# LongFinBERT: A Language Model for Very Long Financial Documents*

Tri Minh Phan,[a] Erik-Jan Senn[a]

Version: August 14, 2024

**Abstract**

This paper introduces LongFinBERT, a modern language model specialized for very long financial documents. LongFinBERT shows remarkable efficiency, requiring significantly less computational time and memory usage than other state-of-the-art language models. This unique characteristic enables LongFinBERT to process entire accounting and financial documents, which is typically infeasible with other language models. Leveraging this advantage, we apply LongFinBERT to detect financial misreporting using 10-K filings from January 1999 to December 2018. Misreporting detection utilizing text-based features from LongFinBERT outperforms that of canonical accounting variables or other text models. Finally, investors are responsive to year-over-year alterations of accounting disclosures, measured by LongFinBERT.

*Keywords:* 10-K filings, deep learning, language models, machine learning, misreporting detection
*JEL classification:* C53, G14, M41

*We thank participants of the 2023 Conference of Computational and Financial Econometrics (CFE 2023), the 2023 GPEF seminar at the University of St. Gallen and Francesco Audrino, Simon Feistle, Matthias Fengler, Lyudmila Grigoryeva, and Jule Schuettler for their valuable feedback.

*a*: Department of Economics, University of St.Gallen.

Corresponding author: Tri Minh Phan, Department of Economics, University of St.Gallen, Switzerland, Bodanstrasse 6, 9000 St. Gallen, Email: triminh.phan@unisg.ch.

# 1 Introduction

Current accounting and financial research has rapidly adopted natural language processing (NLP) techniques to decrypt the complexities of financial documents (Loughran and McDonald, 2016; Gentzkow et al., 2019; Ash and Hansen, 2023; Bae et al., 2023; Bochkay et al., 2023). However, most studies are still based on classical NLP methods, e.g., word counting, despite recent advancements in NLP, notably language models (LMs). One prominent barrier is the intense computational demand of LMs (Huang et al., 2023), especially when handling long documents, which are prevalent in accounting and financial research (Cazier and Pfeiffer, 2016; Dyer et al., 2017; SEC, 2003). This gap in the literature is noteworthy given that accounting documents have not been extensively explored using powerful tools like LMs.[1]

Attempting to bridge the gap, this paper introduces LongFinBERT, an LM that efficiently handles lengthy financial documents. Different from other traditional LMs such as BERT (Devlin et al., 2019) or FinBERT (Huang et al., 2023), LongFinBERT is constructed based on the innovative attention mechanism of Ding et al. (2023) called *LongNet*. It facilitates connections between words separated by significant distances in a lengthy document while pruning off unnecessary interactions between unrelated words. This unique mechanism allows LongFinBERT to harmoniously balance the computational efficiency and the linguistic intelligence inherent in LMs.

The construction of LongFinBERT is the first focus of this paper. Besides incorporating

[1] We distinguish the concepts of language models (LMs) from large language models (LLMs), acknowledging that several authors equate their definitions, e.g., Huang et al. (2023). In particular, we define LLMs as generative LMs whose number of parameters exceeds 1 billion. The motivation for our differentiation is that LLMs are, as the name suggests, generally large, hence necessitating special techniques for efficient training and fine-tuning such as quantization (Jacob et al., 2018) or low-rank adaptation (Hu et al., 2022). Furthermore, many current LLMs are not entirely published or are accessible at a high-level interface, hindering their potential applications in research. According to our definitions, LLMs include OpenAI's GPT model series; Google's PaLM, Gemini, and Gemma; xAI's Grok; Meta's LLaMA model series; Anthropic's Claude models; Mistral AI's model series; and Databricks' DBRX. LMs then include BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021), etc.

LongNet, we allow LongFinBERT to inherit the architectural and linguistic foundations of FinBERT (Huang et al., 2023), a famous LM customized for financial contexts. LongFinBERT is pre-trained with a large set of 10-K filings of US firms from January 1994 to December 2008, enabling it to grasp the language style used in financial and accounting contexts.[2] Importantly, thanks to the novel attention mechanism, LongFinBERT can process an entire 10-K filing in a single pass, distinguishing it from FinBERT, which analyzes only selected sections. Consequently, different from other LMs, LongFinBERT captures nuanced relationships within 10-K filings, whose sections are typically interconnected.[3] Following the introduction of LongFinBERT, this paper provides empirical evidence consolidating its efficiency in computation and financial applications by addressing two following questions: (i) *Does LongFinBERT exhibit superior computational efficiency compared to other LMs?* and (ii) *How does LongFinBERT perform in accounting and finance tasks?*

While Ding et al. (2023) provide the theoretical proof of LongNet's computational efficiency, the practical performance of LongFinBERT, leveraging LongNet at its core, within real-life accounting and financial documents remains unanswered, especially in comparison to other LMs designed for lengthy documents. Aiming to answer this question, we conduct a simulation study to compare the computational demands of LongFinBERT with those of four other LMs chosen for their proficiency in handling lengthy documents. We show that LongFinBERT requires significantly less time and memory usage than the competing LMs during the training and prediction phases. More importantly, LongFinBERT exhibits the unique capability of processing extremely long documents as a whole, a feature that is infeasible for most competing LMs.[4]

[2]We differentiate *training* and *pre-training* of an LM that *pre-training* attempts to teach an LM fundamental linguistic knowledge, while *training* involves training an LM for a downstream task. In that sense, *pre-training* solely relies on textual data and does not require supervised labels. *training*, on the opposite, necessitates supervised labels for a specific task.

[3]For example, the potential risks faced by a firm are discussed in both Item 1A (Risk Factors) and Item 7 (Management's Discussion and Analysis of Financial Condition and Results of Operations).

[4]The term "infeasible" denotes a scenario where an LM demands extensive memory usage exceeding the available machine memory, causing an out-of-memory error.

We then evaluate LongFinBERT's ability to produce meaningful text-based features in the accounting and financial context. To this end, LongFinBERT is utilized to detect misreporting in financial statements. Initially, we employ LongFinBERT to extract numerical representations from 10-K filings. Subsequently, we train a machine learning (ML) model, integrating these textual features with the accounting variables proposed by Dechow et al. (2011), to detect misreporting incidents. We show that text-based features from LongFinBERT enhance the misreporting detection ability of the accounting variables. Notably, LongFinBERT outperforms other text models, including FinBERT, in misreporting detection, thereby maintaining its superiority in this accounting-related task. While FinBERT is limited to processing partial 10-K filings, LongFinBERT's capability to analyze entire documents contributes significantly to its superior performance in misreporting detection using 10-K filings.

As an additional discussion, we analyze the effects of the data leakage phenomenon in misreporting detection, partly discussed by Walker (2021), on the result interpretation. This data leakage phenomenon arises from the prolonged existence of misreporting incidents spanning multiple years, potentially causing machine-learning models spuriously optimistic about detection accuracy under permissive research design. We demonstrate that the results after controlling for the leakage remarkably change. Through this observation, we raise our concerns about the result interpretation of several papers in this area such as Bao et al. (2020) and Brown et al. (2020).

To consolidate the capability of LongFinBERT in processing accounting disclosures, we revisit the research question explored by Brown and Tucker (2011), in which they examine how the year-over-year similarities of 10-K filings and the corresponding Management's Discussion and Analysis (MD&A) section relate to the subsequent stock market reactions. Different Brown and Tucker (2011) who employ a word-count-based method, we compute the year-over-year cosine similarity of 10-K filings and the MD&A section us-

ing the corresponding document vectors derived from LongFinBERT. We uncover a trend toward the increased similarity of those disclosures over time, consistent with Brown and Tucker (2011)'s findings. However, in contrast to their conclusions, we find that the stock market is indeed responsive to the year-over-year alternations of the MD&A section.

This work primarily contributes to the burgeoning research domain that employs LMs in accounting and finance. Inherit the spirit of Huang et al. (2023), who encourage researchers to leverage NLP advancements in accounting and financial research, LongFinBERT facilitates the utilization of LMs in analyzing accounting disclosures, owing to its computational efficiency. We openly release LongFinBERT's pre-trained set of parameters with the anticipation that it will show value in research using LMs in analyzing lengthy and complicated accounting documents.

Our contribution extends to the field of misreporting detection, which has been under study since the 1990s (Dechow et al., 1996; Beneish, 1997, 1999). While recent research has attempted to leverage NLP techniques to exploit the abundance of accounting textual data (Brown et al., 2020), the increasing size and complexity of such documents have constrained these studies to traditional word-count-based methods. To our knowledge, we are the first to tackle this challenge using a state-of-the-art LM. The superior performance of LongFinBERT in this accounting-related task highlights the potential of text-based features extracted from accounting disclosures using modern NLP techniques in detecting misreporting instances.

We organize this paper into seven sections including this introduction as Section 1. Section 2 reviews the current state of NLP models in accounting and financial research, eliciting the necessity for LongFinBERT. Section 3 elaborates on the construction and pretraining of LongFinBERT. The simulation study of LongFinBERT's computational efficiency is described in Section 4. In Section 5, we delve into the application of LongFinBERT in financial misreporting detection, covering the introduction of the data, research

design, and evaluation metrics. Section 6 analyzes the year-over-year similarity of management disclosures using LongFinBERT and its associations with the stock market. Finally, Section 7 concludes the paper while discussing potential limitations and future applications of LongFinBERT.

## 2 NLP algorithms in economic research

Generally, new studies in economics that use text data rely on several algorithms to represent documents in numerical forms in order to quantify economic relationships between variables. With the recent rapid development in NLP, many algorithms have been introduced to fulfill the representation task. These algorithms, however, can be categorized into three large categories: (i) word-count-based algorithms; (ii) neural embedding algorithms; and (iii) language models.

***Word-count-based algorithms***

The most classical way to represent a document is via the occurrences of words in the document, referred to as *Bag-of-words* (Harris, 1954). Particularly, the algorithm begins by defining a set of vocabulary. A document is then represented by a vector containing the frequency of each word from the pre-defined vocabulary appearing within that document. Later, Manning and Schutze (1999) suggest a transformation of the word frequency, named as *term frequency - inverse document frequency* (tf.idf), to obtain a more efficient document representation.

Due to its straightforward implementation, this method still maintains a foothold in current economic research. Jegadeesh and Wu (2013) represent 10-K filings by the tf.idf of words chosen from the financial sentiment vocabulary of Loughran and McDonald (2011) to identify which words make a stronger impression on investors. Also based on

the word-count method, Chen et al. (2019) examine the gain obtained by FinTech innovations of corporations in the financial service sector. Frankel et al. (2022) combine the word-count-based document representation and various machine learning (ML) methods, demonstrating that sentiment expressed in 10-K filings correlates with contemporaneous market reactions. Chen et al. (2022) propose a supervised approach to measure the tone of NASDAQ articles using their word-count representation. Deviating from document representation, a significant body of literature relies on the word-count method to measure several textual attributes and connect them to economic variables; see Tetlock (2007), Loughran and McDonald (2011), Brown and Tucker (2011), Price et al. (2012), Muslu et al. (2015), Jiang et al. (2019), Cho and Muslu (2021), among others.

Despite the popularity of the word-count method in current economic research, it possesses two well-known downsides. First, word-count-based representation is typically very high-dimensional, thus imposing more pressure on downstream machine learning or econometric models. This fact is caused by the proliferation of words used to compile texts. For example, the vocabulary used in Loughran and McDonald (2011) and Jegadeesh and Wu (2013) contains more than $3,000$ words. Chen et al. (2022) derive vocabulary consisting of approximately $43,500$ words and must resort to regularization. Second, because of the sole reliance on word frequency, the word-count method treats words as independent tokens and, therefore, overlooks other important linguistic features such as word semantics or word order (Mikolov et al., 2013).

***Neural embedding algorithms***

Neural embedding algorithms like Word2Vec (Mikolov et al., 2013) or Doc2Vec (Le and Mikolov, 2014) are introduced to overcome the aforementioned issues of the word-count method. Word2Vec is a neural-network-based algorithm that examines the similarity between words based on their co-occurrences in a corpus. The output of Word2Vec is a set of dense and low-dimensional semantic vectors, often referred to as *word embeddings*,

associated with every word within the vocabulary. Words sharing similar meanings or usage tend to have closer embeddings, based on Word2Vec, in the vector space, thus capturing word semantics. Doc2Vec is a modification of Word2Vec applied for documents instead of words. Consequently, through Doc2Vec, documents are represented by dense, low-dimensional, and semantic vectors, called *document embeddings*, as opposed to the word-count method.

With these advances, neural embedding algorithms have gained much attraction from economic researchers recently. Li et al. (2021) leverage Word2Vec to quantify corporate culture by analyzing earnings call transcripts of US firms. Fengler and Phan (2023) use Word2Vec to model topics in the Management's Discussion and Analysis section of 10-K filings. Furthermore, Miranda-Belmonte et al. (2023) utilize various neural embedding algorithms to construct an economic policy uncertainty index derived from US and Mexican newspapers. Using Doc2Vec, Rawte et al. (2020) measure the similarity of 10-K filings of the same firms between two consecutive years. They show that textual similarity strongly predicts accounting variables.

A major limitation of neural embedding algorithms, however, lies in their static embedding nature. Once these models are fully trained, each word becomes associated with a fixed vector representation. This can be problematic, as words often carry different meanings depending on context. Huang et al. (2023) provide a canonical example of the word "bank" in which it has different meanings in the two following sentences: "I went to the bank to deposit a check" and "We walk along the river's bank". Nevertheless, Word2Vec embeds the word "bank" into a single vector, failing to capture its contextual nuances.

Similarly, algorithms like Doc2Vec encode entire sentences into fixed vectors, regardless of word order. This means that sentences composed of the same words, even if in different arrangements, receive identical embeddings. For example, in our experiment using Doc2Vec, the two sentences "Sales increase but expenses decrease in this year" and "Sales

decrease but expenses increase in this year" have identical embeddings despite their opposite meanings.[5]

***Language models (LMs)***

To directly address the static embedding drawback of neural embedding algorithms, language models (LMs) are introduced. LMs are generally deep neural networks that rely on *multi-head attention* (Vaswani et al., 2017), a mechanism that allows words to pay attention to other words in a document, thereby producing dynamically contextualized word embeddings. Numerous LMs have been developed and introduced, namely BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), DeBERTa (He et al., 2021). Recent studies regarding how LMs work reveal that their word embeddings consist of both syntactic and semantic knowledge (Rogers et al., 2021), remarkably enhancing their linguistic capability (Devlin et al., 2019).

Aiming to provide a powerful LM specialized in the financial context, Huang et al. (2023) introduce FinBERT, an LM fine-tuned using a large financial corpus of corporate filings, analyst reports, and earnings call transcripts. They show that FinBERT outperforms other text models in sentiment and ESG (environmental, social, and governance) classifications. Using FinBERT, Cao et al. (2023) examine how firms adapt their communication behavior with investors and stakeholders knowing that their disclosures are analyzed by ML models. Besides that, several authors attempt to adapt BERT and other LM's architectures for specialized financial applications, namely SEC-BERT (Loukas et al., 2022), FinChain-BERT (Yang, Zhang, Sun, Pang, Jing, Wa and Lv, 2023), FinGPT (Yang, Liu and Wang, 2023).

[5]To conduct this small experiment, we use a Python package named *gensim* (Řehůřek and Sojka, 2010). However, it is important to acknowledge that the document embeddings generated by this package are subject to some level of randomness. Consequently, a Doc2Vec model trained by *gensim* may produce different embeddings even with identical documents. To mitigate this variability, it is recommended to set a very high value to the `epochs` argument in the `similarity_unseen_docs` method of the Doc2Vec model. In our case, we opt for a setting of $10,000$ epochs and get a cosine similarity between these two sentences of 0.9980, while the cosine similarity between the first sentence and itself is 0.9976.

Although the multi-head attention mechanism is at the heart of the success of LMs, it presents a bottleneck, hindering LMs from effectively handling lengthy documents. Specifically, the multi-head attention mechanism operates by calculating a square matrix, referred to as *attention matrix*, whose dimensionality is the document length (i.e., the number of tokens in a document).[6] This calculation exhibits a quadratic computational complexity in terms of document length. Specifically, when a document is doubled in length, the cost to compute the attention matrix increases fourfold. For example, consider a document containing $55,000$ tokens, which is the average length of 10-K filings (Cazier and Pfeiffer, 2016). To compute a BERT attention matrix with twelve heads using single-precision floating-point format (i.e., 32 bits or four bytes), one needs $4 \times 55,000 \times 55,000 \times 12 = 145,200,000,000$ bytes = 145 gigabytes of memory to process only one layer of BERT, which has twelve layers and other parameters. Such a memory demand exceeds the feasibility of all current Graphics Processing Units (GPUs), which are the main workhorses for training LMs.

Many different attention mechanisms are proposed to deal with this computational challenge. The common idea is to sparsify the attention matrix by a combination of sliding-window and global attention mechanism (Zaheer et al., 2020; Beltagy et al., 2020) or via Nyström method to approximate the attention matrix (Xiong et al., 2021). These methods partly exhibit the ability to scale LMs for longer documents yet the computational burden remains remarkable (see Section 4). Another approach is dividing documents into finer units, such as paragraphs or sentences (Huang et al., 2023). However, this approach put the linguistic coherence between paragraphs in jeopardy.

Overall, this computational challenge remains one of the main factors delaying applica-

[6]Text models that are not LMs treat words as the smallest units in processing texts. Therefore, they break documents into words (or phrases) when generating vocabulary. This process is called *tokenization*. LMs, however, tokenize documents more granularly, further breaking words into *subwords* occasionally. For instance, BERT (Devlin et al., 2019) splits the word "volatility" into three subwords "vol", "ati", and "lity", but does not split the word "cash". We follow Huang et al. (2023) to define *tokens* as words and subwords.

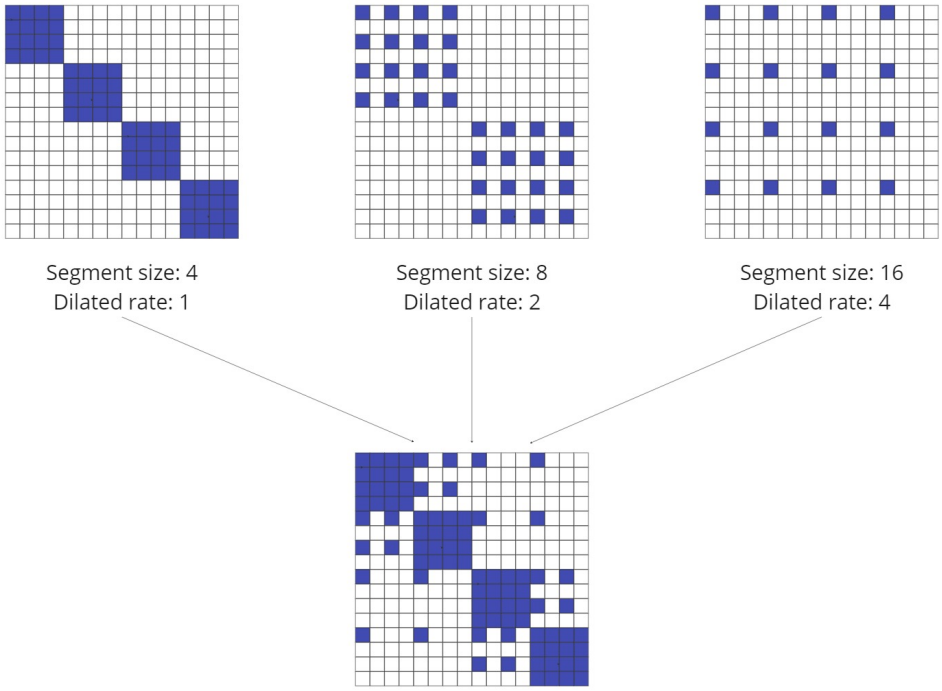



**Figure 1:** This figure visualizes the attention matrix processed by LongNet Ding et al. (2023). Each small square indicates the interaction between two tokens. For example, the second small square in the first row presents the interaction between the first and second tokens in a document. Colored small squares indicate the considered interactions while white squares denote ignored ones, resulting in a highly sparse attention matrix. This contrasts with the vanilla attention matrix used by FinBERT (Huang et al., 2023), where all squares are colored, indicating a dense attention matrix.

tions of LMs in economic research, where text data typically involve very lengthy documents. Our goal with this paper is to bridge this gap by developing an LM, named LongFinBERT, tailored for very long financial documents. We introduce LongFinBERT with the anticipation that it could facilitate the applications of LMs in processing financial disclosures.

# 3 LongFinBERT - A BERT-like language model specialized for long financial documents

At the heart of LongFinBERT is the dilated attention mechanism developed by Ding et al. (2023), which is visualized in Figure 1. Specifically, the dilated attention mechanism ap-

plies multiple segments on a (long) document. Each segment has a different size and dilated rate, aiming to sparsify the attention matrix. Segment size controls for local information captured by the segment. For example, a segment size of four tokens allows the model to focus on every four consecutive tokens. Segment dilated rate controls for the sparsity of the attention matrix. A dilated rate of four implies that the first token only pays attention to the fifth, ninth, and thirteenth tokens, etc. By increasing the dilated rate, one can allow the attention matrix to capture the interactions between tokens located at a far distance in a long document, thus guaranteeing the sparsity of the attention matrix. Finally, the attention matrices processed by multiple segments with increasing segment sizes and dilated rates are aggregated. The aggregated attention matrix, consequently, incorporates both local (by small segment sizes and dilated rates) and global (by large segment sizes) while staying sparse (by large dilated rates).

Relying on the dilated attention mechanism, we build an encoder-only transformer architecture, similar to BERT (Devlin et al., 2019). LongFinBERT is then constructed by stacking twelve encoders together, as shown in Figure 2. Unlike FinBERT which leverages the pre-trained weights (i.e., model parameters) from BERT, LongFinBERT is constructed by a brand-new attention mechanism. Consequently, there are currently no pre-trained weights for LongFinBERT to inherit. Intuitively, while FinBERT can capitalize on fundamental linguistic understanding from BERT's extensive pre-training, LongFinBERT needs to learn these foundational elements anew. However, equipping an LM with fundamental linguistic skills requires massive data and computational costs (Devlin et al., 2019; Zaheer et al., 2020). To address this challenge, we enable LongFinBERT to adopt FinBERT's tokenization and word embedding layer, the two deepest parts of FinBERT's model parameters. This strategy allows LongFinBERT to leverage fundamental linguistic knowledge specific to the financial domain of FinBERT, thus relieving the demand for extensive data and computational power during pre-training LongFinBERT.

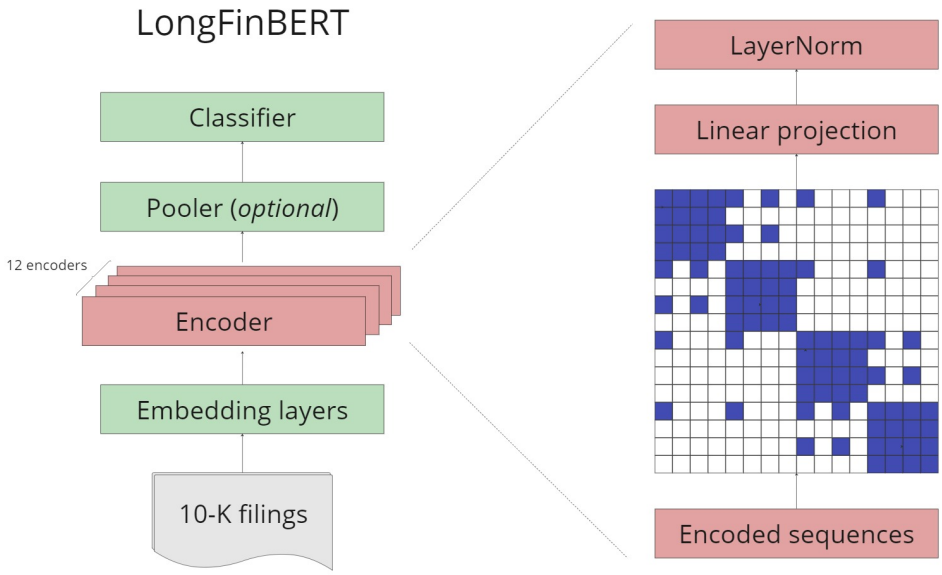



**Figure 2:** This figure illustrates the architecture of LongFinBERT, with the core component being LongNet, detailed in Figure 1. Initially, a document (e.g., 10-K filing) is converted to numerical representations by embedding layers. The encoded sequences are then passed through twelve encoders. Each encoder is constructed by the LongNet attention mechanism, followed by a linear projection and a normalization layer (i.e., LayerNorm). After the twelve encoders, the encoded document is processed by an optional pooling layer (i.e., Pooler) and a classification layer (i.e., Classifier).

Similar to BERT and FinBERT, we employ the masked language modeling task to pre-train LongFinBERT. Specifically, we randomly mask 15% number of tokens in a document. LongFinBERT is then taught to predict what the masked tokens are based on the remaining unmasked tokens. We pre-train LongFinBERT using a large set of 100,717 10-K filings of US firms from 1993 to 2008. The dataset from 2009 to 2013 is used for model validation and includes 44,342 10-K filings. We selected the dataset of 10-K filings for pre-training LongFinBERT because of three reasons. First, 10-K filings are usually very lengthy with approximately 55,000 words on average (Cazier and Pfeiffer, 2016), aligning well with the intended spirit of LongFinBERT in handling long financial documents. Second, 10-K filings are widely used by other economic and financial studies involving text data (Loughran and McDonald, 2011; Li, 2010; Jegadeesh and Wu, 2013; Jiang et al., 2019; Brown et al., 2020; Huang et al., 2023). Third, 10-K filings are chosen over 10-Q filings as Griffin (2003) show that the stock markets are more responsive to the information of 10-K compared to 10-Q filings.

In total, our pre-training sample includes 5.85 billion tokens. Despite containing fewer 10-K documents, our corpus is about 20% and 77% larger than the corpora used to pre-train FinBERT and BERT, respectively. This is because we consider the whole 10-K filings instead of focusing on only several sections or short documents as in Huang et al. (2023). We pre-trained LongFinBERT with an AMD EPYC 7763 machine with 64 cores and an NVIDIA A100-PCIe with 80 gigabytes of GPU memory. We apply five different segments to LongFinBERT with the segment sizes of $\{16, 128, 512, 1024, 2048\}$ and dilated rates of $\{1, 16, 64, 256, 512\}$, respectively. LongFinBERT is trained with eight iterations, in which each iteration takes around 14 hours. Compared with FinBERT, which spent approximately 72 hours on one iteration, LongFinBERT exhibits a five-fold improvement in the pre-training time. We make the pre-trained model weights and training code publicly available.[7]

# 4 Computational efficiency of LongFinBERT

In Section 3, LongFinBERT is introduced aiming to improve the computational efficiency of LMs. Particularly, it simplifies the computational complexity of the standard multi-head attention mechanism from quadratic to linear complexity. Therefore, LongFinBERT is expected to outperform other LMs in practice in terms of computational costs, in which time and memory are the main two aspects.

Accordingly, we design a simulation study in order to compare the computational performance of LongFinBERT with that of four other LMs. The competitors include: (i) Google Big Bird (Zaheer et al., 2020), Longformer (Beltagy et al., 2020), Nyströmformer (Xiong et al., 2021), and FinBERT (Huang et al., 2023). The first three LMs are selected due

[7]The pre-trained model weights: https://huggingface.co/minhtriphan/LongFinBERT-base.
The training code: https://github.com/minhtriphan/LongFinBERT-base/tree/main.

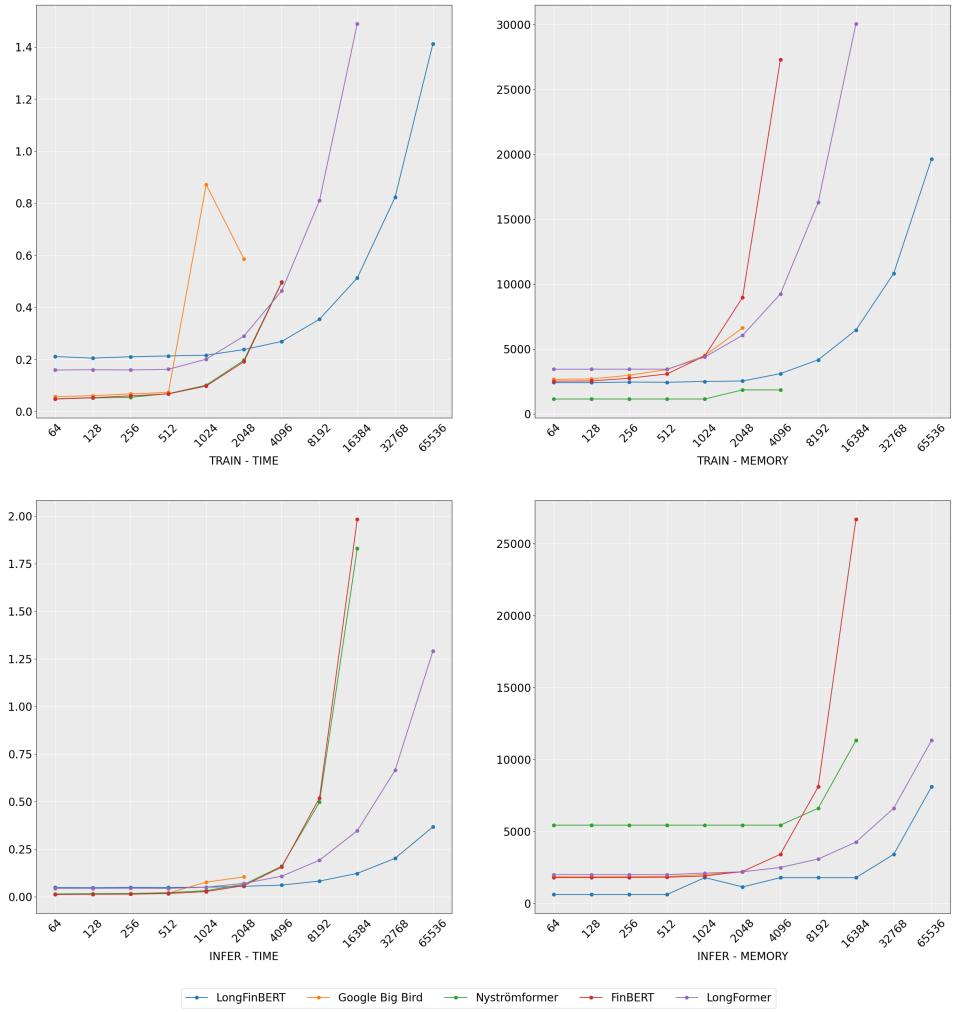



**Figure 3:** This figure reports the results of the simulation study comparing the computational performances of LongFinBERT (blue) with those of four other competing LMs, namely Google Big Bird (orange), Nyströmformer (green), FinBERT (red), and Longformer (violet). The results include the training time (top-left), training memory (top-right), inferring time (bottom-left), and inferring memory (bottom-right) of all five LMs. The document length (number of input tokens) is displayed in the $x$-axes using a logarithmic scale. The time (in seconds) and memory (in megabytes) are displayed in the $y$-axes.

to their capability to handle long documents. FinBERT is chosen, however, because of its popularity in economic/financial/accounting studies (Cao et al., 2023; Park and Cho, 2023; Schimanski et al., 2024). It is worth noting that we refrain from comparisons in the computational performance of LongFinBERT with non-LM algorithms. This is because non-LM algorithms, such as Bag-of-words, Word2Vec, or Doc2Vec, are typically trained on Central Processing Units (CPUs) meanwhile Graphics Processing Units (GPUs) are more commonly used to train LMs. Therefore, a comparison of the computational costs between these two model genres would be generally unreasonable.

To set up the simulation study, we generate a dummy document with variable lengths ranging from 64 ($2^6$) tokens to $65,536$ ($2^{16}$) tokens. Subsequently, we examine the computational time and memory of the five LMs, including LongFinBERT, to process this document in both two phases, training and inferring.[8] In particular, during the training phase, we train each LM with the synthetic document for one iteration per document length value. In the inference phase, we employ each LM to infer the vectorized representation of the dummy document. To mitigate the impact of noise, we run each experiment 500 times and report the median values of time and memory usage. All experiments are conducted using an NVIDIA A100-SXM4 with 40 gigabytes of GPU memory.

It should be noted that there are limits of the competing LMs regarding the maximum length of documents that can be passed through these LMs. For example, FinBERT can only process documents with no longer than 512 tokens (Huang et al., 2023). Similarly, the maximum limit for Google Big Bird, LongFormer, and Nyströmformer is $4,096$ tokens. We attribute these limitations to technical constraints inherent in the positional embedding layer of each LM.[9] For example, when a document with $4,097$ tokens is passed through Google Big Bird, the vector representation for the $4,097^{th}$ position is unidentified because vector representations are provided for only $4,096$ positions. Appearing as a technical problem, this limitation possibly causes unfair comparisons between the LMs, especially with FinBERT, when they can theoretically handle documents longer than these limits given 40 gigabytes of GPU memory. We, therefore, overcome this constraint by expanding the positional embedding layers in the LMs. This modification expectedly ensures an equitable comparison environment for all involved LMs.

[8] Different from the inferring phase, during the training phase, the gradients (i.e., derivatives of the loss function with respect to the model parameters evaluated at the current data point) are computed by the back-propagation algorithm (Linnainmaa, 1976) and stored in the memory of the used GPUs to implement the stochastic gradient descent optimization (Robbins and Monro, 1951). This operation demands high memory capacity from the GPUs in use as well as longer processing time.

[9] Positional embedding layers in LMs aim to encode different positions of input tokens by distinct numerical vectors. Consequently, a token (or word) appearing at two different locations in a text, at the end, will be represented by two different vectors. This mechanism allows LMs to capture better word contexts.

Figure 3 showcases the results of the simulation study. In the training phase, LongFinBERT demonstrates the capability to handle a document with $2^{16}$ tokens. Meanwhile, the other LMs only process a document with only $2^{14}$ tokens at most with LongFormer, which is four times shorter than the longest document supported by LongFinBERT. In the inference phase, LongFinBERT remains the top performer in terms of both time and memory usage. In particular, in comparison with LongFormer, the most efficient among the competing LMs, which requires more than 1.25 seconds and $11,000$ megabytes to vectorize a document with $2^{16}$ tokens, LongFinBERT only needs approximately 0.35 second and $8,000$ megabytes to process a document with the same length. That constitutes an advantage of LongFinBERT by 3.6 (time) $\times$ 1.38 (memory) $\approx$ 5 times over the best competing LM.

We observe that, however, LongFinBERT exhibits longer processing times compared to the other LMs when handling documents shorter than 512 tokens. Its performance improves when the document length surpasses $2^{10}$ tokens and becomes strongest with documents longer than $2^{12}$ tokens. This phenomenon is caused by the utilization of many segments with different sizes in LongFinBERT. Therefore, with short documents, small-size segments collide, leading to redundancy.[10] Notably, this phenomenon is primarily evident in time usage. In contrast, LongFinBERT consistently outperforms other LMs in terms of memory usage, regardless of the document length.

Overall, while LongFinBERT may require more time to process short documents, it demonstrates the capability to efficiently process long documents compared to other LMs, including those that are designed for lengthy texts.

[10] Consider the scenario where LongFinBERT uses two segments of sizes 64 and 512 respectively. When processing a document with 512 tokens, LongFinBERT utilizes both segments to compute the attention matrix. In contrast, the other LMs compute the attention matrix only once.

# 5 Performance of LongFinBERT on financial misreporting detection

So far, LongFinBERT has shown its superior performance in handling long texts. This result opens avenues for research whose objects are extremely long documents. One promising area of exploration involves developing models for detecting financial misreporting using financial disclosures, which often span hundreds of pages.

## 5.1 Literature on financial misreporting detection

Studies of financial misreporting have surged as an important pillar in modern accounting research in recent years. Many studies have highlighted the relationship between misreporting behavior and accounting variables, thus eliciting the usefulness of accounting data to detect misreporting. For instance, Dechow et al. (1996) observe that misreporting firms exhibit a high amount of accruals before the alleged accounting manipulation. Beneish (1999) propose that variables regarding firm revenues, expenses, and sales growth can help to identify earning manipulations. Building upon the empirical evidence, Dechow et al. (2011) construct a measure called *F-score*, a composite of several financial variables, that can detect misreporting firms. However, Larcker and Zakolyukina (2012) argue that financial variables typically yield limited detection results as many measures exhibit contrary behavior to conventional expectation (Purda and Skillicorn, 2015).

To address the weaknesses of the financial variables, new studies have employed language-based measures with the anticipation of enhancing misreporting detection power. Li (2008) discover a positive connection between the more frequent use of causation words, optimistic words, and future tense verbs in the Management's Discussion and Analysis

section of 10-K filings. By studying conference call transcripts, Larcker and Zakolyukina (2012) find that attributes extracted from emotion words have stronger performance in detecting misreporting compared to abnormal accrual measures. Recently, Brown et al. (2020) use Latent Dirichlet Allocation (LDA), a word-count-based topic modeling framework, to analyze 10-K filings. Their findings suggest that thematic contents comprised in 10-K filings assist financial variables to improve misreporting detection ability.

So far, despite the immense linguistic capabilities of state-of-the-art LMs, the utilization of 10-K filings in misreporting detection heavily relies on word-counts-based methods, primarily due to the verbosity of the financial documents. Notably, 10-K filings have $55,000$ words on average (Cazier and Pfeiffer, 2016) with the possibility of lengthening even further (Brown and Tucker, 2011; Dyer et al., 2017). This fact presents a challenge in employing LMs in detecting misreporting behavior using 10-K filings due to their quadratic complexity in both time and memory dimensions, thus necessitating more efficient approaches.

Attempting to address this challenge, applying LongFinBERT to 10-K filings in detecting misreporting is efficient in two ways. First, LongFinBERT inherits the superior linguistic capabilities of other LMs in the same genre such as FinBERT. Therefore, LongFinBERT, in comparison to non-DL approaches, is expected to capture more complicated linguistic structures in 10-K filings. This essence of LongFinBERT allows us to exploit useful text-based features to detect misreporting firms. Second, in comparison to other LMs, LongFinBERT is more efficient in dealing with long documents compared to other LMs regarding computational power as shown in Section 4. Given these two benefits, the introduction of LongFinBERT to misreporting detection holds promise for overcoming the aforementioned challenge.

## 5.2 Data

We rely on three different data sources in this application: (i) a text corpus, from which we extract text-based features; (ii) financial and accounting data, used to construct Dechow et al. (2011)'s financial variables; and (iii) a dataset to identify misreporting incidents.

***The text corpus***

Extending from Brown et al. (2020), we extract text-based features from annual financial statements of US firms, 10-K filings, from 1994 to 2018. We download 10-K filings from 1st January 1994 to 31st December 2018 in the Notre Dame Software Repository for Accounting and Finance (SRAF).[11] The download yields in total 183,057 10-K documents filed by 34,972 US firms over the sample period. In accordance with the current norm in using LMs (Huang et al., 2023), we do not apply intensive text normalization steps except excluding the SEC header in each 10-K file. To apply non-LM approaches, inspired by Frankel et al. (2022), we execute several standard steps of text normalization, including word lowercasing, contraction fixing, number removal, punctuation removal, and lemmatization.

***Financial variables***

Following Dechow et al. (2011), we construct 17 financial variables capturing firm size, audit quality, and involvement in complex business transactions, namely, merger and acquisition activities, and restructurings. These variables include: total assets ($\log(Assets)$), accruals calculated by Richardson et al. (2005) ($RSST_Accruals$), changes in account receivables ($\Delta Receivables$), changes in inventory ($\Delta Inventory$), soft asset ratio ($\%Soft_assets$), changes in return on assets ($\Delta ROA$), dummy on securities issuance ($Actual_issuance$), dummy on carrying future operating lease obligations ($Operating_leases$), dummy on

[11] https://sraf.nd.edu/sec-edgar-data/cleaned-10x-files/

merger and acquisition ($M\&A$), lag of firm stock excess log returns over one year prior to the filing date ($Lag(r)$), dummy on being audited by the Big-4 auditors, i.e., KPMG, Deloitte, EY, and PwC ($Big4Auditors$), dummy on being audited by mid-size auditors, i.e., BDO, Grant Thornton, McGladrey ($MidSizeAuditors$), cash flow from financing activities ($TotFinancing$), dummy on ex-ante financing needs ($ExanteFinancing$), dummy on nonzero restructuring charges ($Restructuring$), and the book-to-market ratio ($B/M$).

We refer readers to Dechow et al. (2011) and Appendix B in Brown et al. (2020) for detailed instructions on calculating these variables. We compute them using the Compustat and CRSP databases. After the construction, we obtain a dataset of $62,037$ observations coming from $8,843$ US firms over the sample period. As Dechow et al. (2011) use these variables to construct *F-score* as a measure to detect misreporting firms, we use these variables to train a reliable benchmark model.

***Identifying misreporting behavior***

We rely on the dataset of SEC Accounting and Auditing Enforcement Releases (AAERs) to identify misreporting firms, as widely used in misreporting detection research (Beneish, 1999; Dechow et al., 2011; Perols et al., 2017; Brown et al., 2020; Bertomeu et al., 2021; Yi et al., 2023). The dataset was initially collected by Dechow et al. (2011) and recently updated by the Marshall Business School of the University of South California. The dataset comprises $4,278$ SEC AAERs, revealing $1,816$ detected cases of misreporting involving $1,768$ distinct firms ranging from 1971 to 2019. Among these cases, two firms committed misreporting three times, 44 firms were found to misreport twice, and $1,722$ firms were implicated in misreporting once. On average, a misreporting case lasts approximately 2.33 years and involves about 1.88 misreporting categories. The longest case involves Comverse Technology, Inc. (Ulticom) lasting from 1991 to 2005 in which they were alleged to have misreported the cost of goods sold, reserve accounts, expenses/shareholder equity, and liabilities.

| Year | Observations | Frequency | | | | | | | | | | | | Percentage (%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | *REV* | *REC* | *COGS* | *INV* | *RES* | *DEBT* | *MKT_SEC* | *EXP_SE* | *ASSET* | *PAY* | *LIAB* | Total | |
| 1994 | 882 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0.11 |
| 1995 | 1,051 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 2 | 1 | 1 | 4 | 0.38 |
| 1996 | 1,601 | 9 | 4 | 3 | 3 | 3 | 1 | 0 | 4 | 2 | 1 | 3 | 13 | 0.81 |
| 1997 | 2,238 | 5 | 5 | 2 | 3 | 1 | 1 | 0 | 8 | 2 | 1 | 1 | 15 | 0.67 |
| 1998 | 2,273 | 14 | 9 | 5 | 5 | 4 | 0 | 0 | 18 | 4 | 1 | 4 | 30 | 1.32 |
| 1999 | 2,139 | 22 | 11 | 3 | 5 | 3 | 1 | 0 | 16 | 5 | 1 | 2 | 34 | 1.59 |
| 2000 | 1,961 | 37 | 26 | 6 | 12 | 4 | 2 | 0 | 19 | 8 | 3 | 3 | 51 | 2.60 |
| 2001 | 1,902 | 27 | 20 | 2 | 5 | 3 | 0 | 0 | 31 | 6 | 3 | 3 | 53 | 2.79 |
| 2002 | 2,095 | 17 | 11 | 4 | 6 | 6 | 0 | 0 | 35 | 10 | 3 | 3 | 42 | 2.00 |
| 2003 | 2,501 | 18 | 10 | 2 | 7 | 3 | 0 | 0 | 38 | 10 | 3 | 5 | 47 | 1.88 |
| 2004 | 2,411 | 9 | 3 | 1 | 4 | 4 | 0 | 0 | 28 | 2 | 2 | 1 | 33 | 1.37 |
| 2005 | 2,347 | 10 | 5 | 2 | 3 | 4 | 0 | 0 | 23 | 5 | 3 | 3 | 30 | 1.28 |
| 2006 | 2,313 | 7 | 2 | 6 | 3 | 3 | 0 | 0 | 12 | 6 | 2 | 3 | 17 | 0.73 |
| 2007 | 2,362 | 8 | 0 | 2 | 1 | 2 | 0 | 0 | 11 | 5 | 1 | 3 | 14 | 0.59 |
| 2008 | 2,365 | 8 | 0 | 2 | 0 | 0 | 0 | 0 | 9 | 5 | 1 | 3 | 12 | 0.51 |
| 2009 | 2,405 | 8 | 0 | 2 | 1 | 0 | 0 | 0 | 9 | 3 | 1 | 4 | 13 | 0.54 |
| 2010 | 2,318 | 9 | 3 | 1 | 2 | 0 | 0 | 0 | 6 | 2 | 0 | 2 | 15 | 0.65 |
| 2011 | 2,204 | 8 | 3 | 2 | 3 | 0 | 0 | 0 | 1 | 1 | 0 | 4 | 13 | 0.59 |
| 2012 | 2,111 | 10 | 6 | 4 | 2 | 1 | 0 | 0 | 1 | 1 | 0 | 3 | 14 | 0.66 |
| 2013 | 2,035 | 3 | 3 | 2 | 1 | 2 | 0 | 0 | 2 | 3 | 0 | 2 | 9 | 0.44 |
| 2014 | 1,933 | 5 | 1 | 1 | 2 | 2 | 0 | 0 | 3 | 0 | 0 | 3 | 9 | 0.47 |
| 2015 | 1,858 | 2 | 0 | 1 | 1 | 2 | 0 | 0 | 3 | 1 | 0 | 2 | 5 | 0.27 |
| 2016 | 1,807 | 4 | 0 | 2 | 3 | 2 | 0 | 0 | 3 | 3 | 0 | 1 | 8 | 0.44 |
| 2017 | 1,774 | 4 | 0 | 1 | 3 | 1 | 0 | 0 | 1 | 3 | 0 | 0 | 7 | 0.39 |
| 2018 | 1,777 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 0.11 |
| *Total* | 50,663 | 245 | 122 | 58 | 77 | 51 | 6 | 1 | 281 | 90 | 27 | 59 | 491 | 0.96 |
| *No. of firms* | 7,791 | 120 | 58 | 31 | 39 | 20 | 4 | 1 | 98 | 40 | 11 | 24 | 212 | |

**Table 1:** This table reports the number of observations, frequency, and the corresponding percentage of misreporting cases by misreporting types and by year obtained from the dataset after the matching. The data sample runs from 1st January 1994 to 31st December 2018.

In line with Dechow et al. (2011), misconduct events unrelated to accounting or auditing, such as bribery, disclosure-related and timing violations, or events that lack clear identification of the underlying cause of the violations are excluded. There are eleven remaining misconducts including the misreporting behaviors regarding: revenues ($REV$), other expense/shareholder equity accounts ($EXP_SE$), capitalized costs as assets ($ASSET$), account receivable ($REC$), inventory ($INV$), cost of goods sold ($COGS$), liabilities ($LIAB$), reserve accounts ($RES$), allowance for bad debt ($DEBT$), marketable securities ($MKT_SEC$), and payables ($PAY$). We identify misreporting firms when they commit to at least one of these misconducts. Accordingly, these firms are tagged with $is_misreporting = 1$ in contrast to non-misreporting firms with $is_misreporting = 0$.

Finally, we match the three datasets, i.e., 10-K filings, financial variables, and AAERs, to obtain the final dataset used to train our misreporting detectors. The final dataset

comprises 50,663 observations from 7,342 distinct firms spanning the period from 1st January 1994 to 31st December 2018. There are 491 firm-year misreporting observations, approximately accounting for 1% of the total number of observations. In comparison with Brown et al. (2020), whose sample period ranges from 1994 to 2010, our dataset exhibits a misreporting rate of 1.21% in the same sub-sample period. This quantity is comparable to the corresponding proportion reported by Brown et al. (2020), which is 1.34%.

## 5.3 Descriptive statistics

Table 1 reports the frequency of misreporting cases by misreporting types and by year obtained from the final dataset. In general, consistent with other studies using the AAERs dataset, the misreporting proportion appears very low. The number of misreporting cases exhibit a sharp increase until 2001, followed by a gradual decline thereafter. This variation coincides with the enactment of the Sarbanes-Oxley Act in 2002 which mandates several practices in financial and accounting reporting to protect investors from fraudulent behaviors of firms. It should be noted that there are many incidents when firms commit to more than one misreporting category. That explains the difference between the total number of misreporting firms (491) and the summation of the category-wise total numbers of cases.

Misreporting regarding incomes/expenses/shareholder equity, revenues, and account receivables are the most common misconducts engaged by firms with 281, 245, and 122 firm-year observations, respectively. While misreporting incomes/expenses/shareholder equity constitutes the highest number of misreporting cases (281), it is worth noting that revenue-related misreporting involves the highest number of firms with 120 distinct corporations found to be implicated. Furthermore, a singular incident of misreporting in marketable securities occurred in 1994, involving Pier 1 Imports, Inc., making it the sole

| | No misreporting ($N = 50,172$) | | Misreporting ($N = 491$) | | Difference |
|---|---|---|---|---|---|
| | Mean | Std. Dev. | Mean | Std. Dev. | |
| $\log(Assets)$ | 5.959 | 2.059 | 6.591 | 1.726 | -0.633*** |
| $RSST_Accurals$ | 0.025 | 0.202 | 0.071 | 0.247 | -0.046*** |
| $\Delta Receivables$ | 0.005 | 0.066 | 0.019 | 0.060 | -0.013*** |
| $\Delta Inventory$ | 0.004 | 0.047 | 0.011 | 0.063 | -0.008*** |
| $\%Soft_assets$ | 0.476 | 0.429 | 0.663 | 0.276 | -0.186*** |
| $\Delta Cash_sales$ | 0.171 | 1.591 | 0.242 | 0.609 | -0.070** |
| $\Delta ROA$ | -0.010 | 0.240 | -0.009 | 0.154 | -0.001 |
| $Actual_issuance$ | 0.884 | 0.320 | 0.921 | 0.271 | -0.036*** |
| $Operating_leases$ | 0.777 | 0.417 | 0.876 | 0.330 | -0.099*** |
| $M\&A$ | 0.151 | 0.358 | 0.088 | 0.283 | 0.063*** |
| $Lag(r)$ | -0.007 | 0.628 | -0.023 | 0.653 | 0.016 |
| $Big4Auditors$ | 0.727 | 0.446 | 0.778 | 0.416 | -0.051*** |
| $MidSizeAuditors$ | 0.101 | 0.301 | 0.088 | 0.283 | 0.014 |
| $TotFinancing$ | 0.030 | 0.196 | 0.029 | 0.134 | 0.001 |
| $ExanteFinancing$ | 0.087 | 0.281 | 0.029 | 0.167 | 0.058*** |
| $Restructuring$ | 0.013 | 0.115 | 0.022 | 0.148 | -0.009 |
| $B/M$ | 0.695 | 0.705 | 0.632 | 0.524 | 0.064*** |

**Table 2:** This table reports the descriptive statistics of 17 financial variables in two samples: one with misreporting and one without misreporting. The table also includes Welch's $t$-tests comparing the means of these variables between the two samples. The definitions of these variables are given in Section 5.2. The data sample spans the period from 1st January 1994 to 31st December 2018. ***, **, and * denote significance at the 1%, 5%, and 10% levels, respectively.

instance of this misconduct type. A chronological analysis reveals that several misreporting categories have disappeared in recent years, namely misstatement of account receivables, payables, and liabilities. Several other misreporting categories occurred less frequently yet consistently over time. Those include misstating costs of goods sold, inventories, reserve accounts, and capitalized costs as assets.

Table 2 displays the descriptive statistics (means and standard deviations) of 17 financial variables described in Section 5.2. Additionally, we also present Welch's $t$-tests of the mean difference of these variables between the no-misreporting and with-misreporting samples. Our statistics align with those of Brown et al. (2020) in terms of magnitude, relation direction, and statistical significance. Specifically, misreporting firms exhibit significantly higher total assets, accruals, and soft asset proportions. They demonstrate notably higher year-by-year changes in account receivables, inventory, and cash sales.

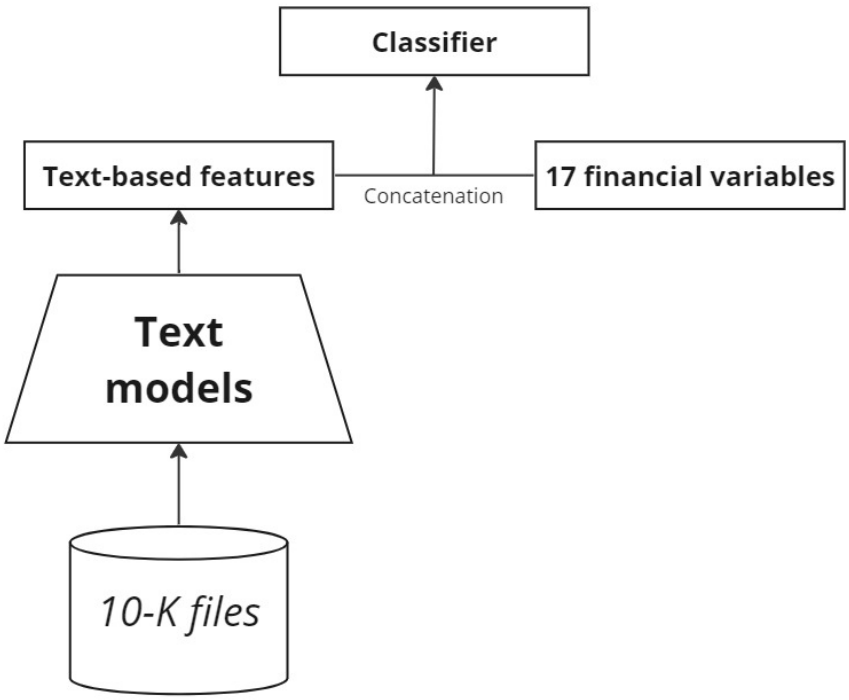



**Figure 4:** This figure illustrates the model diagram used to detect misreporting incidents. Text models include LongFinBERT, FinBERT, LDA, and Doc2Vec. 17 financial variables are introduced in Section 5.2.

Furthermore, misreporting corporations issue greater volumes of securities than their non-misreporting counterparts. Interestingly, consistent with Brown et al. (2020), our findings reveal that misreporting firms are more frequently audited by Big-4 auditing firms. In general, with many variables displaying significant differences between non-misreporting and misreporting samples, these variables demonstrate their potential for identifying misconduct behaviors.

## 5.4 Research design

### *The model*

Figure 4 illustrates the model diagram for our misreporting detection problem. Specifically, each 10-K file is passed through a text model to extract meaningful text-based

features in numerical form. It is worth noting that text models are adopted as feature extraction tools without further training or fine-tuning. The text-based features are then augmented with the 17 financial variables of the corresponding firm-year observation. This augmentation is achieved through vector concatenation, resulting in a unified vector that integrates both textual and numerical financial information for each firm-year observation. Subsequently, this combined vector serves as input to train a pre-defined classifier tasked with detecting misreporting. The classifier generates the probability of misreporting incidents when $is_misreporting = 1$ as its output.

Due to the high dimensionality of text-based features and the low frequency of misreporting cases, this detection challenge is identified as a binary classification with rare events in high-dimensional data. Consequently, we opt for gradient boosting (Freund et al., 1996) as the classifier. With the demonstration of strong performance in high-dimensional settings (Lusa et al., 2017), the algorithm is used by many studies in misreporting detection (Whiting et al., 2012; Bao et al., 2020; Bertomeu et al., 2021). However, Lusa et al. (2017) notice that boosting algorithms may struggle when encountering imbalanced data, particularly being prone to overfitting on training data. Accordingly, they suggest employing two regularization techniques to cope with imbalanced data: (i) shrinkage (Friedman, 2001); and (ii) subsampling (Friedman, 2002). We implement gradient boosting algorithms by adopting *XGBoost*, a Python library for gradient boosting developed by Chen and Guestrin (2016). *XGBoost* stands out as a preferred tool for developing boosting algorithms due to two crucial mechanisms: (i) embedded regularization within loss functions, and (ii) variable and observation subsampling mechanism. According to Lusa et al. (2017), these two features are deemed to mitigate the risk of overfitting, thus expectedly improving the misreporting detection capability.

***The time-series research design***

Figure 5 describes our time-series research design. In contrast to the rolling-window ap-

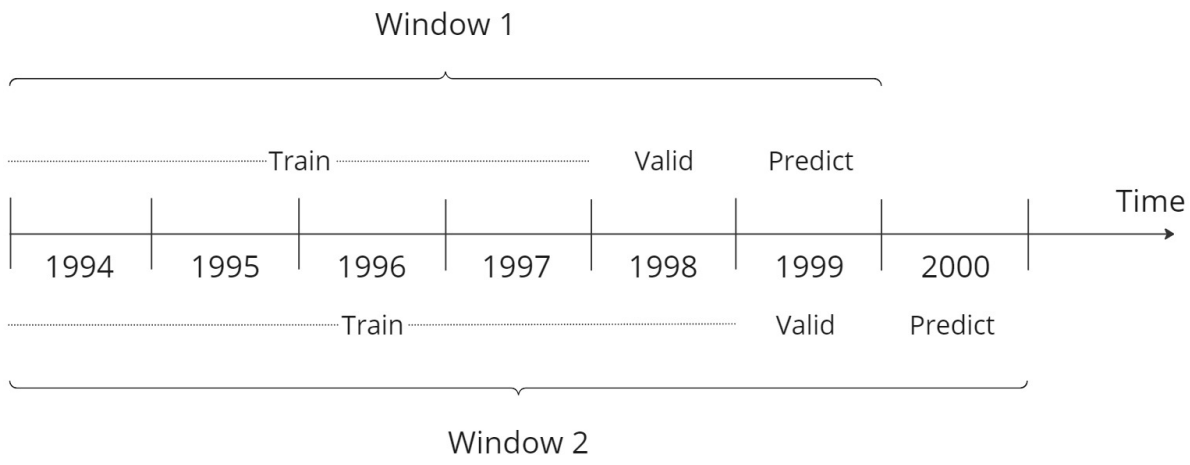



**Figure 5:** This figure illustrates the recursive-window research design. The training of the classifier starts with data from 1994 to 1997. The data in 1998 are used to optimize the classifier hyperparameters (number of boosting rounds), thereby mitigating the risk of overfitting. The predictions are made for the year of 1999. The cycle repeats by expanding the training data by one year in each step. Consequently, the data for hyperparameter optimization and prediction are then shifted by one year to the future. The process continues until the prediction year reaches 2018.

proach employed by Brown et al. (2020), where they maintain a fixed number of five years in the training sets, we opt for the recursive-window procedure (West and McCracken, 1998). This choice is motivated by the high-dimensional nature of our problem, necessitating more data to obtain reliable estimations. We initiate the process by using the data from 1994 to 1997 to train the boosting algorithm. The data in 1998 and 1999 are employed to optimize regularization hyperparameters (i.e., validating the model) and make predictions, respectively. The process continues by extending the training period, adding one year at a time. Concurrently, the sample periods for validation and prediction are shifted by one year with each extension. We provide details regarding training hyperparameters in Appendix A.[12]

Another difference between Brown et al. (2020) and our work is that we prevent cases

[12]LongFinBERT is pre-trained on 10-K filings spanning from 1994 to 2008, while the misreporting detection task employs a recursive rolling window approach with predictions made from 1999 to 2018. However, the practice of re-using pre-training data for a downstream task is, indeed, not unusual in economic research. Particularly, Huang et al. (2023) employ a large collection of analyst reports to pre-train FinBERT, subsequently re-using it for a sentiment classification task. They argue that this is a valid practice because the pre-training data contains only texts without sentiment labels. The same argument is applied to our case with misreporting labels. We refer readers to footnote 11 in Huang et al. (2023) for more details.

in which a misreporting event appears in both training and validating/predicting sets, causing a data leakage that leads to spurious results.[13] This issue is initially pinpointed by Walker (2021) as a critique against Bao et al. (2020) and it is rooted in the fact that a misreporting incident can endure more than two years on average as described in Section 5.3. Therefore, we address this data leakage issue by discarding all misreporting cases that appear in both training and validating/predicting sets. Cases that present in both validating and predicting sets but not in the training sets are retained. We demonstrate that this data handling can remarkably affect the result interpretation in Section 5.6, raising questions about the validity of certain findings presented in Brown et al. (2020).

***The evaluation metrics***

We follow the current stream of research in misreporting detection and use three metrics to evaluate the performance of LongFinBERT (Beneish, 1997; Larcker and Zakolyukina, 2012; Perols et al., 2017; Bao et al., 2020; Brown et al., 2020). They include the Area Under the Receiver Operating Characteristics, the Normalized Discounted Cumulative Gain at the top $k$%, and the Expected Cost of Misclassification. Evaluation using three different types of metrics is expected to improve the robustness of our results.

The Receiver Operating Characteristics (ROC) curve, plotted in a two-dimensional space, contrasts the true positive rate (on the $y$-axis) with the false positive rate (on the $x$-axis) of a prediction across various probability cutoff thresholds. One typically evaluates a model's predictions by considering the area under the ROC curve, denoted as AUC, with higher values indicating better accuracy. AUC values range from 0 to 1 in which a random prediction yields an AUC of 0.5. An AUC of 0 means the prediction is entirely incorrect, whereas a perfectly correct prediction obtains an AUC of 1. An important feature of AUC is that it is independent of the cutoff threshold choices. This characteristic is valuable as it allows for a comprehensive evaluation of the model's predictive ability without being

[13] For brevity, this issue is termed *data leakage* hereafter.

influenced by specific threshold selections.

Inspired by the rarity of misreporting events, the Normalized Discounted Cumulative Gain at the top $k$%, referred to as NDCG@$k$%, provides a means to assess predictions focusing on the top $k$% of the predicted scores. Specifically, NDCG@$k$% compares the top $k$% of observations ranked by predicted probabilities with an ideally ordered set where all misreporting incidents are included. Ranging from 0 to 1, a higher NDCG@$k$% value indicates that more true misreporting incidents appear within the top-$k$% set ranked based on predicted probabilities. This suggests more precise predictions are being made. This metric is particularly useful when only a small proportion of the population requires inspections, aligning well with the characteristics of misreporting detection. Because the overall misreporting rate is approximately 1% as documented in Table 1, we choose $k = 1$. Additionally, Dechow et al. (2011) classify firms with *F-scores* higher than $95^{th}$ percentile as "high risk", we also select $k = 5$ for comprehensive evaluations.

The final evaluation metric that is used in our paper is the Expected Cost of Misclassification, denoted as ECM, which is introduced by Beneish (1997) and later adopted by Perols et al. (2017). According to Perols et al. (2017), ECM is computed as follow,

$$\text{ECM} = C^{FN} \times P(is_misreporting = 1) \times \frac{n^{FN}}{n^{P}} + C^{FP} \times P(is_misreporting = 0) \times \frac{n^{FP}}{n^{N}},$$

in which $C^{FN}$ and $C^{FP}$ are the normalized estimates of the economic cost of false negative and false positive classifications, correspondingly; $P(is_misreporting = 1)$ is the prior probability of misreporting, $P(is_misreporting = 0) = 1 - P(is_misreporting = 1)$; $n^{FN}$ and $n^{FP}$ are the number of false negative and false positive, determined by the predicted probabilities and a cutoff threshold; $n^{P}$ and $n^{N}$ are the number of misreporting and non-misreporting observations in the prediction set. Different from AUC and NDCG@$k$%, which are purely statistical metrics, ECM takes economic costs of misclassification into account, controlled by the ratio between $C^{FN}$ and $C^{FP}$. Bayley and Taylor (2007) estimate

that the cost caused by overlooking a misreporting firm (false negative) is approximately 20 to 40 times more expensive than that of a false allegation against a non-misreporting firm (false positive). Accordingly, we follow Perols et al. (2017) and choose the cost ratio, $C^{FN} : C^{FP}$, of 30 : 1. ECM also depends on the prior probability of misreporting, $P(is_misreporting = 1)$, which is usually chosen by the unconditional frequency of misreporting. Follow Bell and Carcello (2000), we select $P(is_misreporting = 1) = 0.006$. Finally, ECM also relies on a probability cutoff threshold above which an incident is determined as misreporting. This threshold is optimized during the validation phases (Beneish, 1997; Feroz et al., 2000; Perols and Lougee, 2011; Perols et al., 2017). This, however, introduces more computational burdens and may lead the models to overfit the validation data more easily. Therefore, we compute ECM with the cutoff threshold of 0.6 because Perols et al. (2017) show that this threshold generates decisions that are least sensitive to misclassified labels. ECM is always positive and has upper bounds determined by the misclassification cost ratio. Lower ECM values indicate that the prediction incurs lower costs associated with misclassification. Therefore, as opposed to AUC and NDCG@$k$%, we aim to minimize ECM to achieve the optimal detection models.

***Other remarks***

On top of the three metrics, to obtain even more robust results, we conduct each experiment 500 times and report the kernel densities of the metric values over the 500 runs. We define one complete run as a process including the following steps in order: (i) choosing an evaluation metric; (ii) training, optimizing model hyperparameters, and predicting as described in Figure 5; (iii) compute the metric using the predictions pooled over all prediction years. This approach is inspired by the rarity of the misreporting events (Perols et al., 2017), making model estimation and predictions highly unstable. Different from Brown et al. (2020) who perform bootstrapping on a single prediction, we generate 500 predictions and obtain 500 real values for each evaluation metric per model. Con-

sequently, our interpretation of results relies on the distribution of actual metrics rather than bootstrapped metrics.

Before delving into the empirical results, it is crucial to elaborate on the interpretation of the statistics of the four metrics under consideration, namely AUC, ECM, NDCG@1%, and NDCG@5%. The four metrics are examined via their descriptive statistics as well as their kernel densities obtained from 500 experimental runs. These statistics include the mean, the standard deviation, the minimum, the maximum, 1%-, 50%- and 99%-percentiles, and skewness. For AUC, NDCG@1%, and NDCG@5%, models with higher means, 1%-percentile, median, minimum, and maximum values are favorable. Also with these metrics, models with small skewness, indicating a tendency towards left-skewness in the empirical distributions, suggest a higher probability of surpassing a certain value, thus being desirable. For ECM, the interpretation of these statistics turns upside-down as we seek to minimize ECM instead of maximizing it. A model producing a small standard deviation of the metric values is preferred because it serves as a signal of a stable predictive model. Accordingly, we emphasize the preferred quantities in Table 6 by making them bold for ease of identification. From a visual standpoint, for AUC, NDCG@1%, and NDCG@5%, models with predictions skewed towards the left of the $x$-axis are preferred, while the opposite holds for ECM.

## 5.5 Empirical results

### 5.5.1 Performance of LongFinBERT compared to the 17 financial variables

The descriptive statistics of the metric values obtained from the three models under examination are presented in Table 6. The three models are those that contain (i) only the 17 financial variables, (ii) only the text-based features extracted from LongFinBERT, and (iii) a concatenation of the 17 financial variables and the text-based LongFinBERT fea-

| | AUC | | | ECM | | | NDCG@1% | | | NDCG@5% | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Only Fin. Var. | Only Text | Text and Fin. Var. | Only Fin. Var. | Only Text | Text and Fin. Var. | Only Fin. Var. | Only Text | Text and Fin. Var. | Only Fin. Var. | Only Text | Text and Fin. Var. |
| *Mean* | **0.571** | 0.566 | 0.568 | 0.186 | 0.181 | **0.180** | 0.024 | 0.032 | **0.032** | 0.068 | **0.095** | 0.090 |
| *Std.* | 0.033 | 0.027 | **0.024** | 0.003 | **0.001** | 0.002 | **0.014** | 0.018 | 0.019 | **0.024** | 0.025 | 0.027 |
| *Min* | 0.476 | 0.484 | **0.510** | 0.179 | 0.175 | **0.173** | 0.000 | 0.000 | 0.000 | 0.022 | **0.034** | 0.024 |
| $1\% - percentile$ | 0.488 | 0.497 | **0.516** | - | - | - | 0.003 | 0.003 | **0.003** | 0.031 | **0.042** | 0.037 |
| *Median* | **0.573** | 0.566 | 0.569 | 0.186 | 0.181 | **0.180** | 0.021 | **0.030** | 0.029 | 0.064 | **0.094** | 0.089 |
| $99\% - percentile$ | - | - | - | 0.192 | **0.184** | 0.185 | - | - | - | - | - | - |
| *Max* | **0.644** | 0.638 | 0.638 | 0.193 | **0.184** | 0.187 | 0.092 | 0.102 | **0.112** | 0.153 | 0.162 | **0.168** |
| *Skewness* | **-0.286** | -0.081 | 0.018 | 0.019 | **0.206** | -0.229 | 1.264 | **0.999** | 1.036 | 0.719 | 0.229 | **0.211** |

**Table 3:** This table reports the descriptive statistics of the four metrics, AUC, ECM, NDCG@1%, and NDCG@5% obtained from three different models with 500 runs. The three models are those which contain (i) only 17 financial variables (Only Fin. Var.); (ii) only text-based features extracted from LongFinBERT (Only Text); and (iii) a concatenation of the 17 financial variables and the text-based features from LongFinBERT (Text and Fin. Var.). The bold-faced numbers indicate the best quantities among the three models evaluated by each metric. Certain statistics are omitted (-) as they are either irrelevant or redundant for comparisons with others. The prediction sample spans the period from 1999 to 2018.

tures. We follow the current literature and use the first model, using only the 17 financial variables, as the benchmark (Brown et al., 2020). The comparisons are made using four evaluation metrics. These kernel densities are made using the metric values obtained from 500 repetitions per experimental run as described in Section 5.4. Besides numerical results presented in Table 6, we also provide visual comparisons of the three models using kernel densities in Figure 6.

Despite having the highest average AUC, the model trained solely on financial variables, i.e., the benchmark model, shows an AUC of 0.488 at the 1%-percentile and a minimum of 0.476. These figures suggest that this model fails to beat random guesses at the 1% significance level. Conversely, obtaining an AUC of 0.516 and 0.510 with the same statistics, the model combining both LongFinBERT-based and financial features consistently outperforms random predictions, which the baseline model could fail as described. The performance of the model using only LongFinBERT textual features falls between those of the benchmark model and the combined model. Additionally, it is demonstrated that the LongFinBERT-based features contribute to stabilizing the model performances, evidenced by the smallest standard deviation of the AUC belonging to the combined model. The AUC plot in Figure 6 illustrates the aforementioned analysis. Particularly, the distribution of AUC obtained by the benchmark model showcases the fatter left tail in com-

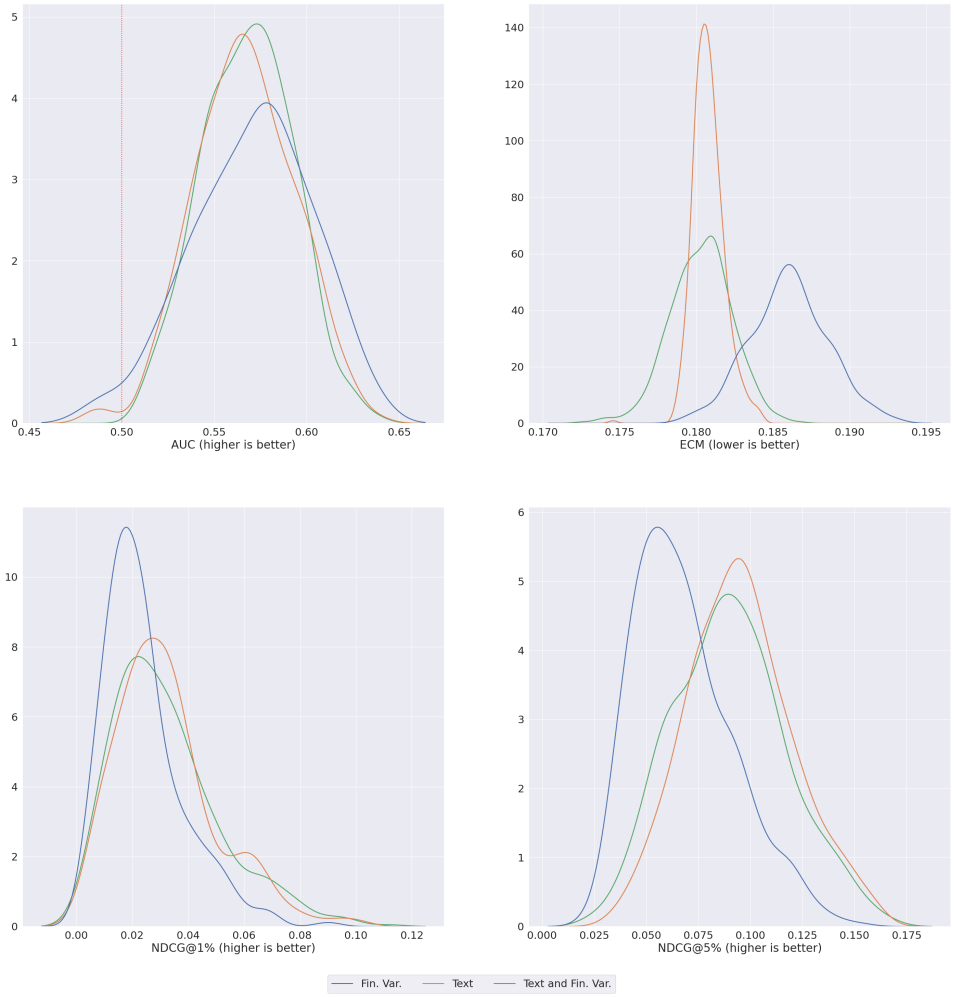



**Figure 6:** This figure presents the kernel densities of the four metrics, i.e., AUC (top-left), ECM (top-right), NDCG@1% (bottom-left), and NDCG@1% (bottom-right), obtained by three models. The three models are those which contain (i) only 17 financial variables (blue), (ii) only text-based features extracted from LongFinBERT (orange), and (iii) a concatenation of the 17 financial variables and the text-based features from LongFinBERT (green), respectively. The vertical red dotted line in the AUC plot implies the AUC value of 0.5 of a random prediction.

parison to the other two densities, implying the possibility for the model to fall below random guesses.

When evaluated by ECM, LongFinBERT strongly demonstrates its superior performance over the benchmark by significantly improving the average ECM upon integration of its text-based features with the financial variables. Particularly, the average ECM enhances from 0.186 of the benchmark to 0.180 of the combined model. This improvement is quantified as twice the magnitude of one standard deviation, calculated from a sample of 500

ECM values derived from the benchmark model. Moreover, we observe favorable statistics across the results of models utilizing LongFinBERT features, whether integrated with financial variables or not. These findings substantiate the conclusion that LongFinBERT enhances significantly the misreporting detection power, as measured by ECM. The results are easily noticeable from the ECM plot depicted in Figure 6, where the ECM distributions of both the text-only and combined models are distinctly positioned to the left, setting them apart from the ECM distribution of the benchmark model.

The superior performance of LongFinBERT still holds when assessing the models using NDCG@$k$%. The models using LongFinBERT features achieve better NDCG@1% and NDCG@5% on average compared to the benchmark models. The kernel densities of NDCG@1% and NDCG@5% in Figure 6 possess more mass on their right tails and are also more right-skewed. Recall that NDCG@$k$% is maximized for the optimal models, these results suggest that models using LongFinBERT text-based features have a higher likelihood of producing high NDCG@$k$% values with $k \in \{1, 5\}$, thus being favorable.

Overall, we have shown that text-based features extracted from LongFinBERT enhance the efficacy of the existing benchmark set of financial variables proposed by Dechow et al. (2011) in detecting misreporting incidents. Notably, the text-based features from LongFinBERT exhibit comparable or superior performance compared to the 17 financial variables. The contribution made by LongFinBERT remains consistent and robust across different choices of evaluation metrics.

### 5.5.2 Performance of LongFinBERT compared to other text models

So far, LongFinBERT has been demonstrated to boost the misreporting detection capability across various evaluation metrics. The next question that naturally arises is: "How does LongFinBERT perform in comparison with other text models in misreporting de-

tection?". This section attempts to provide an answer to this question by comparing LongFinBERT with LDA, Doc2Vec, and FinBERT models in the misreporting detection task. These models are selected as representative examples of each model genre discussed in Section 2, and also because of their popular applications in accounting studies. In general, we compare two types of models: (i) those trained with only text-based features derived by these models (Figure 7); and (ii) those integrating both text-based features and the 17 financial variables (Figure 8).

It is important to note that FinBERT struggles to process an entire 10-K file at once due to its quadratic complexity as discussed in Section 2, thereby requiring truncation. Accordingly, we explore two truncation methods when applying FinBERT to process 10-K filings: (i) retaining only the first 8,196 tokens; and (ii) retaining only the last 8,196 tokens of each 10-K document. Different from FinBERT, which offers readily available pretrained model weights, LDA and Doc2Vec require training from scratch before extracting their text-based features of 10-K filings. Further details on their training procedures are provided in Appendix B.

In general, LongFinBERT (marked by the blue curves) produces text-based features that outperform those extracted from other text models, both when used alone and integrated with the 17 financial variables. Notably, among the models with solely text-based features, LongFinBERT is the only text model yielding predictions that significantly beat the random guesses at the 5% significance level. The superior performance of LongFinBERT compared to LDA and Doc2Vec aligns with our expectations due to the powerful linguistic capability of LMs over non-LM models.

However, FinBERT models with truncated documents exhibit relatively weak performance. We conjecture that this result stems from the limitation of FinBERT's ability to process the whole 10-K filings, rather than from its linguistic capabilities. Consequently, FinBERT generates document embeddings that capture incomplete information. This re-

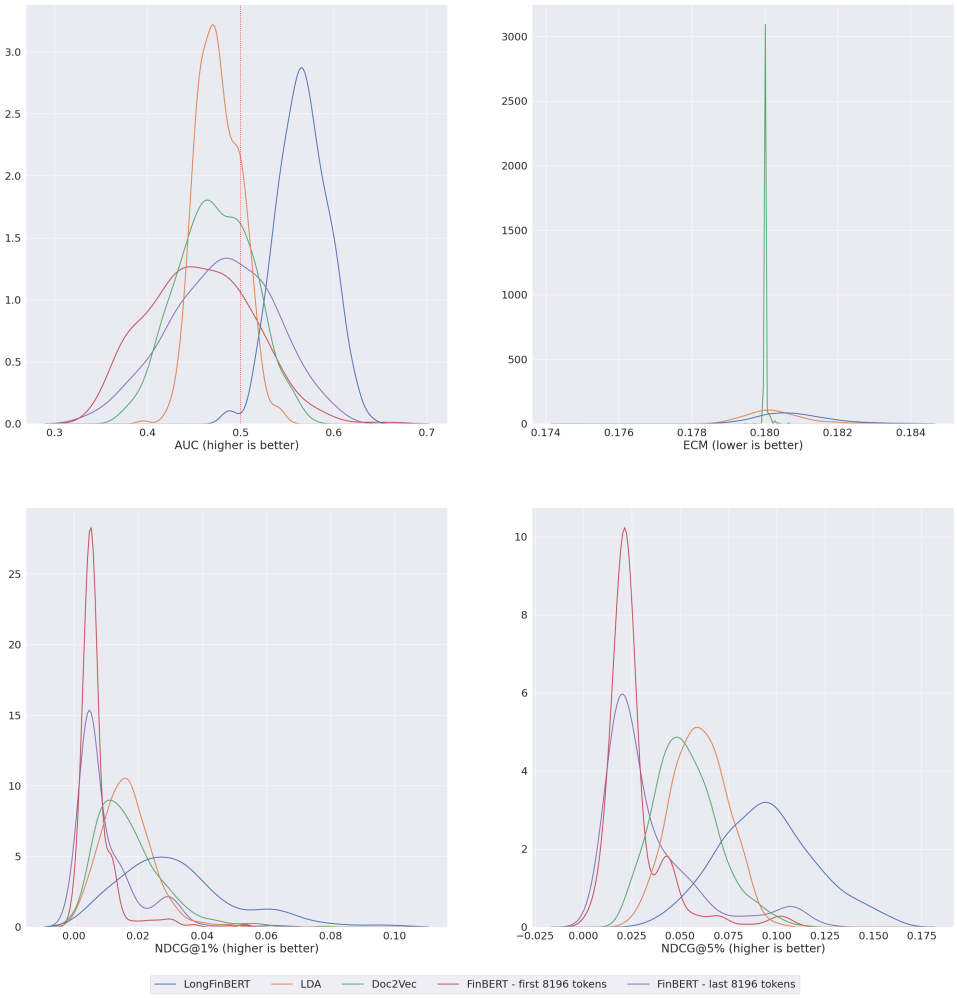



**Figure 7:** This figure presents the kernel densities of the four metrics, i.e., AUC (top-left), ECM (top-right), NDCG@1% (bottom-left), and NDCG@1% (bottom-right), obtained by five models trained with only the corresponding text-based features. The five models include LongFinBERT (blue), LDA (orange), Doc2Vec (green), FinBERT with the first 8,196 tokens (red), and FinBERT with the last 8,196 tokens (violet). The vertical red dotted line in the AUC plot implies the AUC value of 0.5 of a random prediction.

sult highlights the importance of processing the whole document to detect misreporting utilizing 10-K filings, thereby underscoring the advantage of LongFinBERT over the text models, especially with LMs, in processing extremely lengthy accounting disclosures.

In summary, together with Section 5.5.1, our analyses demonstrate the superior performance of LongFinBERT over both financial metrics and alternative text models in detecting misreporting. This finding holds across numerous evaluation metrics, affirming its robust performance.

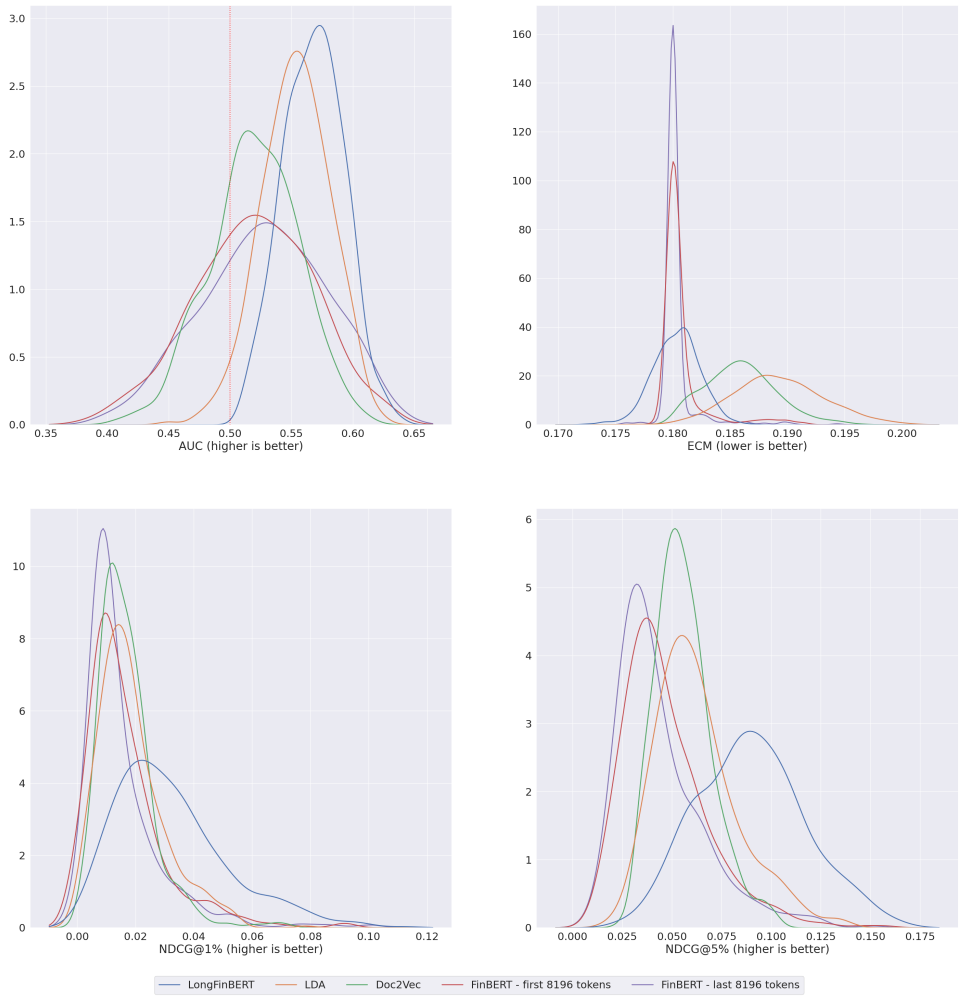



**Figure 8:** This figure presents the kernel densities of the four metrics, i.e., AUC (top-left), ECM (top-right), NDCG@1% (bottom-left), and NDCG@1% (bottom-right), obtained by five models trained with the combination of the 17 financial variables and the corresponding text-based features. The five models include LongFinBERT (blue), LDA (orange), Doc2Vec (green), FinBERT with the first 8, 196 tokens (red), and FinBERT with the last 8, 196 tokens (violet). The vertical red dotted line in the AUC plot implies the AUC value of 0.5 of a random prediction.

## 5.6 Discussion of the data leakage problem

In this section, we discuss the data leakage problem mentioned in Section 5.4. To recall, a misreporting incident likely persists for many years. This imposes a challenge in rolling window time series research designs, where the training data directly precedes the validation/prediction data. In such scenarios, a misreporting case may extend across both the training and prediction datasets. Consequently, models that are trained on the training data are also informed about the information in the prediction data, hence mak-

| *Year (in prediction)* | Number of misreporting cases | | | | | |
|---|---|---|---|---|---|---|
| | Panel A: No delay prediction | | | Panel B: Delay prediction | | |
| | Training | Validating/ Predicting | Leaked cases | Training | Validating/ Predicting | Leaked cases |
| 1999 | 47 | 57 | 19 | | | |
| 2000 | 65 | 63 | 24 | | | |
| 2001 | 85 | 68 | 26 | 47 | 68 | 11 |
| 2002 | 104 | 64 | 26 | 65 | 64 | 12 |
| 2003 | 127 | 54 | 28 | 85 | 54 | 11 |
| 2004 | 142 | 50 | 31 | 104 | 50 | 10 |
| 2005 | 153 | 40 | 25 | 127 | 40 | 15 |
| 2006 | 161 | 34 | 21 | 142 | 34 | 16 |
| 2007 | 168 | 24 | 12 | 153 | 24 | 6 |
| 2008 | 174 | 19 | 9 | 161 | 19 | 4 |
| 2009 | 180 | 18 | 9 | 168 | 18 | 3 |
| 2010 | 184 | 18 | 8 | 174 | 18 | 4 |
| 2011 | 189 | 17 | 8 | 180 | 17 | 4 |
| 2012 | 194 | 20 | 8 | 184 | 20 | 2 |
| 2013 | 198 | 15 | 4 | 189 | 15 | 0 |
| 2014 | 206 | 12 | 5 | 194 | 12 | 0 |
| 2015 | 209 | 11 | 5 | 198 | 11 | 1 |
| 2016 | 213 | 9 | 3 | 206 | 9 | 0 |
| 2017 | 215 | 11 | 4 | 209 | 11 | 0 |
| 2018 | 219 | 7 | 4 | 213 | 7 | 1 |

**Table 4:** This table reports the number of misreporting cases appearing in the training, and validating/predicting sets in two setups: (i) the validating/predicting set immediately follows the training set (i.e., Panel A: No delay prediction); and (ii) the validating/predicting set is delayed by two years after the training set (i.e., Panel B: Delay prediction). The leftmost column presents the years in the predicting sets. Columns *Training* (*Validating/Predicting*) present the number of misreporting cases appearing in the training (validating/predicting) sets in order to make predictions of misreporting in the corresponding years. The *Overlapping* columns present the number of misreporting cases appearing in both training and validating/prediction sets in a corresponding year, thereby being considered as *leaked cases*.

ing the predictions spuriously optimistic. We refer to this phenomenon as *data leakage* and the misreporting cases spanning both training and prediction data as *leaked misreporting cases*. We reckon that a discussion of the data leakage issue is crucial to obtaining a correct research design, especially in research using machine learning. Research designs overlooking this issue potentially generate interpretations under spuriously optimistic results, thereby driving the conclusions astray.

Panel A of Tabel 4 provides an overview of the distribution of the misreporting cases across the years during prediction. The interpretation of the table is as follows: We use the data from 1994 to 1997, which has 47 misreporting cases, to train a model. The model,

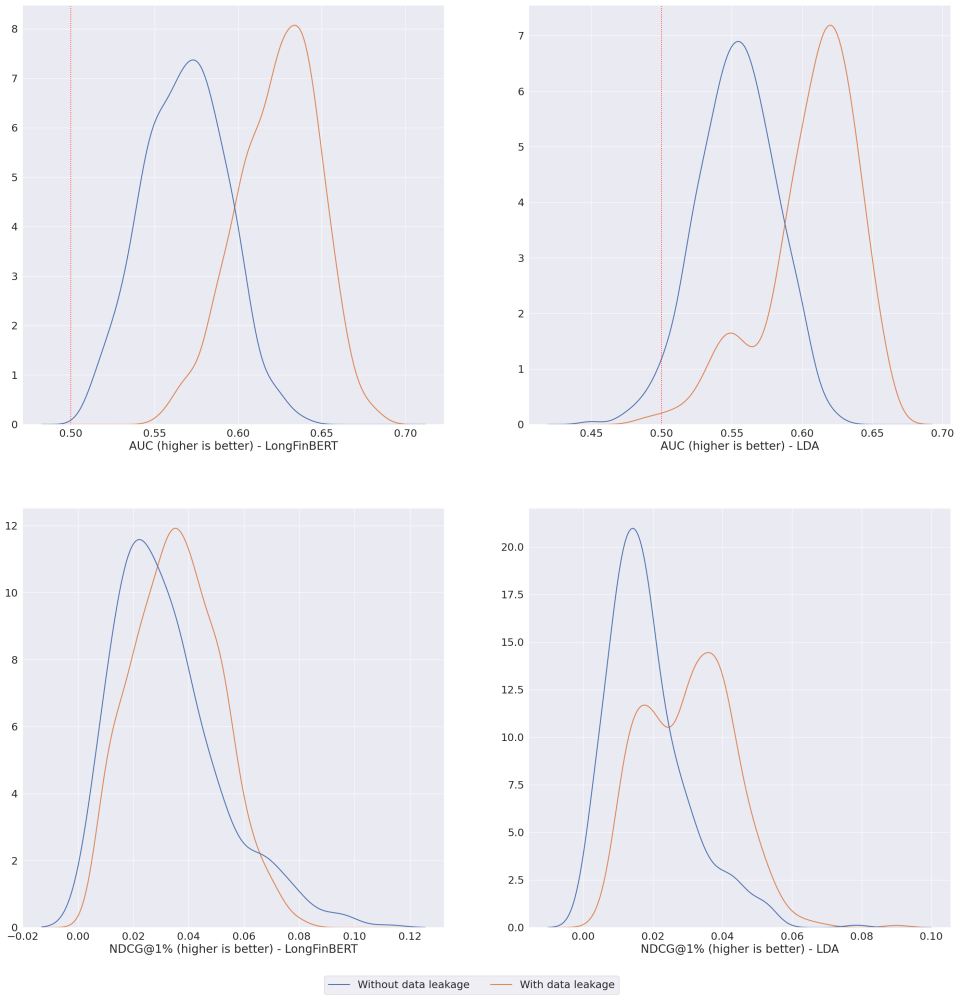



**Figure 9:** This figure illustrates the model performance between two experimental settings: (i) leaked misreporting cases removed (blue), and (ii) leaked misreporting cases retained (orange). We use the combination of the 17 financial variables and the text-based features extracted from LongFinBERT and LDA to train the boosting algorithm. The distributions are aggregated over 500 runs of each setup. The prediction data sample spans the period from 1999 to 2018.

in turn, is validated with the data in 1998, and used to detect misreporting with the data in 1999. The combined validating and predicting datasets contain 57 misreporting cases, with 19 cases overlapping between the training and validating/predicting sets. Overall, leaked cases in this setting account for up to 45.66% of the total misreporting cases in the prediction set. This substantial proportion of leaked cases potentially distorts the result interpretations under inappropriate controls.

We focus on AUC and NDCG@1% to investigate the effects of the data leakage issue on

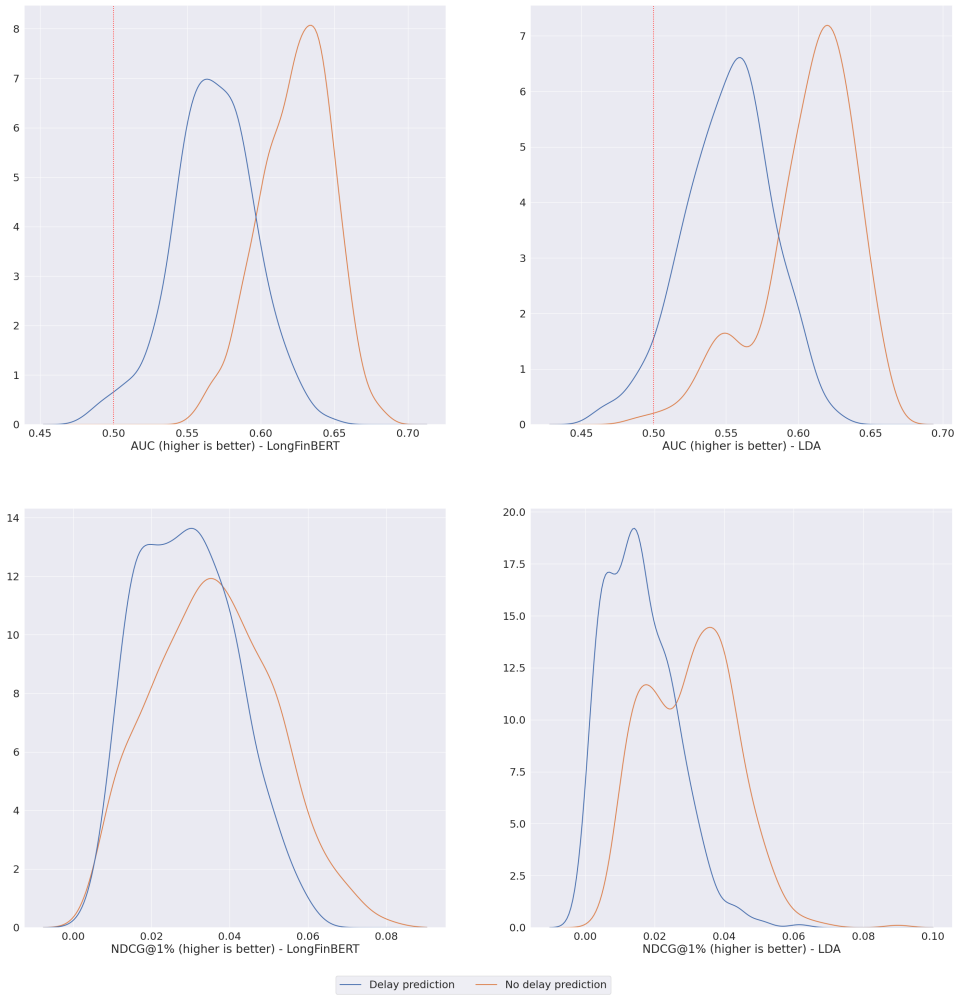



**Figure 10:** This figure illustrates the model performance between two experimental settings: (i) with a two-year delay in prediction (blue), and (ii) without delay in prediction (orange). We use the combination of the 17 financial variables and the text-based features extracted from LongFinBERT and LDA to train the boosting algorithm. The distributions are aggregated over 500 runs of each setup. The prediction data sample spans the period from 1999 to 2018.

result interpretations because they are the main metrics utilized by Brown et al. (2020). We train boosting algorithms using the combination of the 17 financial variables and text-based features from LongFinBERT (our model) and LDA (employed by Brown et al. (2020)). First, we compare the metrics obtained by two setups: (i) leaked misreporting cases removed (i.e., data leakage controlled), and (ii) leaked misreporting cases retained (i.e., data leakage uncontrolled). The results of this analysis are presented in Figure 9. Our results indicate that retaining leaked cases is associated with better metric values, consistently across both AUC and NDCG@1%. Moreover, the removal of leaked cases

appears to have a more pronounced impact on AUC compared to NDCG@1%.

One might argue that the shift in the metric kernel densities occurs due to a change in the distribution of misreporting incidents in the predicting set when leaked cases are discarded, rather than solely as a consequence of the data leakage phenomenon. In response to the argument, we establish the second experiment in which we control the data leakage issue by delaying the prediction by two years for each recursive window. Particularly, in Figure 5, we extend window 1 in the sense that the training horizon remains unchanged but we make validation and prediction in 2000 and 2001, correspondingly. Similar extensions are applied to the consecutive windows. Accordingly, we do not discard the leaked cases to focus on the sole effects of the delayed prediction on model performances.

Panel B of Table 4 reports the distributions of leaked cases under the delay-prediction setting. Although the number of misreporting cases in the training set is shifted by two years, the count in the validating/predicting set remains unchanged. Due to the two-year delay in prediction, the number of leaked cases declines remarkably, now accounting for approximately 20.37% of total misreporting cases in the prediction set.

Figure 10 compares the performance of models under two settings: (i) delay misreporting detection by two years (i.e., data leakage controlled); and (ii) no delay misreporting detection (i.e., data leakage uncontrolled). Similar to the setting of leaked-case removal, delayed prediction weakens model performance. Specifically, when model validation and prediction are postponed for two years after the training dataset, the kernel densities of AUC and NDCG@1% shift to the left. In other words, the absence of delaying prediction is connected with an increase in model performance across the two metrics utilized by Brown et al. (2020).

Overall, we demonstrate that not controlling for data leakage is associated with skeptically optimistic model performance evaluated by AUC and NDCG@1%. We do that

by discarding leaked misreporting cases or by delaying prediction. While we retain any conclusions of misleading result interpretation under data leakage, our discussions raise concerns about obtaining a proper research design for research in misreporting detection.

# 6 Year-over-year 10-K similarity and stock market reaction

In this section, using LongFinBERT, we revisit the connection between the year-over-year similarity of 10-K filings and the associated market reaction, as explored by Brown and Tucker (2011). Specifically, Brown and Tucker (2011) employ a bag-of-words method to gauge the cosine similarity between the Management's Discussion and Analysis (MD&A) sections in the 10-K filings of the same firms across two consecutive years. Consequently, they observe a negative relationship between the MD&A section's similarity and stock market reaction, yet emphasizing that this effect has been eroded over time. With the introduction of LongFinBERT, we are now able to revise this empirical question under the lens of a more state-of-the-art textual analysis approach, thus potentially uncovering more insights than the traditional bag-of-words methods.

To this end, we first utilize LongFinBERT to derive the numerical representations of the 10-K filing and the corresponding MD&A section of firm $i$ in year $t$, denoted by $w_{i,t}^{10K}$ and $w_{i,t}^{MDA}$, respectively. We then gauge the year-over-year similarity between two 10-K filings or MD&A sections by computing the cosine similarity between $w_{i,t}^{k}$ and $w_{i,t-1}^{k}$ with $k \in \{10K, MDA\}$. Technically,

$$Sim_{i,t}^{k} = \frac{\langle w_{i,t}^{k}, w_{i,t-1}^{k} \rangle}{\|w_{i,t}^{k}\|.\|w_{i,t-1}^{k}\|},$$

in which $\langle x, y \rangle$ is the inner product of two vectors $x$ and $y$; and $\|x\|$ is the L2 norm of

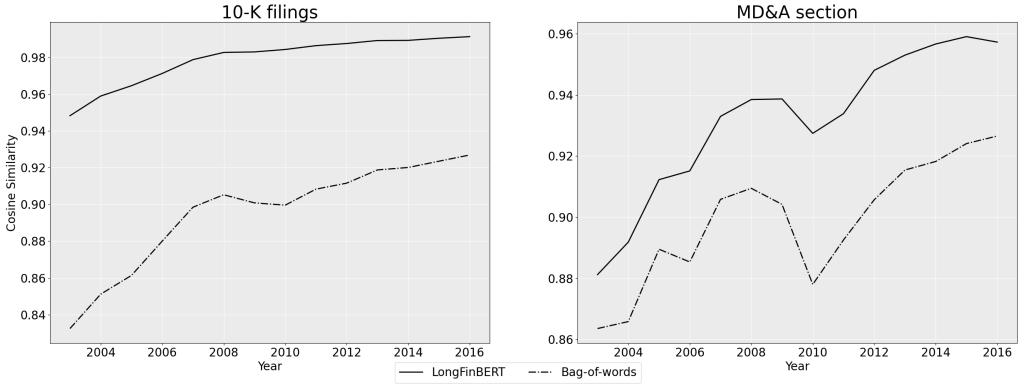



**Figure 11:** This figure illustrates the evolution of the aggregated year-over-year cosine similarity of 10-K filings (blue) and the corresponding MD&A section (orange) over time. These curves present the averages of firm-wise year-over-year similarities for the corresponding documents each year. Grey areas denote the 95% confidence intervals of the point estimates for the means of the year-over-year similarities. The data sample spans the period from January 2002 to December 2016.

vector $x$.[14] According to Brown and Tucker (2011), $Sim^k_{i,t}$, by capturing the linguistic similarity between the accounting disclosures of a firm in two consecutive years, is a good proxy for informational similarity in these documents. In addition to LongFinBERT, we also compute the similarity using the Bag-of-words model of Brown and Tucker (2011) as a reliable reference; see, accordingly, Brown and Tucker (2011) for detailed instructions on representing documents using the Bag-of-words model.

Figure 11 presents the evolution of the aggregated year-over-year similarity of 10-K filings and the corresponding MD&A section over time using LongFinBERT and the Bag-of-words model. Consistent with Brown and Tucker (2011), we observe an increasing trend of the year-over-year similarity of both 10-K filings and the MD&A section over time, showcased by both LongFinBERT and the Bag-of-words model. The MD&A section exhibits lower year-over-year similarity than the entire 10-K filings. This finding is in line with Dyer et al. (2017)'s results and our expectation that 10-K filings consist of more boilerplate and generic sections such as business description (Item 1) or risk factors (Item 1A). Therefore, 10-K filings are less prone to modifications compared to the MD&A sec-

[14]Given two $d$-dimensional vectors $x = (x_1, x_2, ..., x_d)^\top$ and $y = (y_1, y_2, ..., y_d)^\top$, the inner product of $x$ and $y$ is defined as $\langle x, y \rangle = x_1 y_1 + x_2 y_2 + ... + x_d y_d$; the L2 norm of $x$ is $\|x\| = \sqrt{x_1^2 + x_2^2 + ... + x_d^2}$.

tion, where firm managers express their assessments and viewpoints regarding the firm's operations in the current fiscal year, thus exhibiting greater variability on an annual basis. To exemplify this point, we provide a comparison of Apple Inc.'s (CIK: 0000320193) 10-K filings for 2015 and 2016 in Appendix D. Notably, the year-over-year similarity in the MD&A section decreases following the financial crisis, indicating that firm managers tend to substantially modify the language used in this section between 2009 and 2010. Recall that the disclosures in 2009 (2010) discuss the firm's operations in 2008 (2009), this result is consistent with the contemporaneous unstable business conditions.

We now examine the relationship between the magnitude of management disclosures' modifications, measured by either LongFinBERT and the Bag-of-words model, and market reaction. To this end, we adopt the regression study designed by Brown and Tucker (2011) with small modifications. In particular, we use the absolute value of the cumulative abnormal returns around the 10-K filing dates to capture the market reaction to the filing. The cumulative abnormal return, $CAR_{i,t,[0\to3]}$, is based on the Fama-French three-factor model and cumulated from the filing date (inclusively) to 3 days later as in Jegadeesh and Wu (2013) and Jiang et al. (2019). We then regress $|CAR_{i,t,[0\to3]}|$ on $Sim^{k}_{i,t}$ and a set of control variables as follows,

$$|CAR_{i,t,[0\to3]}| = \alpha + \beta Sim^{10K}_{i,t} + \gamma Sim^{MDA}_{i,t} + Controls_{i,t} + \epsilon_{i,t}, \tag{1}$$

where *Controls* includes the market capitalization ($\log(Size)$), the additional financial information measured by new items in the Compustat database ($New_items$), the dummy variable on filing delay ($Late_filing$), and the magnitude of market reaction before the filing date ($|CAR_{i,t,[-22\to-1]}|$, $|CAR_{i,t,[-10\to-1]}|$ and $|CAR_{i,t,[-5\to-1]}|$). It is worth mentioning that we can include both $Sim^{10K}_{i,t}$ and $Sim^{MDA}_{i,t}$ in the regression model due to their low correlations of 0.197 with LongFinBERT and 0.103 using the Bag-of-words method. Hypothetically, a greater (smaller) value of $Sim^{k}_{i,t}$ implies fewer (more) modifications in

| | LongFinBERT | | | Bag-of-words | | |
|---|---|---|---|---|---|---|
| | (1) | (2) | (3) | (4) | (5) | (6) |
| $Sim^{MDA}$ | -0.001* | | -0.001** | -0.001* | | -0.001** |
| | (-1.697) | | (-1.995) | (-1.952) | | (-2.012) |
| $Sim^{10K}$ | | 0.000 | 0.001 | | -0.000 | -0.000 |
| | | (1.062) | (1.390) | | (-0.764) | (-0.648) |
| $\log(Size)$ | -0.010*** | -0.010*** | -0.010*** | -0.010*** | -0.010*** | -0.010*** |
| | (-8.004) | (-8.006) | (-8.007) | (-7.997) | (-7.996) | (-7.991) |
| $New_items$ | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 | 0.004 |
| | (1.341) | (1.340) | (1.340) | (1.338) | (1.340) | (1.338) |
| $Late_filing$ | 0.004** | 0.004** | 0.004** | 0.004** | 0.004* | 0.004* |
| | (2.285) | (2.294) | (2.234) | (2.316) | (2.265) | (2.320) |
| $\lvert CAR_{[-5,-1]}\rvert$ | 0.083*** | 0.083*** | 0.083*** | 0.083*** | 0.083*** | 0.083*** |
| | (12.18) | (12.23) | (12.20) | (12.13) | (12.18) | (12.11) |
| $\lvert CAR_{[-10,-1]}\rvert$ | 0.005*** | 0.005*** | 0.005*** | 0.005*** | 0.005*** | 0.005*** |
| | (3.308) | (3.337) | (3.297) | (3.295) | (2.437) | (3.289) |
| $\lvert CAR_{[-22,-1]}\rvert$ | 0.024*** | 0.024*** | 0.024*** | 0.024*** | 0.024*** | 0.024*** |
| | (18.13) | (18.29) | (18.19) | (18.09) | (18.22) | (18.07) |
| $No.\ Obs$ | 33,049 | 33,049 | 33,049 | 33,049 | 33,049 | 33,049 |
| $R^2$ | 0.031 | 0.031 | 0.031 | 0.031 | 0.0.31 | 0.031 |

**Table 5:** This table reports the regression results of Equation (1) across three different settings, i.e., including either $Sim^{MDA}$ or $Sim^{10K}$ or both. $Sim^{MDA}$ and $Sim^{10K}$ are computed using either LongFinBERT (columns 1, 2, and 3) or Brown and Tucker (2011)'s Bag-of-words model (columns 4, 5, and 6). The dependent variable is the absolute value of the abnormal stock returns based on the Fama-French asset pricing model cumulated from the filing dates to $h$ days later. Inspired by Brown and Tucker (2011), we control for firm size ($\log(Size)$), additional financial information ($New_items$), dummy on late filing ($Late_filing$), and past market reactions ($|CAR_{[-5,-1]}|$, $|CAR_{[-10,-1]}|$, and $|CAR_{[-22,-1]}|$). The intercept ($\alpha$), firm-fixed effects, and time-fixed effects are also included in the regression equation. The coefficients, Newey-West heteroskedasticity- and autocorrelation-robust $t$-statistics (in parentheses), the number of observations, and $R^2$ are reported. The data sample spans the period from January 2002 to December 2016. *, **, and *** denote significance at the 10%, 5%, and 1% levels, respectively.

the management disclosures, thus potentially associated with weaker (stronger) market reactions toward the disclosure releases. Accordingly, $\beta$ and $\gamma$ are expected to be negative.

We use the sample of 10-K filings and MD&A sections from January 2002 and December 2016. Instead of extracting from 10-K filings, we use the dataset of MD&A sections compiled by Ewens et al. (2023), which can be downloaded from the Caltech Database.[15] We use the Compustat, Audit Analytics Late Filers, and Wharton Research Data Service Beta

[15] Available in: https://data.caltech.edu/records/28m72-ysc35

Suite databases to compute the cumulative abnormal returns and the set of control variables. After matching the text-based and accounting/financial datasets, we obtain 33,049 firm-year observations from 4,770 unique companies spanning 14 years.

Table 5 reports the regression results of Equation (1) with market reaction over the three-day horizon across three different settings: (i) solely including $Sim^{MDA}$; (ii) solely including $Sim^{10K}$; and (iii) including both $Sim^{MDA}$ and $Sim^{10K}$. The year-over-year similarities are computed using either LongFinBERT or the Brown and Tucker (2011)'s Bag-of-words model. As anticipated, the stock market reacts significantly to the year-over-year similarities of the MD&A section of 10-K filings. Moreover, the negative coefficients over the regressions imply that more modifications in the MD&A section (low $Sim^{MDA}$) are associated with a higher magnitude of market reaction. Given more up-to-date data samples from 2002 to 2016, we find that the effect of MD&A modifications measured using LongFinBERT on the stock market remains pronounced in recent periods, contradictory to Brown and Tucker (2011). The relation between the year-over-year similarity of 10-K filings and market reaction, however, is relatively weak. Intuitively, while the modifications in the MD&A section attract more attention from investors, those of 10-K filings are usually overlooked. This finding aligns with the current literature documenting the importance of the MD&A section in 10-K filings (Tavcar, 1998; SEC, 2003; Li, 2010; Muslu et al., 2015).

In general, by using LongFinBERT, we demonstrate an increasing trend in the year-over-year similarity of the MD&A section in 10-K filings, aligning with Brown and Tucker (2011). However, while Brown and Tucker (2011) conclude that investors have become indifferent to the MD&A's modifications, our analysis indicates the contrary. This result appears robust to the text models, i.e., LongFinBERT and the Bag-of-words model.

# 7 Conclusion

Despite the current advancements in natural language processing (NLP), the application of state-of-the-art language models (LMs) in economic research remains limited. This is mainly rooted in the substantial computational demands of LMs, especially, applied to financial documents, which are typically lengthy.

Aiming to bridge that gap, we introduce LongFinBERT, a language model (LM) specifically designed for lengthy financial documents. In particular, LongFinBERT adopts the novel dilated attention mechanism developed by Ding et al. (2023). This mechanism allows LongFinBERT to maintain interactions between tokens distant from each other while ensuring sparse attention matrices. Consequently, that allows LongFinBERT to efficiently tackle the computational hurdles of LMs applied to long documents while maintaining the power of LMs in capturing complex linguistic structures. We adapt LongFinBERT to the financial context by pre-training it with a large corpus of 10-K filings.

We demonstrate LongFinBERT's superiority over other text models from two angles: computational efficiency and the ability to extract meaningful textual features. First, we design a simulation study to compare LongFinBERT with four other LMs in handling long documents based on their processing time and memory usage. The results show that LongFinBERT exhibits significantly lower demands in both time and memory than the competing LMs in processing extensively long documents. Notably, LongFinBERT showcases a five-fold improvement in computational demands over LongFormer, another LM specialized for long documents.

Second, to assess the ability of LongFinBERT to extract meaningful textual features for economic applications, we employ LongFinBERT for misreporting detection using 10-K filings, which are known for their extensive length. We find that the textual features extracted from LongFinBERT enhance the ability to detect misreporting of Dechow et al.

(2011)'s financial variables. Furthermore, LongFinBERT outperforms other text models in detecting misreporting incidents. The results remain consistent over various evaluation metrics commonly used in the misreporting detection literature. Coupled with its computational superiority, LongFinBERT has showcased its efficiency in processing financial documents.

Third, we apply LongFinBERT to examine the year-over-year similarities of 10-K filings and the corresponding Management's Discussion and Analysis (MD&A) section. In line with the current literature, we reveal a trend towards increased similarity over time in both disclosures, with 10-K filings exhibiting greater similarity than the MD&A section. Complementary to the extant literature, we document a significant relationship between the market reaction and the modifications of the MD&A section but not 10-K filings. This result suggests that investors may respond more sensitively to alterations within the MD&A section, reaffirming the importance of the MD&A section in 10-K filings.

A potential limitation of LongFinBERT, however, resides in the interpretations of its extracted textual features, a challenge shared by many LMs. Nevertheless, we anticipate that LongFinBERT can be effectively utilized to extract interpretable topics from financial documents, which currently rely on the word-count-based Latent Dirichlet Allocation model. With the demonstrated linguistic capability, LongFinBERT is expected to produce more accurate topical information in economic and financial documents.

**Authors' contributions**

Tri Minh Phan = 1, Erik-Jan Senn = 2.

- LongFinBERT (idea and implementation): 1;
- Computational comparison study (idea and implementation): 1;
- Misreporting detection study:

  - Idea: 1, 2;
  - Implementation: 1;
  - Result analysis: 1, 2;
- Stock market reaction study (idea and implementation): 1;
- Data collection: 1;
- Writing: 1;
- Editing: 1, 2.

All authors reviewed the results and approved the final version of the manuscript.

# References


Ash, E. and Hansen, S. (2023). Text algorithms in economics, *Annual Review of Economics* **15**: 659–688.

Bae, J., Yu Hung, C. and van Lent, L. (2023). Mobilizing text as data, *European Accounting Review* **32**(5): 1085–1106.

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

Bayley, L. and Taylor, S. (2007). Identifying earnings management: A financial statement analysis (red flag) approach, *Proceedings of the American Accounting Association Annual Meeting, Sarasota, FL: AAA*.

Bell, T. B. and Carcello, J. V. (2000). A decision aid for assessing the likelihood of fraudulent financial reporting, *Auditing: A Journal of Practice & Theory* **19**(1): 169–184.

Beltagy, I., Peters, M. E. and Cohan, A. (2020). Longformer: The long-document transformer, *arXiv preprint arXiv:2004.05150* .

Beneish, M. D. (1997). Detecting GAAP violation: Implications for assessing earnings management among firms with extreme financial performance, *Journal of Accounting and Public Policy* **16**(3): 271–309.

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

Bertomeu, J., Cheynel, E., Floyd, E. and Pan, W. (2021). Using machine learning to detect misstatements, *Review of Accounting Studies* **26**: 468–519.

Bochkay, K., Brown, S. V., Leone, A. J. and Tucker, J. W. (2023). Textual analysis in accounting: What's next?, *Contemporary Accounting Research* **40**(2): 765–805.

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

Brown, S. V. and Tucker, J. W. (2011). Large-sample evidence on firms' year-over-year MD&A modifications, *Journal of Accounting Research* **49**(2): 309–346.

Cao, S., Jiang, W., Yang, B. and Zhang, A. L. (2023). How to talk when a machine is listening: Corporate disclosure in the age of AI, *The Review of Financial Studies* **36**(9): 3603–3642.

Cazier, R. A. and Pfeiffer, R. J. (2016). Why are 10-K filings so long?, *Accounting Horizons* **30**(1): 1–21.

Chen, C. Y.-H., Fengler, M. R., Härdle, W. K. and Liu, Y. (2022). Media-expressed tone, option characteristics, and stock return predictability, *Journal of Economic Dynamics and Control* **134**: 104290.

Chen, M. A., Wu, Q. and Yang, B. (2019). How valuable is FinTech innovation?, *The Review of Financial Studies* **32**(5): 2062–2106.

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system, *Proceedings of the 22nd ACM KDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Cho, H. and Muslu, V. (2021). How do firms change investments based on MD&A disclosures of peer firms?, *The Accounting Review* **96**(2): 177–204.

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

Dechow, P. M., Sloan, R. G. and Sweeney, A. P. (1996). Causes and consequences of earnings manipulation: An analysis of firms subject to enforcement actions by the SEC, *Contemporary Accounting Research* **13**(1): 1–36.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Long and Short Papers)*, Vol. 1, Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186.

Ding, J., Ma, S., Dong, L., Zhang, X., Huang, S., Wang, W. and Wei, F. (2023). LongNet: Scaling transformers to 1,000,000,000 tokens, *Proceedings of the 10th International Conference on Learning Representations*.

Dyer, T., Lang, M. and Stice-Lawrence, L. (2017). The evolution of 10-K textual disclosure: Evidence from Latent Dirichlet Allocation, *Journal of Accounting and Economics* **64**(2-3): 221–245.

Ewens, M., Peters, R. and Wang, S. (2023). Measuring intangible capital with market prices, *Working Paper* .

Fengler, M. and Phan, T. M. (2023). *A Topic Model for 10-K Management Disclosures*, School of Economics and Political Science, Department of Economics.

Feroz, E. H., Kwon, T. M., Pastena, V. S. and Park, K. (2000). The efficacy of red flags in predicting the sec's targets: an artificial neural networks approach, *Intelligent Systems in Accounting, Finance & Management* **9**(3): 145–157.

Frankel, R., Jennings, J. and Lee, J. (2022). Disclosure sentiment: Machine learning vs. dictionary methods, *Management Science* **68**(7): 5514–5532.

Freund, Y., Schapire, R. E. et al. (1996). Experiments with a new boosting algorithm, *the Thirteenth International Conference on Machine Learning*, Vol. 96, Citeseer, pp. 148–156.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine, *Annals of Statistics* pp. 1189–1232.

Friedman, J. H. (2002). Stochastic gradient boosting, *Computational Statistics & Data Analysis* **38**(4): 367–378.

Gentzkow, M., Kelly, B. and Taddy, M. (2019). Text as data, *Journal of Economic Literature* **57**(3): 535–574.

Griffin, P. A. (2003). Got information? Investor response to form 10-K and form 10-Q EDGAR filings, *Review of Accounting Studies* **8**: 433–460.

Harris, Z. S. (1954). Distributional structure, *Word* **10**(2-3): 146–162.

He, P., Liu, X., Gao, J. and Chen, W. (2021). Deberta: Decoding-enhanced bert with disentangled attention, *International Conference on Learning Representations*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W. (2022). LoRA: Low-rank adaptation of large language models, *International Conference on Learning Representations*.

Huang, A. H., Wang, H. and Yang, Y. (2023). Finbert: A large language model for extracting information from financial text, *Contemporary Accounting Research* **40**(2): 806–841.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. and Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2704–2713.

Jegadeesh, N. and Wu, D. (2013). Word power: A new approach for content analysis, *Journal of Financial Economics* **110**(3): 712–729.

Jiang, F., Lee, J., Martin, X. and Zhou, G. (2019). Manager sentiment and stock returns, *Journal of Financial Economics* **132**(1): 126–149.

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

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents, *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32 of *Proceedings of Machine Learning Research*, PMLR, Bejing, China, pp. 1188–1196.

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

Li, F. (2010). The information content of forward-looking statements in corporate filings—a naïve Bayesian machine learning approach, *Journal of Accounting Research* **48**(5): 1049–1102.

Li, K., Mai, F., Shen, R. and Yan, X. (2021). Measuring corporate culture using machine learning, *The Review of Financial Studies* **34**(7): 3265–3315.

Linnainmaa, S. (1976). Taylor expansion of the accumulated rounding error, *BIT Numerical Mathematics* **16**(2): 146–160.

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

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

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

Loukas, L., Fergadiotis, M., Chalkidis, I., Spyropoulou, E., Malakasiotis, P., Androutsopoulos, I. and Paliouras, G. (2022). FiNER: Financial numeric entity recognition for XBRL tagging, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Dublin, Ireland, pp. 4419–4431.

Lusa, L. et al. (2017). Gradient boosting for high-dimensional prediction of rare events, *Computational Statistics & Data Analysis* **113**: 19–37.

Manning, C. and Schutze, H. (1999). *Foundations of statistical natural language processing*, MIT press.

Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Miranda-Belmonte, H. U., Muñiz-Sánchez, V. and Corona, F. (2023). Word embeddings for topic modeling: An application to the estimation of the economic policy uncertainty index, *Expert Systems with Applications* **211**: 118499.

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

Park, J. and Cho, S. (2023). Incorporation of company-related factual knowledge into pre-trained language models for stock-related spam tweet filtering, *Expert Systems with Applications* **234**: 121021.

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

Perols, J. L. and Lougee, B. A. (2011). The relation between earnings management and financial statement fraud, *Advances in Accounting* **27**(1): 39–53.

Price, S. M., Doran, J. S., Peterson, D. R. and Bliss, B. A. (2012). Earnings conference calls and stock returns: The incremental informativeness of textual tone, *Journal of Banking & Finance* **36**(4): 992–1011.

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

Rawte, V., Gupta, A. and Zaki, M. J. (2020). A comparative analysis of temporal long text similarity: Application to financial documents, *Workshop on Mining Data for Financial Applications*, Springer, pp. 77–91.

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

Richardson, S. A., Sloan, R. G., Soliman, M. T. and Tuna, I. (2005). Accrual reliability, earnings persistence and stock prices, *Journal of Accounting and Economics* **39**(3): 437–485.

Robbins, H. and Monro, S. (1951). A stochastic approximation method, *The Annals of Mathematical Statistics* pp. 400–407.

Rogers, A., Kovaleva, O. and Rumshisky, A. (2021). A primer in BERTology: What we know about how BERT works, *Transactions of the Association for Computational Linguistics* **8**: 842–866.

Schimanski, T., Reding, A., Reding, N., Bingler, J., Kraus, M. and Leippold, M. (2024). Bridging the gap in ESG measurement: Using NLP to quantify environmental, social, and governance communication, *Finance Research Letters* **61**: 104979.

SEC (2003). Interpretation: Commission guidance regarding management's discussion and analysis of financial condition and results of operations, *Securities Act Release* (33-8350): 34–48960.

Tavcar, L. R. (1998). Make the md&a more readable, *The CPA Journal* **68**(1): 10.

Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market, *The Journal of Finance* **62**(3): 1139–1168.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u. and Polosukhin, I. (2017). Attention is all you need, *Advances in Neural Information Processing Systems*, Vol. 30, Curran Associates, Inc.

Walker, S. (2021). Critique of an article on machine learning in the detection of accounting fraud, *Econ Journal Watch* **18**(2): 61.

West, K. and McCracken, M. (1998). Regression-based tests of predictive ability, *International Economic Review* **39**(4): 817–40.

Whiting, D. G., Hansen, J. V., McDonald, J. B., Albrecht, C. and Albrecht, W. S. (2012). Machine learning methods for detecting patterns of management fraud, *Computational Intelligence* **28**(4): 505–527.

Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Li, Y. and Singh, V. (2021). Nyströmformer: A nyström-based algorithm for approximating self-attention, *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, pp. 14138–14148.

Yang, H., Liu, X.-Y. and Wang, C. D. (2023). FinGPT: Open-source financial large language models, *FinLLM Symposium at IJCAI 2023* .

Yang, X., Zhang, C., Sun, Y., Pang, K., Jing, L., Wa, S. and Lv, C. (2023). FinChain-BERT: A high-accuracy automatic fraud detection model based on NLP methods for financial scenarios, *Information* **14**(9): 499.

Yi, Z., Cao, X., Pu, X., Wu, Y., Chen, Z., Khan, A. T., Francis, A. and Li, S. (2023). Fraud detection in capital markets: A novel machine learning approach, *Expert Systems with Applications* **231**: 120760.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L. et al. (2020). Big bird: Transformers for longer sequences, *Advances in Neural Information Processing Systems* **33**: 17283–17297.

# A The hyperparameters of the gradient boosting algorithm

| Hyperparameter | Value | Description |
|---|---|---|
| n_estimators | 250 | The number of rounds for boosting |
| booster | gbtree | The base booster. In our case, gbtree is a decision tree |
| max_depth | 2 | The maximum depth of a tree |
| objective | binary:logistic | The loss function. In our case, the binary cross-entropy loss |
| colsample_bytree | 0.5 | The subsample ratio of columns when constructing each tree |
| reg_alpha | 0.1 | The L1 regularization term on tree weights (a higher value makes the model more conservative) |
| subsample | 0.9 | The subsample ratio of the training observations |
| learning_rate | 0.05 | The step size shrinkage used in update to prevents overfitting (a higher value makes the model more conservative) |
| tree_method | gpu_hist | The tree construction algorithm used in XGBoost |
| min_child_weight | 10 | The minimum sum of instance weight (Hessian) needed in a child |
| max_bin | 64 | The maximum number of discrete bins to bucket continuous features (only use if tree_method = hist or gpu_hist) |
| early_stopping_round | 25 | This parameter will stop training if the validation metric is not improving after the last early stopping round |

**Table 6:** This table presents the hyperparameters used to train the gradient boosting algorithms described in Section 5.4.

# B Training details of the LDA and Doc2Vec models

Different from LMs, which typically require intense computational costs, non-LM models such as LDA and Doc2Vec can be efficiently trained without GPUs. We, therefore, adopt the rolling window approach to train the LDA and Doc2Vec models to capture the linguistic variations of 10-K filings, following Brown et al. (2020). We utilize *gensim* to obtain efficient training of both models.

We implement an expanding window strategy to train the LDA model, beginning with the initial training data covering the period from 1994 to 1997. Subsequently, the model is employed to infer latent topics within the 10-K filings in 1998. These two steps constitute the first window. Transitioning to the second window, we update the LDA model trained in the first window with the 1998 filings. Subsequently, we utilize this updated model to infer topics from the 1999 10-K documents. This iterative process continues until we

reach the year 2018 in terms of topic inference. Suggesting by Brown et al. (2020), we fix the number of topics to 31. Other hyperparameters are set by default following *gensim*.

Unlike LDA which *gensim* offers a continual training mechanism,[16] Doc2Vec requires training from scratch upon the arrival of unseen data. Consequently, the expanding-window strategy potentially renders training Doc2Vec computationally challenging, especially when the training window spans several years. Therefore, we apply the rolling-window approach to train Doc2Vec in which we maintain a two-year training window. This method significantly reduces computational time while effectively capturing linguistic changes over time, leveraging a substantial training dataset of approximately 270 million words on average. As for the training hyperparameters, we set the context window to ten adjacent words, discard words appearing in less than 15 filings, and assign document vectors with 300 dimensions.

# C Performance of misreporting detection models over time

# D Comparison of Apple Inc.'s 10-K filings

[16] This feature enables a fully trained model to be revived and continue training with new, unseen data.

| Year | AUC | | ECM | | NDCG@1% | | NDCG@5% | |
|---|---|---|---|---|---|---|---|---|
| | Fin. Var. | Text | Fin. Var. | Text | Fin. Var. | Text | Fin. Var. | Text |
| 1999 | 0.511 | 0.505 | 0.180 | 0.191 | 0.013 | 0.009 | 0.029 | 0.028 |
| 2000 | 0.584 | 0.541 | 0.184 | 0.180 | 0.042 | 0.026 | 0.079 | 0.051 |
| 2001 | 0.611 | 0.579 | 0.180 | 0.180 | 0.022 | 0.023 | 0.045 | 0.048 |
| 2002 | 0.599 | 0.586 | 0.184 | 0.180 | 0.025 | 0.010 | 0.072 | 0.036 |
| 2003 | 0.616 | 0.585 | 0.188 | 0.193 | 0.013 | 0.012 | 0.044 | 0.041 |
| 2004 | 0.635 | 0.609 | 0.180 | 0.180 | 0.038 | 0.009 | 0.071 | 0.026 |
| 2005 | 0.576 | 0.507 | 0.182 | 0.181 | 0.014 | 0.001 | 0.038 | 0.008 |
| 2006 | 0.537 | 0.459 | 0.193 | 0.180 | 0.007 | 0.001 | 0.022 | 0.003 |
| 2007 | 0.634 | 0.552 | 0.180 | 0.180 | 0.016 | 0.003 | 0.050 | 0.013 |
| 2008 | 0.602 | 0.592 | 0.185 | 0.180 | 0.005 | 0.015 | 0.020 | 0.043 |
| 2009 | 0.586 | 0.456 | 0.180 | 0.180 | 0.005 | 0.001 | 0.019 | 0.009 |
| 2010 | 0.531 | 0.374 | 0.180 | 0.180 | 0.008 | 0.002 | 0.027 | 0.005 |
| 2011 | 0.601 | 0.503 | 0.184 | 0.180 | 0.016 | 0.005 | 0.044 | 0.016 |
| 2012 | 0.639 | 0.59 | 0.185 | 0.179 | 0.017 | 0.023 | 0.043 | 0.050 |
| 2013 | 0.694 | 0.647 | 0.188 | 0.176 | 0.009 | 0.017 | 0.031 | 0.056 |
| 2014 | 0.709 | 0.543 | 0.196 | 0.180 | 0.020 | 0.000 | 0.057 | 0.002 |
| 2015 | 0.740 | 0.539 | 0.171 | 0.180 | 0.031 | 0.005 | 0.100 | 0.015 |
| 2016 | 0.603 | 0.631 | 0.19 | 0.180 | 0.018 | 0.003 | 0.038 | 0.012 |
| 2017 | 0.557 | 0.544 | 0.181 | 0.180 | 0.006 | 0.000 | 0.020 | 0.001 |
| 2018 | 0.192 | 0.456 | 0.206 | 0.180 | 0.000 | 0.000 | 0.000 | 0.001 |

**Table 7:** This table reports the means of four metrics obtained by two models using (i) only the 17 financial variables (Fin. Var.), and (ii) only LongFinBERT text-based features (Text). Each entry in this table is the average metric value in each year over 500 runs. The prediction sample spans the period from 1999 to 2018.

| | 2015 | 2016 | Similarity |
|---|---|---|---|
| *Item 1 - Business* | **Company Background**<br><br>The Company designs, manufactures and markets mobile communication and media devices, personal computers and portable digital music players, and sells a variety of related software, services, accessories, networking solutions and third-party digital content and applications. The Company's products and services include iPhone®, iPad®, Mac®, iPod®, Apple Watch®, Apple TV®, a portfolio of consumer and professional software applications, iOS, macOS™, watchOS® and tvOS™ operating systems, iCloud®, Apple Pay® and a variety of accessory, service and support offerings. The Company sells and delivers digital content and applications through the iTunes Store®, App Store®, Mac App Store, TV App Store, iBooks Store™ and Apple Music® (collectively "Internet Services"). The Company sells its products worldwide through its retail stores, online stores and direct sales force, as well as through third-party cellular network carriers, wholesalers, retailers and value-added resellers. In addition, the Company sells a variety of third-party Apple compatible products, including application software and various accessories through its retail and online stores. The Company sells to consumers, small and mid-sized businesses and education, enterprise and government customers. The Company's fiscal year is the 52 or 53-week period that ends on the last Saturday of September. The Company is a California corporation established in 1977. | **Company Background**<br><br>The Company designs, manufactures and markets mobile communication and media devices, personal computers and portable digital music players, and sells a variety of related software, services, accessories, networking solutions and third-party digital content and applications. The Company's products and services include iPhone®, iPad®, Mac®, iPod®, Apple Watch®, Apple TV®, a portfolio of consumer and professional software applications, iOS, macOS™, watchOS® and tvOS™ operating systems, iCloud®, Apple Pay® and a variety of accessory, service and support offerings. The Company sells and delivers digital content and applications through the iTunes Store®, App Store®, Mac App Store, TV App Store, iBooks Store™ and Apple Music® (collectively "Internet Services"). The Company sells its products worldwide through its retail stores, online stores and direct sales force, as well as through third-party cellular network carriers, wholesalers, retailers and value-added resellers. In addition, the Company sells a variety of third-party Apple compatible products, including application software and various accessories through its retail and online stores. The Company sells to consumers, small and mid-sized businesses and education, enterprise and government customers. The Company's fiscal year is the 52 or 53-week period that ends on the last Saturday of September. The Company is a California corporation established in 1977. | 1.000<br>(whole Item: 0.828) |
| *Item 1A - Risk Factors* | The Company's operations and performance depend significantly on global and regional economic conditions. Uncertainty about global and regional economic conditions poses a risk as consumers and businesses may postpone spending in response to tighter credit, higher unemployment, financial market volatility, government austerity programs, negative financial news, declines in income or asset values and/or other factors. These worldwide and regional economic conditions could have a material adverse effect on demand for the Company's products and services. Demand also could differ materially from the Company's expectations as a result of currency fluctuations because the Company generally raises prices on goods and services sold outside the U.S. to correspond with the effect of a strengthening of the U.S. dollar. Other factors that could influence worldwide or regional demand include changes in fuel and other energy costs, conditions in the real estate and mortgage markets, unemployment, labor and healthcare costs, access to credit, consumer confidence and other macroeconomic factors affecting consumer spending behavior. These and other economic factors could materially adversely affect demand for the Company's products and services. | The Company's operations and performance depend significantly on global and regional economic conditions. Uncertainty about global and regional economic conditions poses a risk as consumers and businesses may postpone spending in response to tighter credit, higher unemployment, financial market volatility, government austerity programs, negative financial news, declines in income or asset values and/or other factors. These worldwide and regional economic conditions could have a material adverse effect on demand for the Company's products and services. Demand also could differ materially from the Company's expectations as a result of currency fluctuations because the Company generally raises prices on goods and services sold outside the U.S. to correspond with the effect of a strengthening of the U.S. dollar. Other factors that could influence worldwide or regional demand include changes in fuel and other energy costs, conditions in the real estate and mortgage markets, unemployment, labor and healthcare costs, access to credit, consumer confidence and other macroeconomic factors affecting consumer spending behavior. These and other economic factors could materially adversely affect demand for the Company's products and services. | 1.000<br>(whole Item: 0.7713) |
| *Item 7 - MD&A* | *Fiscal 2015 highlights*<br><br>Net sales rose 28% or $50.9 billion during 2015 compared to 2014, driven by a 52% year-over-year increase in iPhone® net sales. iPhone net sales and unit sales in 2015 increased in all of the Company's reportable operating segments. The Company also experienced year-over-year net sales increases in Mac®, Services and Other Products. Apple Watch®, which launched during the third quarter of 2015, accounted for more than 100% of the year-over-year growth in net sales of Other Products. Net sales growth during 2015 was partially offset by the effect of weakness in most foreign currencies relative to the U.S. dollar and lower iPad® net sales. Total net sales increased in each of the Company's reportable operating segments, with particularly strong growth in Greater China where year-over-year net sales increased 84%. | *Fiscal 2016 highlights*<br><br>Net sales declined 8% or $18.1 billion during 2016 compared to 2015, primarily driven by a year-over-year decrease in iPhone net sales and the effect of weakness in most foreign currencies relative to the U.S. dollar, partially offset by an increase in Services. In April 2016, the Company announced a significant increase to its capital return program by raising the expected total size of the program from $200 billion to $250 billion through March 2018. This included increasing its share repurchase authorization from $140 billion to $175 billion and raising its quarterly dividend from $0.52 to $0.57 per share beginning in May 2016. During 2016, the Company spent $29.0 billion to repurchase shares of its common stock and paid dividends and dividend equivalents of $12.2 billion. Additionally, the Company issued $23.9 billion of U.S. dollar-denominated term debt and A$1.4 billion of Australian dollar-denominated term debt during 2016. | 0.611<br>(whole Item: 0.764) |

**Table 8:** This table compares parts of the 10-K filings and the corresponding MD&A sections of Apple Inc. (CIK: 0000320193) in 2015 and 2016. We also present the cosine similarity between the two text snippets and the entire corresponding Items (in parentheses) using LongFinBERT.