: Performance comparison for AR(1) and Stack for loss firms using SAFE measure to forecast earnings

Profit Firms

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
AR(1)	0.474	0.548	0.605	0.579	0.661	0.585	0.598	0.546	0.505	0.478	0.507	0.427	0.531	0.549	0.366	0.572	0.533
Stack Ensemble	0.48	0.566	0.623	0.564	0.684	0.52	0.493	0.453	0.406	0.419	0.488	0.397	0.514	0.494	0.36	0.577	0.502

Table A.34: Performance comparison for AR(1) and Stack for profit firms using SAFE measure to forecast earnings

No Non GAAP (mgr_exclude = 0) and Loss Firms

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
AR(1)	0.587	0.938	0.768	0.699	0.81	0.64	1.048	0.925	0.867	0.961	0.983	0.779	0.594	0.692	0.618	0.708	0.789
Stack Ensemble	0.601	0.972	0.72	0.649	0.794	0.691	0.76	0.62	0.563	0.709	0.771	0.624	0.582	0.618	0.611	0.745	0.689

Table A.35: Performance comparison for AR(1) and Stack for no Non-GAAP loss firms using SAFE measure to forecast earnings

Non GAAP (mgr_exclude = 1) and Loss Firms

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
AR(1)	0.529	0.967	0.885	0.875	1.054	0.815	0.887	1.084	0.791	1.024	0.99	0.708	0.856	0.804	0.765	0.979	0.876
Stack Ensemble	0.564	1.04	0.737	0.805	1.039	0.871	0.818	0.767	0.532	0.849	0.779	0.547	0.862	0.733	0.777	0.964	0.793

Table A.36: Performance comparison for AR(1) and Stack for Non-GAAP loss firms using SAFE measure to forecast earnings

No Non GAAP (mgr_exclude = 0) and Profit Firms

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
AR(1)	0.433	0.528	0.607	0.591	0.664	0.522	0.612	0.532	0.506	0.503	0.415	0.384	0.498	0.494	0.261	0.468	0.501
Stack Ensemble	0.438	0.553	0.617	0.58	0.685	0.459	0.509	0.443	0.403	0.466	0.389	0.408	0.491	0.469	0.28	0.499	0.480

Table A.37: Performance comparison for AR(1) and Stack for no Non-GAAP profit firms using SAFE measure to forecast earnings

Non GAAP (mgr_exclude = 1) and Profit Firms

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
AR(1)	0.578	0.521	0.605	0.573	0.678	0.655	0.615	0.597	0.532	0.488	0.548	0.459	0.557	0.567	0.406	0.595	0.561
Stack Ensemble	0.574	0.541	0.633	0.556	0.699	0.589	0.504	0.489	0.424	0.415	0.533	0.404	0.536	0.499	0.386	0.584	0.523

Table A.38: Performance comparison for AR(1) and Stack for Non-GAAP profit firms using SAFE measure to forecast earnings

Surviving Firms

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
AR(1)	0.447	0.478	0.527	0.585	0.638	0.607	0.553	0.507	0.477	0.481	0.438	0.384	0.497	0.532	0.301	0.529	0.499
Stack Ensemble	0.456	0.503	0.53	0.571	0.649	0.581	0.434	0.408	0.405	0.433	0.426	0.392	0.514	0.492	0.321	0.546	0.479

Table A.39: Performance comparison for AR(1) and Stack for surviving firms using SAFE measure to forecast earnings

Non-surviving firms

Year	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	Average
AR(1)	0.431	0.569	0.622	0.542	0.797	0.628	0.611	0.567	0.512	0.472	0.555	0.486	0.545	0.503	0.399	0.655	0.556
Stack Ensemble	0.432	0.584	0.633	0.519	0.818	0.589	0.507	0.452	0.375	0.393	0.499	0.414	0.513	0.449	0.386	0.653	0.513

Table A.40: Performance comparison for AR(1) and Stack for Non-surviving firms using SAFE measure to forecast earnings

References

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

S. Agrawal and J. Agrawal. Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60:708–713, 2015.

C. Alberti, K. Lee, and M. Collins. A bert baseline for the natural questions. *arXiv preprint arXiv:1901.08634*, 2019.

W. S. Albrecht, L. L. Lookabill, and J. C. McKeown. The time-series properties of annual earnings. *Journal of Accounting Research*, pages 226–244, 1977.

F. A. Amani and A. M. Fadlalla. Data mining applications in accounting: A review of the literature and organizing framework. *International Journal of Accounting Information Systems*, 24:32–58, 2017.

A. Argyrou. Auditing journal entries using self-organizing map. In *AMCIS*, 2012.

S. T. Aroyehun and A. Gelbukh. Aggression detection in social media: Using deep neural networks, data augmentation, and pseudo labeling. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 90–97, 2018.

R. Ball and R. Watts. Some time series properties of accounting income. *The Journal of Finance*, 27(3):663–681, 1972.

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

O. E. Barron, M. H. Stanford, and Y. Yu. Further evidence on the relation between analysts' forecast dispersion and stock returns. *Contemporary Accounting Research*, 26(2):329–357, 2009.

S. Bay, K. Kumaraswamy, M. G. Anderle, R. Kumar, and D. M. Steier. Large scale detection of irregularities in accounting data. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 75–86. IEEE, 2006.

R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

I. Beltagy, K. Lo, and A. Cohan. Scibert: A pretrained language model for scientific text. *arXiv preprint arXiv:1903.10676*, 2019.

J. Berns, P. Bick, R. Flugum, and R. Houston. Do changes in md&a section tone predict investment behavior? *Financial Review*, 2021a.

J. Berns, P. Bick, R. Flugum, and R. Houston. Do changes in md&a section tone predict investment behavior? *The Financial Review*, 2021b.

J. Bertomeu. Machine learning improves accounting: discussion, implementation and research opportunities. *Review of Accounting Studies*, 25(3):1135–1155, 2020.

J. Bertomeu and I. Marinovic. A theory of hard and soft information. *The Accounting Review*, 91(1):1–20, 2016.

J. Bertomeu, E. Cheynel, and D. Cianciaruso. Strategic withholding and imprecision in asset measurement. *Journal of Accounting Research*, 2019.

J. Bertomeu, I. Vaysman, and W. Xue. Voluntary versus mandatory disclosure. *Review of Accounting Studies*, 26(2):658–692, 2021.

I. Bhattacharya and E. R. Lindgreen. A semi-supervised machine learning approach to detect anomalies in big accounting data. In *ECIS*, 2020.

O. Binz, K. Schipper, and K. Standridge. What can analysts learn from artificial intelligence about fundamental analysis? *Available at SSRN 3745078*, 2020.

D. Birant and A. Kut. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering*, 60(1):208–221, 2007.

R. J. Bolton, D. J. Hand, et al. Unsupervised profiling methods for fraud detection. *Credit scoring and credit control VII*, pages 235–255, 2001.

R. M. Bowen, A. K. Davis, and D. A. Matsumoto. Do conference calls affect analysts' forecasts? *The accounting review*, 77(2):285–316, 2002.

S. Bozinovski. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44(3), 2020.

L. Breiman. Arcing the edge. Technical report, Technical Report 486, Statistics Department, University of California at . . . , 1997.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

P. L. Brockett, X. Xia, and R. A. Derrig. Using kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. *Journal of Risk and Insurance*, pages 245–274, 1998.

L. D. Brown. Earnings forecasting research: its implications for capital markets research. *International journal of forecasting*, 9(3):295–320, 1993.

L. D. Brown. A temporal analysis of earnings surprises: Profits versus losses. *Journal of accounting research*, 39(2):221–241, 2001.

N. C. Brown, R. M. Crowley, and W. B. Elliott. What are you saying? using topic to detect financial misreporting. *Journal of Accounting Research*, 58(1):237–291, 2020.

S. Brown, S. A. Hillegeist, and K. Lo. Conference calls and information asymmetry. *Journal of Accounting and Economics*, 37(3):343–366, 2004.

S. H. Bryan. Incremental information content of required disclosures contained in management discussion and analysis. *Accounting Review*, pages 285–301, 1997.

B. J. Bushee, I. D. Gow, and D. J. Taylor. Linguistic complexity in firm disclosures: Obfuscation or information? *Journal of Accounting Research*, 56(1):85–121, 2018.

R. M. Bushman, A. Lerman, and X. F. Zhang. The changing landscape of accrual accounting. *Journal of Accounting Research*, 54(1):41–78, 2016.

K. Cao and H. You. Fundamental analysis via machine learning. *Available at SSRN 3706532*, 2020.

M. Cecchini, H. Aytug, G. J. Koehler, and P. Pathak. Detecting management fraud in public companies. *Management Science*, 56(7):1146–1160, 2010a.

M. Cecchini, H. Aytug, G. J. Koehler, and P. Pathak. Making words work: Using financial text as a predictor of financial events. *Decision Support Systems*, 50(1):164–175, 2010b.

M. Çelik, F. Dadaşer-Çelik, and A. Ş. Dokuz. Anomaly detection in temperature data using dbscan algorithm. In *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pages 91–95. IEEE, 2011.

A. Ceresney. Sec.gov — financial reporting and accounting fraud. 2013. URL `https://www.sec.gov/news/speech/spch091913ac`. (Accessed on 05/11/2021).

V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

C.-I. Chang and S.-S. Chiang. Anomaly detection and classification for hyperspectral imagery. *IEEE transactions on geoscience and remote sensing*, 40(6):1314–1325, 2002.

O. Chapelle, B. Scholkopf, and A. Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

S. Chen, B. Miao, and T. Shevlin. A new measure of disclosure quality: The level of disaggregation of accounting data in annual reports. *Journal of Accounting Research*, 53 (5):1017–1054, 2015.

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

F. S. Cheong and J. Thomas. Why do eps forecast error and dispersion not vary with scale? implications for analyst and managerial behavior. *Journal of Accounting Research*, 49(2): 359–401, 2011.

F. S. Cheong and J. Thomas. Management of reported and forecast eps, investor responses, and research implications. *Management Science*, 64(9):4277–4301, 2018.

M. B. Clement and S. Y. Tse. Do investors respond to analysts' forecast revisions as if forecast accuracy is all that matters? *The Accounting Review*, 78(1):227–249, 2003.

P. Craja, A. Kim, and S. Lessmann. Deep learning for detecting financial statement fraud. *Decision Support Systems*, 139:113421, 2020.

P. M. Dechow, W. Ge, C. R. Larson, and R. G. Sloan. Predicting material accounting misstatements. *Contemporary accounting research*, 28(1):17–82, 2011.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. *Proceedings of the 2019 Conference of the North*, 2019. doi: 10.18653/v1/n19-1423. URL `http://dx.doi.org/10.18653/v1/N19-1423`.

I. D. Dichev and V. W. Tang. Matching and the changing properties of accounting earnings over the last 40 years. *The Accounting Review*, 83(6):1425–1460, 2008.

G. Dickey, S. Blanke, and L. Seaton. Machine learning in auditing. *The CPA Journal*, pages 16–21, 2019.

G. Ding, S. Zhang, S. Khan, Z. Tang, J. Zhang, and F. Porikli. Feature affinity based pseudo labeling for semi-supervised person re-identification. *IEEE Transactions on Multimedia*, 2019.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification.* John Wiley & Sons, 2012.

A. Durnev and C. Mangen. The spillover effects of md&a disclosures for real investment: The role of industry competition. *Journal of Accounting and Economics*, 70(1):101299, 2020.

A. Dyck, A. Morse, and L. Zingales. Who blows the whistle on corporate fraud? *The journal of finance*, 65(6):2213–2253, 2010.

I. Dyck, A. Morse, and L. Zingales. How pervasive is corporate fraud? *Rotman School of Management Working Paper*, (2222608), 2013.

P. D. Easton, M. Kapons, P. Kelly, and A. Neuhierl. Attrition bias and inferences regarding earnings properties; evidence from compustat data. *Available at SSRN 3040354*, 2020a.

P. D. Easton, M. M. Kapons, S. J. Monahan, H. H. Schütt, and E. H. Weisbrod. Forecasting earnings using k-nearest neighbor matching. 2020b.

L. Ertoz, E. Eilertson, A. Lazarevic, P.-N. Tan, V. Kumar, J. Srivastava, and P. Dokas. Minds-minnesota intrusion detection system. *Next generation data mining*, pages 199–218, 2004.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

R. Feldman, S. Govindaraj, J. Livnat, and B. Segal. Management's tone change, post earnings announcement drift and accruals. *Review of Accounting Studies*, 15(4):915–953, 2010.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1-2):18–28, 2009.

D. Givoly and G. C. Biddle. Financial analysts and their contribution to well-functioning capital markets, 2018.

D. Givoly and C. Hayn. The changing time-series properties of earnings, cash flows and accruals: Has financial reporting become more conservative? *Journal of accounting and economics*, 29(3):287–320, 2000.

S. Goel and J. Gangolly. Beyond the numbers: Mining the annual reports for hidden cues indicative of financial statement fraud. *Intelligent Systems in Accounting, Finance and Management*, 19(2):75–89, 2012.

S. Goel, J. Gangolly, S. R. Faerman, and O. Uzuner. Can linguistic predictors detect fraudulent financial filings? *Journal of Emerging Technologies in Accounting*, 7(1):25–46, 2010.

M. Goldstein and S. Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, 46:235–262, 2013.

J. Green, H. Louis, and J. Sani. Intangible investments, scaling, and the trend in the accrual–cash flow association. *Journal of Accounting Research*, 2021.

S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. *Information systems*, 25(5):345–366, 2000.

D. M. Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

M. Hendriock. Forecasting earnings with predicted, conditional probability density functions. *Conditional Probability Density Functions (August 8, 2021)*, 2021.

G. Hoberg and C. Lewis. Do fraudulent firms produce abnormal disclosure? *Journal of Corporate Finance*, 43:58–85, 2017.

L. Holder-Webb and J. R. Cohen. The association between disclosure, distress, and failure. *Journal of Business Ethics*, 75(3):301–314, 2007.

O.-K. Hope. Disclosure practices, enforcement of accounting standards, and analysts' forecast accuracy: An international study. *Journal of accounting research*, 41(2):235–272, 2003.

K. Hou, M. A. Van Dijk, and Y. Zhang. The implied cost of capital: A new approach. *Journal of Accounting and Economics*, 53(3):504–526, 2012.

S. L. Humpherys, K. C. Moffitt, M. B. Burns, J. K. Burgoon, and W. F. Felix. Identification of fraudulent financial statements using linguistic credibility analysis. *Decision Support Systems*, 50(3):585–594, 2011.

L.-S. Hwang, C.-L. Jan, and S. Basu. Loss firms and analysts' earnings forecast errors. *The Journal of Financial Statement Analysis*, 1(2), 1996.

R. J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

R. J. Hyndman and A. B. Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

A. K. Jain and R. C. Dubes. Algorithms for clustering data. *Englewood Cliffs: Prentice Hall, 1988*, 1988.

M. Jans, N. Lybaert, and K. Vanhoof. Internal fraud risk reduction: Results of a data mining case study. *International Journal of Accounting Information Systems*, 11(1):17–41, 2010.

J. M. Karpoff, A. Koester, D. S. Lee, and G. S. Martin. Proxies and databases in financial misconduct research. *The Accounting Review*, 92(6):129–163, 2017.

G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154, 2017.

R. Khan, M. Corney, A. Clark, and G. Mohay. Transaction mining for fraud detection in erp systems. *Industrial Engineering and Management Systems*, 9(2):141–156, 2010.

W. Kim. Parallel clustering algorithms: survey. *Parallel Algorithms, Spring*, 34:43, 2009.

A. Klein and C. A. Marquardt. Fundamentals of accounting losses. *The Accounting Review*, 81(1):179–206, 2006.

R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

T. Kohonen. Exploration of very large databases by self-organizing maps. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, volume 1, pages PL1–PL6. IEEE, 1997.

S. Kothari. Capital markets research in accounting. *Journal of accounting and economics*, 31(1-3):105–231, 2001.

S. P. Kothari, S. Shu, and P. D. Wysocki. Do managers withhold bad news? *Journal of Accounting research*, 47(1):241–276, 2009.

D. F. Larcker and A. A. Zakolyukina. Detecting deceptive discussions in conference calls. *Journal of Accounting Research*, 50(2):495–540, 2012.

E. E. LeDell. *Scalable ensemble learning and computationally efficient variance estimation.* University of California, Berkeley, 2015.

D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36 (4):1234–1240, 2020.

G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL `http://jmlr.org/papers/v18/16-365`.

E. Leung and D. Veenman. Non-gaap earnings disclosure in loss firms. *Journal of Accounting Research*, 56(4):1083–1137, 2018.

B. Lev and F. Gu. *The end of accounting and the path forward for investors and managers.* John Wiley & Sons, 2016.

F. Li. Annual report readability, current earnings, and earnings persistence. *Journal of Accounting and economics*, 45(2-3):221–247, 2008.

K. K. Li. How well do investors understand loss persistence? *Review of Accounting Studies*, 16(3):630–667, 2011.

K. K. Li and P. Mohanram. Evaluating cross-sectional forecasting models for implied cost of capital. *Review of Accounting Studies*, 19(3):1152–1185, 2014.

J. Lin, E. Keogh, A. Fu, and H. Van Herle. Approximations to magic: Finding unusual medical time series. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 329–334. Citeseer, 2005.

M. Liu. Assessing human information processing in lending decisions: A machine learning approach. *Journal of Accounting Research*, 60(2):607–651, 2022.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

T. Loughran and B. McDonald. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. *The Journal of Finance*, 66(1):35–65, 2011.

T. Loughran and B. McDonald. Textual analysis in accounting and finance: A survey. *Journal of Accounting Research*, 54(4):1187–1230, 2016.

M. Michailidis. *Investigating machine learning methods in recommender systems.* PhD thesis, UCL (University College London), 2017.

S. Minhas and A. Hussain. From spin to swindle: identifying falsification in financial text. *Cognitive computation*, 8(4):729–745, 2016.

K. Moffit, M. Burns, W. Felix, and J. Burgoon. Using lexical bundles to discriminate between fraudulent and non-fraudulent financial reports on. In *SIG-ASYS Pre-ICIS 2010 workshop*, 2010.

S. Monahan. Financial statement analysis and earnings forecasting. *Foundations and Trends® in Accounting*, 12(2):105–215, 2018.

V. Muslu, S. Radhakrishnan, K. Subramanyam, and D. Lim. Forward-looking md&a disclosures and the information environment. *Management Science*, 61(5):931–948, 2015.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

J. L. Perols, R. M. Bowen, C. Zimmermann, and B. Samba. Finding needles in a haystack: Using data analytics to improve fraud prediction. *The Accounting Review*, 92(2):221–245, 2017.

F. Petropoulos, D. Apiletti, V. Assimakopoulos, M. Z. Babai, D. K. Barrow, S. B. Taieb, C. Bergmeir, R. J. Bessa, J. Bijak, J. E. Boylan, et al. Forecasting: theory and practice. *arXiv preprint arXiv:2012.03854*, 2020.

T. Pires, E. Schlinger, and D. Garrette. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*, 2019.

L. Portnoy. *Intrusion detection with unlabeled data using clustering.* PhD thesis, Columbia University, 2000.

G. A. Pratt. Is a cambrian explosion coming for robotics? *Journal of Economic Perspectives*, 29(3):51–60, 2015.

L. Purda and D. Skillicorn. Accounting variables, deception, and a bag of words: Assessing the tools of fraud detection. *Contemporary Accounting Research*, 32(3):1193–1223, 2015.

N. Rahmah and I. Sitanggang. Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra. *IOP Conference Series: Earth and Environmental Science*, 31:012012, 01 2016. doi: 10.1088/1755-1315/31/1/012012.

M. Ramadas, S. Ostermann, and B. Tjaden. Detecting anomalous network traffic with self-organizing maps. In *International Workshop on Recent Advances in Intrusion Detection*, pages 36–54. Springer, 2003.

S. Ramnath, S. Rock, and P. B. Shane. Financial analysts' forecasts and stock recommendations: A review of the research. 2008.

R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

J. L. Rogers, A. Van Buskirk, and S. L. Zechman. Disclosure tone and shareholder litigation. *The Accounting Review*, 86(6):2155–2183, 2011.

G. Rogova. Combining the results of several neural network classifiers. *Neural Networks*, 7 (5):777–781, 1994. ISSN 0893-6080. doi: https://doi.org/10.1016/0893-6080(94)90099-X. URL `https://www.sciencedirect.com/science/article/pii/089360809490099X`.

M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer. Detection of anomalies in large scale accounting data using deep autoencoder networks. *arXiv preprint arXiv:1709.05254*, 2017.

A. J. C. Sharkey. On combining artificial neural nets. *Connection science*, 8(3-4):299–314, 1996.

H. Shi. *Best-first decision tree learning.* PhD thesis, The University of Waikato, 2007.

F. Siano and P. Wysocki. Transfer learning and textual analysis of accounting disclosures: Applying big data methods to small (er) datasets. *Accounting Horizons*, 35(3):217–244, 2021.

N. V. Smirnov. Estimate of deviation between empirical distribution functions in two independent samples. *Bulletin Moscow University*, 2(2):3–16, 1939.

R. Smith, A. Bivens, M. Embrechts, C. Palagiri, and B. Szymanski. Clustering approaches for anomaly based intrusion detection. *Proceedings of intelligent engineering systems through artificial neural networks*, 9, 2002.

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

E. C. So. A new approach to predicting analyst forecast errors: Do investors overweight analyst forecasts? *Journal of Financial Economics*, 108(3):615–640, 2013.

A. Srivastava. Why have measures of earnings quality changed over time? *Journal of Accounting and Economics*, 57(2-3):196–217, 2014.

C. Sun, X. Qiu, Y. Xu, and X. Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

X. Sun, M. Liu, and Z. Sima. A novel cryptocurrency price trend forecasting model based on lightgbm. *Finance Research Letters*, 2018.

J. A. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

P.-N. Tan. *Introduction to data mining.* Pearson Education India, 2018.

S. Thiprungsri and M. A. Vasarhelyi. Cluster analysis for anomaly detection in accounting data: An audit approach. *International Journal of Digital Accounting Research*, 11, 2011.

M. Van den Bogaerd and W. Aerts. Applying machine learning in accounting research. *Expert Systems with Applications*, 38(10):13414–13424, 2011.

T. Versano. Silence can be golden: On the value of allowing managers to keep silent when information is soft. *Journal of Accounting and Economics*, 71(2-3):101399, 2021.

J. Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.

D. Wang, Y. Zhang, and Y. Zhao. Lightgbm: an effective mirna classification method in breast cancer patients. In *Proceedings of the 2017 International Conference on Computational Biology and Bioinformatics*, pages 7–11. ACM, 2017.

R. L. Watts and R. W. Leftwich. The time series of annual accounting earnings. *Journal of Accounting Research*, pages 253–271, 1977.

D. H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

W. Yang, Y. Xie, A. Lin, X. Li, L. Tan, K. Xiong, M. Li, and J. Lin. End-to-end open-domain question answering with bertserini. *arXiv preprint arXiv:1902.01718*, 2019.

J. Zhu, Y. Xia, L. Wu, D. He, T. Qin, W. Zhou, H. Li, and T.-Y. Liu. Incorporating bert into neural machine translation. *arXiv preprint arXiv:2002.06823*, 2020.

X. J. Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

English Summary

Auditing standards require quantitative investigations of data to identify the risk of material misstatements to produce accurate financial reports. Chapter 2 introduces a novel method that detects anomalous observations in large-scale accounting data and helps auditors to detect suspicious observations in big accounting data for follow-up investigations. On the other hand, we have also found how companies and large institutions delve into fraudulent activities by deliberately misguiding the shareholders. Chapter 3 introduces an economically significant method that contextually learns from the business texts to infer the likelihood of accounting fraud. Shareholders and other stakeholders tend to analyze the financial reports to estimate how a company is going to perform in the future and that makes forecasting the earnings of these companies an important subject to study. Chapter 4 introduces a method that combines both structured and unstructured data to produce more accurate earnings forecasting and improve the prediction for firms that are essentially hard to predict.

In this thesis, I introduce state-of-the-art machine learning algorithms into the accounting field to produce cost and time-effective solutions for practical problems. All three chapters explore the optimization of combining the domain expertise from the accounting field and the best practices from the machine learning literature. I use both structured and unstructured data to obtain the findings and Python as the programming language to obtain the results.

Dutch Summary

Controlestandaarden vereisen kwantitatief onderzoek van gegevens om het risico van materiële afwijkingen te identificeren om nauwkeurige financiële rapporten te produceren. Hoofdstuk 2 introduceert een nieuwe methode die afwijkende waarnemingen in grootschalige boekhoudgegevens detecteert en auditors helpt om verdachte waarnemingen in grote boekhoudgegevens te detecteren voor vervolgonderzoeken. Aan de andere kant hebben we ook ontdekt hoe bedrijven en grote instellingen frauduleuze activiteiten ontduiken door de aandeelhouders bewust op het verkeerde been te zetten. Hoofdstuk 3 introduceert een economisch significante methode die contextueel leert van de zakelijke teksten om de waarschijnlijkheid van boekhoudfraude af te leiden. Aandeelhouders en andere belanghebbenden hebben de neiging om de financiële rapporten te analyseren om in te schatten hoe een bedrijf in de toekomst gaat presteren en dat maakt het voorspellen van de winst van deze bedrijven een belangrijk onderwerp om te bestuderen. Hoofdstuk 4 introduceert een methode die zowel gestructureerde als ongestructureerde gegevens combineert om nauwkeurigere winstprognoses te produceren en de voorspelling te verbeteren voor bedrijven die in wezen moeilijk te voorspellen zijn.

In dit proefschrift introduceer ik state-of-the-art algoritmen voor machine learning in het boekhoudveld om kosten- en tijdbesparende oplossingen voor praktische problemen te produceren. Alle drie de hoofdstukken onderzoeken de optimalisatie van het combineren van de domeinexpertise uit het accountingveld en de best practices uit de machine learning-literatuur. Ik gebruik zowel gestructureerde als ongestructureerde data om de bevindingen te verkrijgen en Python als programmeertaal om de resultaten te verkrijgen.

B Acknowledgements

Now that I have almost reached the end of my Ph.D. journey, as I look back, I am filled with a deep sense of gratitude and joy. This experience has been truly wonderful and fulfilling. In this section, I would like to express my heartfelt appreciation to the people who have made this journey truly remarkable. Without their support, encouragement, and guidance, my Ph.D. experience would have been vastly different.

Dear Edo, I want to dedicate the opening words to you. Without you, this journey would never have begun. I still remember our first interview when I was in India and the power went out and I panicked. But you made me feel comfortable and positive. Since then, you've been a huge support for me. Whether we talked about research or personal matters, your presence always gave me a great sense of comfort. You gave me the freedom to conduct my research independently. Whenever I felt down, you had a unique way of lifting my spirits, whether through anecdotes or life lessons. Your unconventional approach never failed to cheer me up. Most importantly, you truly understood me. You knew exactly when I needed to take things slow and when it was time to push myself further. Your understanding of my capabilities and limitations has been incredibly valuable. The only thing I regret is not getting the chance to ever pay for the coffee or lunch. Next time, I'll make sure to treat you so I can finally get rid of this regret.

I would like to show my gratitude to my co-promoter Prof. Dr. Jan Bouwens. You have always been highly supportive of me. I would never forget the support you provided during the toughest days of this journey and I am sincerely grateful for that. Moreover, I deeply value the life lessons we have shared and the insightful discussions we have had about career choices. Your guidance and perspectives have been invaluable to me.

I am thankful to each and every member of our department. I also want to extend my thanks to the participants of the brown bag sessions for the encouragement and the valuable discussions we had. My heartfelt thanks to Sophia De Jong, for your relentless effort in carrying out all administrative work. I am especially thankful to David Veenman, for not only the insightful research discussions that we had but also for the Ajax game ticket which was an amazing experience.

My Ph.D. journey was colorful because of the amazing colleagues I had. I want to convey my appreciation to Mate, Ejona, Pouyan, and Nan for the wonderful times we shared together. Razvan, our discussions about movies and online gaming were always fascinating. Jort and Reka, you've been the kind of friends with whom I could talk about anything in life for hours. Sandra and Anil, words cannot express how much our friendship means to me. From sharing the same office to sharing every update in our lives, our bond has grown stronger over time. Ioan, I trust you to keep our secrets forever, and if I ever need an energy boost, I know I can count on Tjibbe and Casper. Matijn, I'll miss our football chats, and I promise not to mention Lukas Moura in front of you ever again. Jenna - I would definitely need your advice in the future when I open my own restaurant.

I want to express my heartfelt appreciation to two of my coauthors, Sanjay and Ana. Working with you both has been an absolute pleasure. I still remember how we achieved so much together in such a short period of time. It wasn't just about learning and collaborating, it was also a thrilling experience. Our relationship extended beyond work and turned into a true friendship, where I felt completely at ease discussing any aspect of life. I genuinely look forward to future collaborations with both of you, while having a cup of exotic coffee.

They say it is not easy for someone to leave their homeland and spend several years in a foreign country. You miss your family and there is always uncertainty about when you will see them again. Thankfully, I have been fortunate enough to have a big family here in Europe, which has made this challenge much easier for me. Arkajyotida, Basundharadi, Mayukh, Arnab, Manna, Mamba, Aliva, Deepanda, Tulipdi, Shashwat, Rudrada, Tamalda, Rajarshida, Aditi, Saswatidi, Sagnikda, Ritamda, and Chinmoyda - I want to express my sincere appreciation for the friendship we share. Special thanks to Saagnik for designing the cover. Here is a special shout-out to all the fun animal parties we have had and will continue to have in the future.

I would now like to remember my days at the Indian Statistical Institute, Kolkata. I thank my professors for introducing me to the amazing potential of Applied Mathematics, Statistics, and Computer Science. Of course, I can not forget to thank my batch mates: Abishanka, Amlan, Ankit, Apoorv, Arnab, Atanu, Chutku, Diganta, Debarya, Indrayudh, Irani, Monojit, Po, Prosenjit, Richick, and Saroni. I would gladly revisit my life with you during 2010-2015 and stay there forever if I could.

I have been lucky to have some incredible friends from my school days and my neighborhood. I want to give a special mention to Souro, Somuda, Ayan, Indu, Bhuia, Krisam, Naru, Vivek, Knachu, Swapno, Sohail, Debanjan, and Anirban. The football matches we played together, regardless of the quality, will always hold a special place in my heart, and I hope we continue playing for years to come. Those days had a profound impact on shaping my character which I have always deeply appreciated.

Dear Tushar Sir, I guess I never had a chance before to say how much thankful I am to you. You were my friend, philosopher, and guide throughout my childhood. You taught me science from when I was 9 years old, until 16. I would have never believed what I am capable of unless you had shown me. I feel blessed to have you as my teacher for all the subjects and you made me believe that there is no boundary when it comes to learning. I feel that where I stand now is majorly contributed by you and I have never found a second Tushar Sir.

Well, now, let's talk about my family. Where do I even begin? I am blessed with a large and loving family. Bordida, you have always believed in me, and get ready, because I am coming soon to annoy you even more. Bordadu, I know you would have been incredibly proud of me today. I wish I could sit with you and share every detail of my Ph.D. journey. I know you would have shared those stories with other kids and people around you to inspire them as well. Baromama and Chotomama, you were my role models as I was growing up. Let's plan to jam together, watch a football match, and play some card games to relive the good old times. Mani, Manti, Q-Bhai, Moshai, and Baponmoni, your overwhelming love and support have never gone unnoticed. I know that if I ever have another exam knocking on my door, all of you would come the night before, and we would celebrate the end of the exam again, just like before.

Pisimoni, Prosunda, Bobokaka, Tarunkaka, Totokaka, Somakaka, Kakima, Budoda, Dolidi, Tapu, Gopa, and Babaikaka - I am at a loss for words to express my gratitude to you all. Whenever I feel down, I draw strength from your resilience and perseverance. It fills me with a tremendous amount of energy. Your lives have profoundly influenced my philosophical outlook, and I am proud of that. I want you to know that I would be a completely different person if you were not a part of my life.

Didi, Buban, and Bapanda - I would make sure you read this paragraph. I am remembering our childhood days, and what an incredible journey it was. Despite our silly fights, the way we also protected and stood up for each other is truly remarkable. Your endurance to fight and come out victorious, even when no one else believed in you, is a huge inspiration to me.

Baba, Maa - I want you to know that saying thank you will never be enough to express the impact you have had on my life. Baba, I continue to learn from you, and you remain my role model for how to navigate through life. Maa, your constant motivation and encouragement mean everything to me. You are one of the smartest people I have known in my life and I would consider myself truly successful if I can carry on your exceptional qualities of always lending a helping hand to people around. You made many sacrifices to ensure that Didi, Buban, and I have successful lives. I hope you feel proud of yourself now and take some time to relax. Keep blessing me always.

Last but not least, I want to express all my love to my wife, Jayeeta. I have already learned so much from you, and yet I know there is still much more to learn. I understand how challenging and stressful it can be when both of us are working on our theses. However, I am in awe of how calm and composed you remain throughout it all. I am genuinely proud to have you by my side. You inspire me every single day. When you are with me, I feel just like the saying goes, “Gar firdaus bar-rue zamin ast, hami asto, hamin asto, hamin ast.”

I would like to offer my sincere apologies and many thanks to those who I have missed in this paragraph.