Kaya, Devrimi; Reichmann, Doron; Reichmann, Milan

Out-of-sample predictability of firm-specific stock price crashes: A machine learning approach

Journal of Business Finance & Accounting

Provided in Cooperation with:
John Wiley & Sons

Suggested Citation: Kaya, Devrimi; Reichmann, Doron; Reichmann, Milan (2024) : Out-of-sample predictability of firm-specific stock price crashes: A machine learning approach, Journal of Business Finance & Accounting, ISSN 1468-5957, Wiley, Hoboken, NJ, Vol. 52, Iss. 2, pp. 1095-1115, https://doi.org/10.1111/jbfa.12831

Received: 13 March 2023 | Revised: 8 August 2024 | Accepted: 12 August 2024

DOI: 10.1111/jbfa.12831

JBFA

ARTICLE

Out-of-sample predictability of firm-specific stock price crashes: A machine learning approach

Devrimi Kaya[1] | **Doron Reichmann**[2] | **Milan Reichmann**[3]

[1]Friedrich-Alexander-University of Erlangen-Nürnberg, Chair of Business Analytics and Sustainability, Nürnberg, Germany

[2]Accounting Department, Goethe University Frankfurt, Frankfurt, Germany

[3]Chair of Banking and Finance, Leipzig University, Leipzig, Germany

Correspondence
Doron Reichmann, Theodor- W.-Adorno-Platz 1, 60629 Frankfurt am Main, Germany.
Email: d.reichmann@econ.uni-frankfurt.de

Abstract

We use machine learning methods to predict firm-specific stock price crashes and evaluate the out-of-sample prediction performance of various methods, compared to traditional regression approaches. Using financial and textual data from 10-K filings, our results show that a logistic regression with financial data inputs performs reasonably well and sometimes outperforms newer classifiers such as random forests and neural networks. However, we find that a stochastic gradient boosting model systematically outperforms the logistic regression, and forecasts using suitable combinations of financial and textual data inputs yield significantly higher prediction performance. Overall, the evidence suggests that machine learning methods can help predict stock price crashes.

KEYWORDS
machine learning, natural language processing, stock price crash risk, textual disclosures

1 | INTRODUCTION

Stock price crashes are prevalent in international financial markets (An et al., 2018; Jin & Myers, 2006). Studies extensively show that economic factors, such as financial opacity, agency costs and managerial incentives, can help explain stock price crashes (Hong et al., 2017; Hutton et al., 2009; Kim et al., 2019; Kim et al.,

2011a).[1] Most studies relate economic determinants to stock price crash risks using *within-sample* analyses. However, inferences about the *predictability* of stock price crashes are still limited. If investors fail to detect potential threats, stock prices can deviate from their fundamental values, increasing the risk of future price crashes. Hence, methods that help to identify crash-prone firms would offer significant value to investors.

We use machine learning methods to predict firm-specific stock price crashes.[2] Specifically, we evaluate the *out-of-sample* prediction performance of machine learning, compared to traditional regression approaches. While research on machine learning in accounting and finance typically focuses on numerical data as predictors (Bali et al., 2023; Chen et al., 2022; Gu et al., 2020), we also consider textual disclosures as an integral part of the financial reporting package. For instance, Lewis and Young (2019) document a substantial increase in textual firm disclosures over time. Machine learning methods can uncover complex patterns in both financial and textual data that help predict firm outcomes (e.g., Bertomeu et al., 2021; Bochkay et al., 2023; El-Haj et al., 2020). Hence, our study aims to leverage these empirical methods and test their performance for out-of-sample stock price crash predictions.

To conduct our analyses, we collect a sample of 39,583 US firm-year observations from the period 1996 to 2018 and calculate a wide set of 37 financial data inputs. We further retrieve the Management Discussion and Analysis (MD&A) sections of 10-K filings from the SEC's online EDGAR system to build textual inputs because research finds that textual disclosures of 10-K filings contain predictive signals of stock price crashes (e.g., Ertugrul et al., 2017; Kim et al., 2019; Reichmann, 2023).

As a benchmark model, we use a logistic regression (LOGIT) with numerical financial data inputs. This traditional regression approach is commonly used in both research and practice (Butaru et al., 2016; Jones & Hensher, 2004, 2007). In addition, we consider the support vector machine (SVM; Vapnik, 1998), two models based on decision trees, namely, the random forest (RF; Breiman, 2001) and the stochastic gradient boosting (SGB; Friedman, 2002), and two neural networks, a dense neural network for numerical inputs (NN) and a convolutional neural network for textual inputs (CNN) (e.g., Hinton et al., 2006; LeCun et al., 1989, 2015).

While numerals can be directly used as inputs to a machine learning model, text represents natural language, which requires conversion to numerical representations. Our empirical approach is as follows. First, we mimic previous studies on crash risk. Specifically, we use simple content analyses to estimate high-level textual characteristics that capture the tone, ambiguity and complexity of MD&A and use those as inputs to a LOGIT (Ertugrul et al., 2017; Kim et al., 2019). Second, to provide our model with more detailed textual information, we estimate numerical document representations using term frequency-inverse document frequency (TF-IDF) weights.[3] This approach yields large vectors that contain relative term frequencies for each MD&A, which can be used as inputs to a LOGIT, SVM, RF and SGB. Finally, we consider a word embedding approach. Word embeddings are vector representations of words and phrases that capture semantic information. We use word2vec to generate word embeddings (Mikolov, Chen, et al., 2013), which serve as inputs to a CNN. This is our only model capable of considering *sequential* dependencies in financial narratives.

To train our models, we employ a rolling sample splitting scheme (e.g., Chen et al., 2022; Gu et al., 2020). For each year in our test period, 2001–2018, we use the five preceding years to train, validate and estimate the models that are then tested in next year's hold-out-sample. While this approach requires us to freshly train new models every year in the test period, it allows the models to be continuously trained on recent data (Chen et al., 2022). Moreover, we recode serial stock price crashes of firms that span both the model estimation and test set (henceforth: serial crashes) as non-crash observation in the training set. This approach mitigates concerns that flexible machine learning models identify crash-prone firms rather than crash-prone firm-years. Consistent with recent work (e.g., Bertomeu et al., 2021; Chen et al., 2022; Mai et al., 2019), we evaluate our models using the area under the receiver operating characteristic (ROC) curve (AUC) and a set of catch rates with varying probability cut-offs.

[1] The literature defines stock price crashes as extreme negative outliers in the distribution of firm returns (e.g., Hutton et al., 2009; Kim et al., 2019; Kim et al., 2011a, 2011b).

[2] There have been several recent studies using machine learning in accounting and finance, including in the area of earnings prediction (Chen et al., 2022; Jones et al., 2023), accounting misstatements (Bertomeu et al., 2021) and distress prediction (Jones et al., 2015, 2017).

[3] See Brown and Tucker (2011) for an explanation of TF-IDF weights.

Our results suggest that a LOGIT with numerical inputs (henceforth: LOGIT(Num)) performs reasonably well and serves as a strong benchmark (Jones et al., 2015, 2017). The model yields an AUC of 55.30%, significantly higher than the 50% of a random guess. Moreover, it outperforms the SVM(Num), RF(Num) and NN(Num). However, the SGB(Num) yields an AUC of 56.26%, which significantly exceeds the AUC of LOGIT(Num), suggesting that machine learning methods can help improve the prediction of stock price crashes. The SGB is also the best-performing model with textual inputs from 10-K filings. We find that an SGB that uses TF-IDF weights, SGB(Text), yields an AUC of 56.18%, outperforming the *numerical* benchmark LOGIT(Num). Moreover, using a LOGIT model with a set of high-level textual characteristics as commonly employed in within-sample studies on crash risk significantly *underperforms* SGB(Text) (Ertugrul et al., 2017; Kim et al., 2019). This finding supports the view that machine learning is powerful to detect predictive patterns that are difficult to uncover with simple content analyses (e.g., Bochkay et al., 2023; El-Haj et al., 2019).

We also test whether *combining* numerical and textual data inputs further improves crash predictions. Therefore, we test two approaches. First, we concatenate numerical and textual inputs, resulting in a new input vector with combined data inputs. We then retrain our models using these combined inputs. Second, we employ an average ensemble approach by combining the probability estimates of two separate models using a simple average. We find that the easy-to-implement average ensemble approach performs better. Combining the probability estimates of SGB(Num) and SGB(Text) using a simple average leads to a significant improvement in predictive power over using SGB(Num). This result suggests that textual inputs contain information that is incrementally informative to numerical financial data for out-of-sample crash predictions. An analysis of the model performance over time reveals that this finding is systematic during our test period, exists during almost all test years and does not diminish over time.

We then examine the inner workings of our models by identifying meaningful important predictors for both SGB(Num) and SGB(Text). Specifically, we perform permutation feature importance, a model inspection technique that observes the degradation of a model's predictive power when randomly shuffling values of a single predictor. We find that common numerical financial variables used in the literature, such as firm size, the negative skewness of returns, return on assets and book-to-market ratios help predict stock price crashes (e.g., Hutton et al., 2009; Kim et al., 2019). Turning to textual predictors, we find that terms related to firm growth such as "growth," "acquisition" and "spending" as well as terms signaling threats such as "downgrading" and "may never succeed" help SGB(Text) distinguish crash from non-crash observations. These predictors are generally consistent with the literature (Hutton et al., 2009; Reichmann, 2023).

Finally, we test the sensitivity of our models to recoding serial crashes as non-crash observations. Failing to correct for serial crashes can lead to inflated model performance if models tend to identify crash-prone firms rather than crash-prone firm-years. We provide evidence that failing to correct for serial crashes substantially inflates the model performance of various model architectures. Our findings should caution future research when testing the out-of-sample predictability of stock price crashes.

This study contributes to prior literature in two ways. First, it extends the broad literature on stock price crash risk. A dominant theme in the literature is that economic factors, such as financial opacity, agency costs or managerial incentives help explain stock price crashes (e.g., Hong et al., 2017; Hutton et al., 2009; Kim et al., 2011a), but most studies focus on *within-sample* analyses to identify the determinants of crash risk. We contribute to the literature by examining the *out-of-sample* predictability of stock price crashes using machine learning methods. Further, our results draw attention to the high prediction power of textual data based on MD&A disclosures. Suitable combinations of financial and textual data inputs yield significantly higher prediction performance, a finding that should matter to investors and regulators. For instance, the SEC and other regulators aim to put more emphasis on firms' narrative disclosures (Eaglesham, 2013). Our results suggest that machine learning methods can help uncover signals from disclosures that are associated with crash risk.

Second, we complement the literature in accounting and finance using machine learning with big data in financial markets (e.g., Bianchi et al., 2020; Chen et al., 2022; Gu et al., 2020; Jones et al., 2023). For example, Gu et al. (2020) apply machine learning algorithms to study the behavior of expected stock returns. Chen et al. (2022) use

TABLE 1 Sample selection.

Data filters	Firm-years	Firms
Active nonfinancial/nonutility US firms on Refinitiv (1996–2018)	85,584	8,026
Fiscal year-end price ≥ 1$	70,780	7,794
≥ 26 weekly returns per year	64,258	6,783
Management Discussion and Analysis (MD&A) data from EDGAR	39,583	3,563

Note: This table presents the data filters of our sample selection.

high-dimensional financial data to predict the direction of one-year-ahead earnings changes. We extend this research by examining whether machine learning can detect predictive signals in both financial and textual data that help predict stock price crashes. We find that an SGB systematically outperforms other algorithms in predicting stock price crashes. Further, our results suggest that the combination of numerical and textual inputs can improve model accuracy.

While the literature on stock price crashes predominantly focuses on US capital markets, the use and importance of machine learning to predict stock price crashes in international financial markets is likely to grow in the near future. Starting with 2027, certain financial and sustainability-related information of European listed firms will be filed via the centralized European Single Access Point (ESAP), which is likely to foster the access and processing of firm disclosures (e.g., El-Haj et al., 2019; Kaya & Seebeck, 2019). Further, data collection costs are likely to be lower through a centralized document depository, such as ESAP, where annual reports will be accessible in a structured, electronic reporting format (e.g., the European Single Electronic Format).[4]

2 | SAMPLE SELECTION AND MODEL INPUTS

2.1 | Sample selection

Our initial sample consists of nonfinancial and nonutility US listed firms available on Refinitiv Datastream and Refinitiv Worldscope (Reichmann & Reichmann, 2022).[5] The sample starts in 1996 when all publicly listed US firms had to file financial reports electronically and ends in 2018. We drop observations with a fiscal year-end share price below $1 and observations with less than 26 available weekly returns in a fiscal year. For each firm, we collect available 10-K filings from the SEC's EDGAR system and combine the files with the sample.[6] We focus on the MD&A sections of 10-K filings, which prior literature finds to be associated with future crash risk (Reichmann, 2023; Reichmann et al., 2022). The sample selection is summarized in Table 1. Our final sample consists of 39,583 firm-year observations for 3,563 unique firms.

2.2 | Stock price crashes

We follow prior literature to measure stock price crashes and calculate firm-specific weekly returns ($W_{j,t}$) by estimating the following expanded index model for each firm-year (Hutton et al., 2009; Kim et al., 2019; Kim et al., 2011b; Reichmann & Reichmann, 2022):

$$r_{j,\tau} = \beta_0 + \beta_1 r_{m,\tau-1} + \beta_2 r_{i,\tau-1} + \beta_3 r_{m,\tau} + \beta_4 r_{i,\tau} + \beta_5 r_{m,\tau+1} + \beta_6 r_{i,\tau+1} + \epsilon_{j,\tau}.$$

[4] We refer interested readers to El-Haj et al. (2019).

[5] We apply the screens and filters proposed by Schmidt et al. (2019) when constructing our initial sample.

[6] We use the term "10-K" to refer to the form 10-K and its variants: 10-K405, 10KSB and 10KSB40.

$r_{j,\tau}$ denotes stock returns for firm j in week τ. $r_{m,\tau}$ is the weekly return on the CRSP value-weighted market index m and $r_{i,\tau}$ denotes the weekly return on the Fama–French value-weighted index for industry i. To avoid look-ahead biases, we define a fiscal year as the 12-month period ending three months after the fiscal year-end to account for the reporting lag of the 10-K (e.g., Kim et al., 2019; Reichmann et al., 2022). The firm-specific weekly return $W_{j,\tau}$ is calculated as the natural logarithm of 1 plus the regression residual $\epsilon_{j,\tau}$. Finally, the outcome $CRASH_{t+1}$ is an indicator variable that equals 1 if a firm-specific weekly return $W_{j,\tau}$ drops 3.09 standard deviations below its yearly mean in the period $t + 1$ and 0 otherwise (Reichmann & Reichmann, 2022). We chose 3.09 standard deviations to generate a 0.1% frequency in the log-normal distribution (e.g., Hutton et al., 2009; Kim et al., 2019; Kim et al., 2011a, 2011b).

2.3 | Numerical inputs

We compile a list of financial variables that are associated with crash risk (e.g., Chen et al., 2001; Hutton et al., 2009; Kim et al., 2019; Wu & Lai, 2020). Because the flexibility of machine learning methods enables considering a broader input set compared to traditional econometric techniques, we further include financial ratios that are influential predictors in related tasks such as bankruptcy prediction (e.g., Mai et al., 2019; Reichmann & Reichmann, 2022). In total, we collect a set of 37 numerical financial variables. Table 2, panel A, provides definitions of our numerical inputs. All numerical inputs are winsorized at the 1% and 99% levels to avoid extreme outliers. We impute missing values with zeros.

2.4 | Textual inputs

We also examine the predictive power of textual data, which cannot be directly fed into a machine learning model. Hence, we convert text to numerical representations that capture the content of a given document. We consider three different approaches with an increasing degree of complexity (Reichmann & Reichmann, 2022).

First, we estimate textual characteristics using a simple content analysis. This approach converts text into a score that represents a textual characteristic such as tone or linguistic complexity. Following previous research, we estimate textual characteristics that are associated with future crash risk. For instance, previous research finds that firms with more ambiguous and complex financial reports are prone to stock price crashes (e.g., Ertugrul et al., 2017; Kim et al., 2019). Therefore, we calculate the fraction of ambiguous words using the Fin-Unc (*UNCERTAIN*) and MW-Weak (*WEAK_MODAL*) word lists of Loughran and McDonald (2011) and the (modified) FOG index proposed by Kim et al. (2019) as textual predictors (*FOG* and *MODFOG*).

In addition, we proxy for the tone of the MD&A using the Fin-Neg word list (*NEGATIVE*) of Loughran and McDonald (2011) because disclosure tone is likely to be informative about crash risk (e.g., Fu et al., 2021; Reichmann, 2023). Finally, we use proxies for information quantity by calculating the natural logarithm of 1 plus the total number of words in an MD&A (*LOGLENGTH*) and its file size in megabytes (*LOGFILESIZE*; e.g., Ertugrul et al., 2017; Loughran & McDonald, 2014). Table 2, panel B, provides definitions of textual MD&A characteristics.

Second, we estimate more detailed document representations by computing the term TF-IDF weights for words and phrases in each MD&A as follows (Reichmann & Reichmann, 2022):

$$w_{i,j} = \begin{cases} tf_{i,j} \log\left(\frac{N}{df_i}\right), & \text{if } tf_{i,j} \geq 1 \\ 0 \quad \text{otherwise.} \end{cases}$$

N denotes the total number of documents in a sample; df_i is the number of documents containing the term i; and $tf_{i,j}$ presents the word count of term i in document j. The TF $tf_{i,j}$ measures the importance of a term within a document, whereas the IDF $\log(\frac{N}{df_i})$ adjusts for the frequency of a term in the entire sample of documents (see Brown &

TABLE 2 Variable definitions.

Predictor	Definition	Predictor	Definition
Panel A: Numerical predictors			
ACTLCT	Current assets/total liabilities	*LCTSALE*	Current liabilities/sales
ADJROTA	Industry-adjusted return on tangible assets	*LEV*	Total liabilities/total assets
APSALE	Accounts payable/sales	*LOGAT*	Log(total assets)
CASHAT	Cash and short-term investments/total assets	*LOGMV*	Log(market value)
CHAT	Cash/total assets	*LOGSALE*	Log(sales)
CHLCT	Cash/current liabilities	*MTB*	Market value/book value
CFVOL	Standard deviation of cash flow/total assets over the five preceding fiscal years	*NCSKEW*	Negative skewness of firm-specific weekly stock returns
DTURN	Detrended monthly turnover	*NIAT*	Net income/total assets
EARNVOL	Standard deviation of EBIT/total assets over the five preceding fiscal years	*NISALE*	Net income/sales
EBITAT	EBIT/total assets	*OPAQUE*	Three-years moving sum of discretionary accruals
EBITDAAT	EBITDA/total assets	*RETAT*	Retained earnings/total assets
EBITSALE	EBIT/sales	*RELCT*	Retained earnings/current liabilities
FAT	Total debt/total assets	*ROA*	Operating income/total assets
HHI	Herfindahl–Hirschman Index	*ROS*	Operating income/sales
INVCHINVT	Growth of inventories/inventories	*ROTA*	Return on tangible assets
LCTAT	Current liabilities/total assets	*SALESVOL*	Standard deviation of sales/total assets over the five preceding fiscal years
LCTCHAT	(Current liabilities—cash)/total assets	*SIGMA*	Standard deviation of weekly firm-specific stock returns
LCTLT	Current liabilities/total liabilities		
Panel B: Textual MD&A characteristics			
LOGLENGTH	Log(number of words)	*NEGATIVE*	Negative words/total words
WEAK_MODAL	Weak modal words/total words	*UNCERTAIN*	Uncertainty words/total words
FOG	(words per sentence + percentage of complex words) × 0.4	*MODFOG*	(words per sentence + percentage of modified complex words) × 0.4
LOGFILESIZE	Log(filesize in megabytes)		

Note: This table presents variable definitions. Panel A provides the definitions of 37 financial ratios employed in our numerical models. Panel B provides the definitions of textual MD&A characteristics.

Tucker, 2011, for details). Given a vocabulary of size $|V|$, a document j is represented as a vector with $|V|$ dimensions. Each dimension represents a word in the vocabulary and its corresponding TF-IDF weight, $w_{i,j}$. Because our machine learning models require inputs to have the same lengths, we consider only the 40,000 most common terms in our sample of MD&A.[7] Imposing an upper limit for the dimensionality of document vectors is common in the

[7] For our main tests, we employ a rolling sample splitting scheme. Hence, for each rolling split, we estimate new TF-IDF representations for the sample of documents in a given rolling split.

literature because it ensures that the dimensionality of document vectors is not driven by extreme outliers (e.g., Mai et al., 2019).[8]

Third, we consider a word embedding approach. Word embeddings are a type of word representation that allows words to be represented as vectors in a vector space. These vector representations are derived through contextual analysis of word occurrences within the corpus of MD&A, enabling the capture of semantic relationships between words. For instance, words that are semantically similar tend to be closer in the vector space. To construct word embeddings for our sample of MD&A, we follow previous literature (e.g., Du et al., 2022; Li et al., 2021; Mai et al., 2019) and use word2vec (Mikolov, Chen, et al., 2013). We estimate 50-, 100-, 200- and 300-dimensional word embeddings for words occurring in our sample of MD&A.[9]

For both the estimations of TF-IDF weights and word embeddings, extensive text preprocessing is performed to reduce feature dimensionality, improve generalization and form phrases that help our models to differentiate context. Specifically, we (i) replace named entities such as company names, person, money values or dates with predefined tags using named entity recognition; (ii) perform lemmatization to reduce words to their base form; (iii) form phrases using a data-driven approach that joins words with significant co-occurrence (e.g., Mikolov, Sutskever, et al., 2013); and (iv) remove stopwords (e.g., Li et al., 2021; Reichmann & Reichmann, 2022; Reichmann et al., 2022).[10]

2.5 | Descriptive statistics

Table 3 provides summary statistics for our sample. We report our main outcome variable, common controls in the crash risk literature and textual characteristics of our MD&A sample. The descriptive statistics suggest that 23% of all firm-years in our sample experience a firm-specific stock price crash. Moreover, our sample is characterized by growth firms as indicated by a mean (median) market-to-book ratio (*MTB*) of 3.223 (2.229). The average firm-year has financial leverage of 0.616 (0.475) at the sample mean (median).

We find that an MD&A contains, on average, 1.2% negative words (*NEGATIVE*). Only 0.4% of words are weak modal words (*WEAK_MODAL*), whereas 1.4% are associated with financial uncertainty (*UNCERTAIN*), generally consistent with Loughran and McDonald (2011). Moreover, the mean of the FOG index is 18.378 and adjusting the FOG index for financial terms that are typically not considered complex by investors yields a substantially lower average score of 13.284 (*MODFOG*). Collectively, our sample resembles previous ones used in research on crash risk (e.g., Ertugrul et al., 2017; Kim et al., 2019).

3 | APPROACH TO PREDICTION

3.1 | Sample partitioning

To train and test our models, we simulate a realistic forecasting scenario. Each model is trained on past data and then evaluated on next year's hold-out sample. Specifically, our approach implements a rolling sample splitting scheme, in which the training and validation samples gradually shift forward in time, but the number of years in each sample is held constant (Chen et al., 2022).

[8] Choosing an upper bound also depends on different aspects such as the number of training samples or the applied text preprocessing steps. When we consider a rolling split for the test year 2018 that includes the five previous years for model estimation, untabulated results suggest that an upper bound of 40,000 words considers all words that appear more than five times in the sample of documents. This suggests that choosing an upper bound of the 40,000 most frequently occurring words seems reasonable.

[9] To train word2vec, we chose a skip-gram approach with a window size of five, 20 iterations over the corpus, a minimum word count of five and negative sampling to accelerate training (e.g., Li et al., 2021; Reichmann & Reichmann, 2022). To ensure the validity of our word2vec embeddings, we perform descriptive analyses in Appendix A1.

[10] To better illustrate our text preprocessing, consider the following sentence: "Gross margins decreased as a result of lower sales in 2003." After text preprocessing the sentence reads as follows: "gross_margin decrease result low sale -date-."

TABLE 3 Descriptive statistics.

	N	Mean	Std.	Q25	Median	Q75
Outcome						
$CRASH_{t+1}$	39,583	0.230	0.421	0.000	0.000	0.000
Common Crash Controls						
$LOGMV_t$	39,583	12.950	2.363	11.279	13.043	14.581
MTB_t	39,583	3.223	9.809	1.206	2.229	4.108
LEV_t	39,583	0.616	0.935	0.279	0.475	0.655
ROA_t	39,583	−0.004	0.318	−0.014	0.070	0.140
$DTURN_t$	39,583	0.002	0.109	−0.016	0.000	0.026
$NCSKEW_t$	39,583	0.116	0.927	−0.412	0.002	0.519
RET_t	39,583	−0.261	0.466	−0.258	−0.101	−0.040
$SIGMA_t$	39,583	0.057	0.045	0.028	0.045	0.072
MD&A Characteristics						
$LOGLENGTH_t$	39,583	3.787	0.324	3.616	3.846	4.012
$NEGATIVE_t$	39,583	0.012	0.005	0.009	0.012	0.015
$WEAK\_MODAL_t$	39,583	0.004	0.003	0.003	0.004	0.005
$UNCERTAIN_t$	39,583	0.014	0.005	0.011	0.014	0.017
FOG_t	39,583	18.376	1.711	17.327	18.326	19.338
$MODFOG_t$	39,583	13.284	1.573	12.317	13.224	14.160
LOGFILESIZE	39,583	0.052	0.031	0.028	0.048	0.070

Note: This table presents descriptive statistics of our outcome variable, common numerical crash controls and MD&A characteristics. $CRASH_{t+1}$ is an indicator variable that equals 1 if a firm-specific weekly return drops 3.09 standard deviations below its yearly mean in the period $t + 1$ and 0 otherwise. All other variables are defined in Table 2.

For each year in the test period from 2001 to 2018 (e.g., 2001), the models are trained in the five preceding years. The first four years (e.g., 1996−1999) are used for training, and the fifth year is used for validation (e.g., 2000) to tune model hyperparameters (Chen et al., 2022). The prediction model is then estimated on the validation and training sample (e.g., 1996−2000) (Bertomeu et al., 2021). Finally, we analyze the model performance on the hold-out-test sample (e.g., 2011).

Although we acknowledge that using a rolling sample splitting scheme is computationally demanding, it has at least two benefits. First, it strictly avoids temporal leakage as each model is trained on past data and then tested on next year's hold-out sample. Second, it allows the models to be continuously trained on new data.

3.2 | Machine learning models

To provide a comprehensive analysis of the predictability of stock price crashes, we consider a variety of different models. Specifically, we consider (i) a traditional regression model, (ii) SVMs, (iii) models based on decision trees and (iv) neural networks. Table 4 provides an overview of the models and their respective data inputs. Moreover, for each rolling split, we separately tune model hyperparameters using the validation set. Appendix A2 provides the hyperparameter space used for tuning each of our models.[11]

[11] For more details on widely used machine learning methods, we refer to Bochkay et al. (2023).

TABLE 4 Model overview.

Type	Model	Definition	Numerical input	Textual input
Regression model	LOGIT	Logistic regression	37 financials	Textual characteristics; term frequency-inverse document frequency (TF-IDF) representations
Support vector machine	SVM	Support vector machine	37 financials	TF-IDF representations
Decision tree	RF	Random forest	37 financials	TF-IDF representations
	SGB	Stochastic gradient boosting	37 financials	TF-IDF representations
Neural network	NN	Dense neural network	37 financials	–
	CNN	Convolutional neural network	–	word2vec embeddings

Note: This table presents an overview of the machine learning models and their respective input features used in our study.

3.2.1 | Regression model

As a benchmark, we consider a logistic regression, LOGIT(Num) with numerical financial data as inputs. LOGIT models are extensively employed for binary classification tasks in both financial research and practice (e.g., Baesens et al., 2003; Butaru et al., 2016; Dechow et al., 2011; Jones & Hensher, 2004, 2007; Shumway, 2001). We implement a LOGIT with L1 regularization to prevent the model from overfitting and choose a LOGIT(Num) as a reasonable benchmark an investor would use to predict stock price crashes.

In addition, we use a LOGIT to implement two models with textual inputs. First, because previous research typically uses LOGIT to model the association between textual characteristics and stock price crashes (e.g., Ertugrul et al., 2017; Kim et al., 2019), LOGIT(TextChar) denotes a LOGIT model that uses the textual characteristics described in Section 2.4 as inputs. Second, LOGIT(Text) denotes a LOGIT model that uses TF-IDF representations of MD&A as inputs.

3.2.2 | Support vector machines

We estimate an SVM with a nonlinear kernel function to transform the training data into a higher dimensional feature space. SVM(Num) denotes an SVM that uses numerical financial data as inputs, and SVM(Text) is an SVM that uses TF-IDF weights as inputs.

3.2.3 | Decision trees

Following previous literature, we consider the RF and SGB as models based on decision trees (e.g., Chen et al., 2022; Jones et al., 2023). RF(Num) and SGB(Num) denote models that use financial numerical inputs, whereas RF(Text) and SGB(Text) use TF-IDF weights as representations of MD&A.

3.2.4 | Neural networks

We consider two separate neural networks for numerical and textual inputs. Feeding numerical data to a neural network follows a straightforward path. Specifically, we feed the numerical financial data forward to a set of dense

layers, aiming to forecast whether a firm-year experiences a stock price crash in the subsequent year (Reichmann & Reichmann, 2022). This specific model is denoted as NN(Num).

For textual inputs, we employ a CNN (LeCun et al., 1989, 2015; Mai et al., 2019; Reichmann & Reichmann, 2022). Unlike the previous models that use TF-IDF weights that do not account for word order within an MD&A, the CNN can consider word sequences, potentially revealing further insights into future crashes (Reichmann & Reichmann, 2022). Therefore, we use the word2vec embeddings to enable the CNN to consider each word in an MD&A as a numerical representation, capturing the semantic essence of each word.

Because training machine learning models typically requires using equally sized inputs, we perform padding to normalize each MD&A to a length of 10,000 words by truncating longer MD&A and adding vectors of zeros to shorter MD&A (Mai et al., 2019). 10,000 words approximate the 97.5% quantile of input features in our sample of MD&A sections after text preprocessing. Hence, this approach captures the full content of most MD&A documents in our sample while, at the same time, it does not extend the dimension of vectors for outliers in our sample. We denote a CNN with word2vec embeddings as CNN(Text).

Neural networks are commonly initiated with different random seeds that affect the final model fit. Thus, estimating the same architecture can lead to different results, potentially resulting either in inflated or deflated performance relative to the true (average) performance of the model. Therefore, we adopt the approach outlined by Gu et al. (2020) and take the average probability estimate of five models initiated with different random seeds (average ensemble).

3.3 | Performance evaluation

We use the AUC as our main metric for out-of-sample performance (e.g., Chen et al., 2022; Cheng et al., 2018; Mai et al., 2019). The ROC curve aims to graphically represent the trade-off between the true positive rate, also referred to as the catch rate, and the false positive rate. The catch rate is defined as $\frac{True\ Positive}{True\ Positive+False\ Negative}$ and, therefore, represents the fraction of correctly predicted crashes relative to the total number of crashes in the sample. Conversely, the false positive rate, defined as $\frac{False\ Positive}{False\ Positive+True\ Negative}$, estimates the fraction of false alarms relative to the total number of non-crashes in the sample. The AUC is then defined as the area under the ROC curve and spans from 0 and 1. According to Fawcett (2006), the AUC denotes the probability that a model will rank a randomly chosen positive (crash) observation higher than a randomly chosen negative (non-crash) observation. With the AUC of a random model set at 50%, any model with an AUC above 50% indicates predictive power.

In addition, we also consider catch rates using probability cut-offs in {10%, 20%, …, 50%} (e.g., Bertomeu et al., 2021). For example, assuming a cut-off of 10%, we investigate a model's 10% highest probability estimates within a year and estimate how many stock price crashes these estimates would have detected relative to all stock price crashes in the sample. Since selecting 10% of observations randomly would, on average, yield a catch rate of 10%, we report the excess catch rate. This metric represents the catch rate exceeding the probability cut-off.

3.4 | Serial crashes

Machine learning models are flexible which raises concerns that they fit a firm rather than a firm-year. Hence, observations of firms that experience consecutive stock price crashes (i.e., serial crashes) spanning both the training and test samples can inflate the performance. To improve the generalizability of our results, we recode serial crashes in the training set as zeros if they span both the training and test sets. For instance, let the test year be 2018 and the model is estimated on the five preceding years from 2013 to 2017. If a firm has a crash observation in 2018 (test) *and* in 2017 (train), we recode the 2017 observation as a non-crash observation. Recoding serial crash observations likely impedes model performance because the training set contains fewer crash observations that enable the model to detect predictive signals. Therefore, the results reported in our main tests are likely to be conservative.

TABLE 5 Out-of-sample predictability of stock price crashes.

Panel A: Numeric models										
		AUC versus 50%		*AUC versus LOGIT(Num)*		*Excess catch rate*				
Models	**AUC**	**Diff.**	***p*-val.**	**Diff.**	***p*-val.**	**10%**	**20%**	**30%**	**40%**	**50%**
LOGIT(Num)	55.30%	5.30%	<0.01			2.97%	4.00%	4.61%	5.00%	5.16%
SVM(Num)	52.90%	2.90%	<0.01	−2.40%	<0.01	0.22%	−0.14%	−0.36%	−0.61%	−0.28%
RF(Num)	53.42%	3.42%	<0.01	−1.88%	<0.01	0.55%	1.40%	1.69%	2.23%	2.27%
SGB(Num)	**56.26%**	**6.26%**	**<0.01**	**0.96%**	**<0.01**	**3.10%**	**5.19%**	**6.01%**	**6.37%**	**6.72%**
NN(Num)	54.73%	4.73%	<0.01	−0.57%	<0.01	2.27%	3.38%	4.43%	4.74%	4.94%
Panel B: Textual models										
		AUC versus 50%		*AUC versus LOGIT(Num)*		*Excess catch rate*				
Models	**AUC**	**Diff.**	***p*-val.**	**Diff.**	***p*-val.**	**10%**	**20%**	**30%**	**40%**	**50%**
LOGIT(TextChar)	53.11%	3.11%	<0.01	−2.19%	<0.01	1.09%	0.74%	1.54%	1.62%	1.76%
LOGIT(Text)	52.84%	2.84%	<0.01	−2.45%	<0.01	9.60%	0.09%	0.83%	1.42%	2.02%
SVM(Text)	49.39%	−0.61%	0.081	−5.91%	<0.01	−5.72%	−7.68%	−8.47%	−7.17%	−5.61%
RF(Text)	56.17%	6.17%	<0.01	0.87%	0.023	**3.10%**	4.43%	5.40%	5.45%	5.43%
SGB(Text)	**56.18%**	**6.18%**	**<0.01**	**0.88%**	**0.013**	3.00%	**4.79%**	**5.69%**	**6.07%**	**6.12%**
CNN(Text)	54.77%	4.77%	<0.01	−0.53%	0.211	2.63%	4.41%	5.11%	5.38%	5.92%

Note: This table presents the results of predicting one-year-ahead stock price crashes. Panel A reports the results of models using numerical inputs. Panel B reports the results of models using textual inputs. "Num" denotes the 37 financial variables defined in Table 2, panel A. "TextChar" denotes textual characteristics defined in Table 2, panel B. "Text" denotes word2vec embeddings for the CNN and TF-IDF weights for the remaining text models. The table reports the area under the receiver operating characteristic (ROC) curve (AUC) and excess catch rates for varying cut-offs. Differences between AUC scores are tested for statistical significance using the DeLong test. The best-performing models in each panel are presented in bold.

4 | OUT-OF-SAMPLE PERFORMANCE

4.1 | Predicting stock price crashes

Panel A of Table 5 reports the out-of-sample performance evaluation for models using 37 numerical data inputs. The results show that all models yield AUC above 50%. DeLong tests further suggest that differences are statistically higher than a random guess ($p < 0.01$; DeLong et al., 1988). Therefore, our results indicate that machine learning models detect one-year-ahead stock price crashes using financial predictors.

Consistent with previous literature on financial prediction tasks (e.g., Jones et al., 2015, 2017), we find that LOGIT(Num) performs reasonably well and serves as a strong benchmark for crash prediction. Specifically, it yields an AUC of 55.30%, which significantly outperforms the SVM(Num), RF(Num) and NN(Num) with AUC of only 52.90%, 53.42% and 54.73%, respectively. However, SGB(Num) has both the highest AUC and the highest catch rates across all cut-offs. Specifically, it yields an AUC of 56.26%, which is significantly higher than the AUC of LOGIT(Num) ($p < 0.01$). In relative terms, this is equivalent to an improvement of $(\frac{6.26\%}{5.30\%} - 1 =)$ 18% in accuracy to predict stock price crashes better than chance.

Panel B shows the classification results of models using textual inputs. Except for SVM(Text), all models yield AUC above 50% ($p < 0.01$), suggesting that text is informative about future stock price crashes. The LOGIT model does not benefit from using high-dimensional TF-IDF weights as inputs (LOGIT(Text), AUC = 52.84%), compared to using high-level text characteristics from previous literature (LOGIT(TextChar), AUC = 53.11%). However, RF(Text), SGB(Text)

and CNN(Text) all have significantly higher AUC compared to LOGIT(TextChar) ($p < 0.01$). An important implication is that textual characteristics identified by previous research using basic content analyses are unlikely to capture all relevant aspects of textual disclosures that are informative about crash risk.

Although CNN(Text) is the most elaborate model architecture (AUC = 54.77%), it underperforms compared to models based on decision trees.[12] The strongest decision tree, SGB(Text), yields a significantly higher AUC of 56.18% ($p < 0.01$) and has higher catch rates across all probability cut-offs. Notably, SGB(Text) even outperforms the *numerical* benchmark model LOGIT(Num) ($p < 0.05$) and performs like SGB(Num) ($p = 0.845$), the best-performing numerical machine learning model. This finding emphasizes the high information content of textual disclosures for stock price crash risk.

Collectively, our results suggest that an SGB is the best-performing model for both numerical and textual inputs. SGB predicts stock price crashes significantly better than chance and outperforms traditional approaches. Our findings support previous literature that machine learning can help investors to improve their decision-making (e.g., Bianchi et al., 2020; Gu et al., 2020; Jones et al., 2023).

4.2 | Combining numerical and textual inputs

Previous research suggests that *combining* different information sets can yield higher predictive power (e.g., Mai et al., 2019; Peat & Jones, 2012). Therefore, we test two approaches to combine numerical and textual inputs. First, we concatenate numerical and textual inputs and use the resulting feature vector to train new models. This standard approach allows a machine learning model to consider both input sources during training (e.g., Mai et al., 2019).[13] Second, we combine two models using an average ensemble approach. Specifically, given two separate models Model(Num) and Model(Text), we take the simple average of the models' probability estimates. Combining forecasts using averages is easy to implement, needs little probabilistic interpretation and has achieved strong performance in the financial domain (e.g., Kuncheva et al., 2001; Rapach et al., 2010).

Table 6 presents the results of combining numerical and textual inputs for our best-performing decision tree, the SGB, the benchmark model LOGIT and the neural networks NN and CNN. Notably, combining numerical and textual inputs using the simpler average ensemble approach (Panel B) yields consistently higher AUC, compared to retraining the models with concatenated inputs (Panel A). When using the concatenated inputs, only SGB(Num+Text) shows a slight yet statistically insignificant ($p = 0.532$) increase in AUC and consistently higher catch rates than using only numerical financial inputs. However, when using an average ensemble approach, both SGB(Num+Text) and NN-CNN(Num+Text) show statistically significant increases in AUC ($p < 0.01$), compared to SGB(Num) and NN-CNN(Num). These results generally suggest that textual data contain incremental information to numerical inputs that helps improve out-of-sample predictions.

Collectively, our results suggest that using machine learning and textual data can improve crash prediction relative to traditional approaches such as a LOGIT with numerical inputs. To provide more intuition into the economic significance of our results, we consider an investor who, depending on risk profile, monitors the top 20% to 30% predictions of a predictive model.[14] Considering a probability threshold of 20% (30%), the benefit of using LOGIT(Num) relative to a random guess is equivalent to the detection of an additional 331 (381) stock price crashes during our sample period. Using SGB(Num+Text) instead would result in detecting an additional 436 (557) stock price crashes relative to

[12] Mai et al. (2019) also test an average embedding model and find that it outperforms a CNN for bankruptcy prediction. Untabulated tests suggest that an average embedding model with word2vec embeddings yields an AUC of 54.90%, which is significantly higher than the 50% of a random model ($p < 0.01$) but statistically indifferent, compared to the performance of the CNN ($p = 0.745$).

[13] For our neural network models, NN(Num) and CNN(Text), this approach is infeasible because both data sources are fed into different model architectures. Hence, to combine numerical and textual inputs for the neural networks, we follow Mai et al. (2019) and concatenate the final hidden layers of NN(Num) and CNN(Text) before feeding the combined neuron layer to a softmax output.

[14] Note that 20% to 30% is a reasonable threshold because stock price crashes occur in 23% of all firm-years.

TABLE 6 Combining numerical and textual inputs.

Panel A: Concatenating inputs								
		AUC versus Model(Num)		*Excess catch rate*				
Models	**AUC**	**Diff.**	***p*-val.**	**10%**	**20%**	**30%**	**40%**	**50%**
LOGIT(Num+Text)	53.34%	−1.96%	<0.01	0.02%	0.76%	1.59%	2.09%	2.68%
SGB(Num+Text)	**56.46%**	**0.20%**	**0.532**	**3.67%**	**5.52%**	**6.76%**	**7.21%**	**7.24%**
DL-CNN(Num+Text)	54.48%	−0.26%	0.572	2.10%	3.28%	4.07%	4.43%	4.53%
Panel B: Average ensemble								
		AUC versus Model(Num)		*Excess catch rate*				
Models	**AUC**	**Diff.**	***p*-val.**	**10%**	**20%**	**30%**	**40%**	**50%**
LOGIT(Num+Text)	54.63%	−0.67%	<0.01	0.02%	0.76%	1.59%	2.09%	2.68%
SGB(Num+Text)	**57.10%**	**0.85%**	**<0.01**	**3.42%**	**5.27%**	**6.74%**	**7.63%**	**7.97%**
DL-CNN(Num-Text)	55.80%	1.07%	<0.01	2.10%	3.28%	4.07%	4.43%	4.53%

Note: This table presents the results of combining numerical and textual inputs. Panel A presents the results of training new models on concatenated inputs. Panel B presents the results of an average ensemble approach that combines two predictions of two separate models using a simple average. "Num" denotes the 37 financial variables defined in Table 2, panel A. "Text" denotes word2vec embeddings for the DL-CNN and TF-IDF weights for the remaining text models. The table reports the AUC and catch rates for varying cut-offs. Differences between AUC scores are tested for statistical significance using the DeLong test. The best-performing models in each panel are presented in bold.

a random guess, which translates to an overall increase of 31.76% (46.15%). Overall, we consider these results to be economically meaningful.

4.3 | Additional analyses

4.3.1 | Model performance over time

In this section, we provide descriptive evidence on whether the results and differences among the models are systematic and persistent over time. Figure 1 illustrates the yearly AUC of LOGIT(Num), SGB(Num) and SGB(Num+Text). The results show that SGB(Num) systematically outperforms LOGIT(Num), supporting the view that machine learning can improve predictions relative to traditional approaches. At the same time, however, SGB(Num+Text) shows consistently higher AUCs compared to SGB(Num), suggesting a systematic improvement due to the consideration of textual inputs. The results further show that textual data are particularly informative in the years following the 2007–2008 financial crisis. This finding may suggest that textual data become more informative than financial data in times of high economic uncertainty. Collectively, we conclude that our main results are consistent across almost all test years and do not diminish over time.

4.3.2 | Inner workings of machine learning models

In this section, we seek to better understand the inner workings of our best-performing machine learning model SGB. We perform permutation feature importance, a model inspection technique that measures the contribution of model inputs to its out-of-sample performance. This approach randomly shuffles values of single predictors and observes the

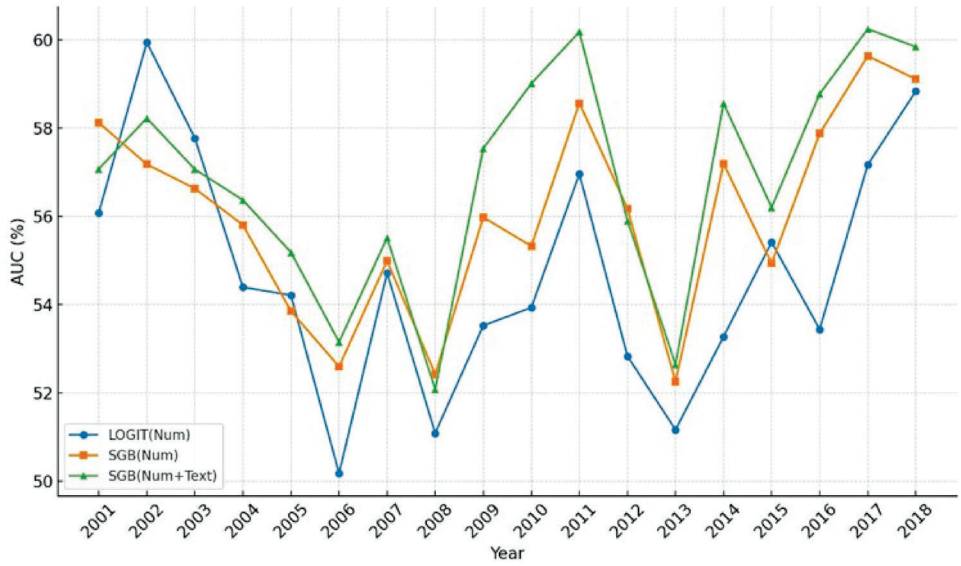

FIGURE 1 Model performance over time. This figure presents yearly area under the receiver operating characteristic (ROC) curve (AUC) scores of crash prediction models. The blue line represents a logistic regression with numerical inputs (LOGIT(Num)). The orange line represents a stochastic gradient boosting (SGB) model with numerical inputs (SGB(Num)). The yellow line represents the average ensemble of a SGB that uses numerical inputs and a SGB that uses textual inputs (SGB(Num+Text)).

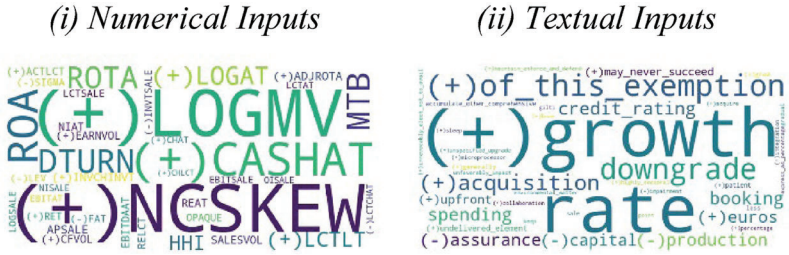

FIGURE 2 Feature importance. This figure presents word clouds of the most important input features. Larger font indicates higher feature importance. The prefix (+) ((−)) indicates that the given feature is significantly more (less) likely to occur in crash firms, compared to non-crash firms ($p < 0.05$). The first plot shows numerical features. The second plot shows textual features.

model's predictive performance decrease (e.g., Breiman, 2001; Chen et al., 2022). We estimate the importance score for each sample year and average the importance values for each predictor.

Figure 2 presents word clouds for the most important (i) numerical inputs and (ii) textual inputs for SGB(Num) and SGB(Text), respectively. A larger font size suggests higher feature importance scores. In addition, we conduct *t*-tests between crash and non-crash firms to identify descriptive differences between predictors. The prefix (+) ((−)) suggests that a given predictor is significantly more (less) pronounced in crash firms relative to non-crash firms ($p < 0.05$).

The results suggest that the model SGB(Num) considers most control variables that are commonly used in the literature to be important predictors of stock price crashes (e.g., Hutton et al., 2009; Kim et al., 2011a, 2011b). Negative firm-specific return skewness (*NCSKEW*) and firm size (*LOGMV*) are the most important numerical predictors. Other common variables, such as return on asset (*ROA*), detrended turnover (*DTURN*) and book-to-market ratios (*MTB*) are

also considered important by the SGB(Num). Interestingly, we find that financial opacity (*OPAQUE*), a standard control variable in the literature (Hutton et al., 2009), only ranks as the 22nd most important numerical predictor, suggesting relatively low importance.

Turning to textual predictors, we consider the most important terms used by SGB(Text) to predict stock price crashes. We find that the word "growth" is the most important predictor in MD&A text and that firms writing more about growth are more likely to be crash firms as indicated by the suffix (+). Moreover, terms related to investments such as "acquisition," "spending" and "production" help our model predict stock price crashes. This notion is consistent with the view that high-growth firms are more likely to experience stock price crashes (e.g., Chen et al., 2001; Hutton et al., 2009). Moreover, in line with previous research suggesting that tone is an important determinant of crash risk (e.g., Fu et al., 2021; Reichmann, 2023), we find that SGB(Text) also captures terms such as "downgrade," "may never succeed" and "unfavorably impact" to distinguish crash from non-crash observations.

Collectively, our results corroborate that our models consider reasonable signals to predict stock price crashes. However, we caution the reader against interpreting the results as indicative of the causal influence of predictors (Chen et al., 2022). Instead, we aim to foster transparency and visualize underlying data inputs that drive the predictive performance of our main models.

4.3.3 | Sensitivity to serial crashes

In our main test, we modify the data used to train the model by setting the values of observations to zero if the same firm is identified as a crash observation in the following year's hold-out sample. This design choice mitigates concerns that our models simply predict firms instead of firm-years, thereby improving the robustness of our main inferences. In this section, we test the sensitivity of our models to this design choice.

Table 7 shows the results of estimating all main models *without* correcting for serial crashes in the training data. The results in panel A suggest that failing to correct for serial crashes in the training data inflates model performance for almost all models. This effect is strongest for RF(Num)—an increase in AUC of 3.76%, almost doubling its performance relative to a random guess. By contrast, the AUC of SGB(Num) only increases by 1.50%, suggesting that the model is more likely to identify generalizable patterns in financial data, compared to RF(Num).

Interestingly, turning to textual inputs, the results in panel B suggest that the model performance of less flexible models like LOGIT(Text) and SVM(Text) significantly increase when failing to correct for serial crashes, suggesting that these models are more likely to identify firm characteristics in text rather than generalizable language that predicts stock price crashes. SGB(Text) is the only model with textual inputs that is not affected by the correction of serial crashes. Our results suggest that failing to correct for serial crashes can substantially inflate the performance of various model architectures. Our results should caution future research that examines the predictability of stock price crashes.

5 | CONCLUSION

In this study, we use machine learning methods drawn from the wider literature in computer science to predict stock price crashes. We test various models that incorporate numerical and textual inputs from 10-K disclosures. We find that a LOGIT model as a traditional regression approach serves as a strong benchmark. We further find that a SGB model based on decision trees systematically improves the prediction of one-year-ahead stock price crashes. Our machine learning models are most valuable for out-of-sample predictions using suitable combinations of numerical and textual inputs. We find that the most powerful textual predictors from MD&A sections are such words as "growth," "acquisition" and "spending." The results should be of interest to academics, practitioners and investors who aim to better understand the predictors of stock price crashes. For instance, machine learning algorithms can help investors

TABLE 7 Sensitivity to recoding serial crashes.

Panel A: Numeric models				
Models	**AUC (main test)**	**AUC (w/o recoding)**	**Diff.**	***p*-val**
LOGIT(Num)	55.30%	55.87%	+0.57%	<0.01
SVM(Num)	52.90%	53.33%	+0.43%	<0.01
RF(Num)	53.42%	57.18%	+3.76%	<0.01
SGB(Num)	56.26%	57.76%	+1.50%	<0.01
NN(Num)	54.73%	54.97%	+0.24%	0.4026
Panel B: Textual models				
Models	**AUC (main test)**	**AUC (w/o recoding)**	**Diff.**	***p*-val**
LOGIT(TextChar)	53.11%	53.62%	+0.51%	<0.01
LOGIT(Text)	52.84%	57.32%	+4.48%	<0.01
SVM(Text)	49.39%	55.61%	+6.22%	<0.01
RF(Text)	56.17%	57.93%	+1.76%	<0.01
SGB(Text)	56.18%	56.30%	+0.12%	0.7087
CNN(Text)	54.77%	55.44%	+0.67%	0.0144

Note: This table presents the results of predicting 1-year-ahead stock price crashes without recoding serial crash observations in the training set for model estimation. Panel A reports the results of models using numerical inputs. Panel B reports the results of models using textual inputs. "Num" denotes the 37 financial variables defined in Table 2, panel A. "TextChar" denotes textual characteristics defined in Table 2, panel B. "Text" denotes word2vec embeddings for the CNN and TF-IDF weights for the remaining text models. The table reports the AUC. Differences between AUC scores are tested for statistical significance using the DeLong test.

to position their portfolios against future stock price crashes and thus help them make better-informed investment decisions.

While our study provides early evidence on the use of machine learning methods for predicting stock price crashes, more evidence is needed. Stock price crashes are prevalent in international financial markets, yet the incremental predictive power of other (mandatory) firm disclosures and data sources, such as sustainability reports, social media data and consumer product reviews for future stock price crashes remain largely unexplored (e.g., Al Guindy et al., 2024; El-Haj et al., 2020; Jin, 2023).

ACKNOWLEDGMENTS

This paper is partially based on Chapter 2 of Doron Reichmann's dissertation completed at Ruhr University Bochum, and Chapter 2 was previously titled "Predicting Firm-Specific Stock Price Crashes." We thank participants at the Leipzig Banking and Finance Workshop, HVB Seminar in Paderborn, FAACT and ifu Seminar in Bochum, German Academic Association for Business Research Conference, University of Erlangen-Nürnberg, two anonymous reviewers for the EAA Conference, Andrew Stark (senior editor) and an anonymous reviewer at the *Journal of Business Finance & Accounting* and Jonas Ewertz, Petroula Glachtsiou, Thorsten Knauer, Charlotte Knickrehm, Johannes Kriebel, Rouven Möller, Martin Nienhaus, Stephan Paul, Bernhard Pellens, Matthias Pelster, Andreas Pfingsten, Fleming Schmidt-Skipiol, André Uhde and Gregor Weiss for helpful comments on earlier versions of this paper.

Open access funding enabled and organized by Projekt DEAL.

DATA AVAILABILITY STATEMENT

Data are available from the sources as cited in the text.

ORCID

Doron Reichmann https://orcid.org/0000-0002-2196-1746

REFERENCES

Al Guindy, M., Naughton, J. P., & Riordan, R. (2024). The evolution of corporate twitter usage. *Journal of Business Finance & Accounting*, *51*(3-4), 819–845. https://doi.org/10.1111/jbfa.12758

An, Z., Chen, Z., Li, D., & Xing, L. (2018). Individualism and stock price crash risk. *Journal of International Business Studies*, *49*(9), 1208–1236. https://doi.org/10.1057/s41267-018-0150-z

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, *54*(6), 124–136. https://doi.org/10.1016/j.ejor.2015.05.030

Bali, T. G., Beckmeyer, H., Mörke, M., & Weigert, F. (2023). Option return predictability with machine learning and big data. *The Review of Financial Studies*, *36*(9), 3548–3602. https://doi.org/10.1093/rfs/hhad017

Bertomeu, J., Cheynel, E., Floyd, E., & Wenqiang, P. (2021). Using machine learning to detect misstatements. *Review of Accounting Studies*, *26*, 468–519. https://doi.org/10.1007/s11142-020-09563-8

Bianchi, D., Büchner, M., & Tamoni, A. (2020). Bond risk premiums with machine learning. *The Review of Financial Studies*, *34*(2), 1046–1089. https://doi.org/10.1093/rfs/hhaa062

Bochkay, K., Brown, S. V., Leone, A. J., & Tucker, J. W. (2023). Textual analysis in accounting: What's next? *Contemporary Accounting Research*, *40*(2), 765–805. https://doi.org/10.1111/1911-3846.12825

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Brown, S. V., & Tucker, J. W. (2011). Large-sample evidence on firms' year-over-year MD&A modifications. *Journal of Accounting Research*, *49*(2), 309–346. https://doi.org/10.1111/j.1475-679X.2010.00396.x

Butaru, F., Chen, Q., Clark, B., Das, S., Lo, A. W., & Siddique, A. (2016). Risk and risk management in the credit card industry. *Journal of Banking & Finance*, *72*, 218–239. https://doi.org/10.1016/j.jbankfin.2016.07.015

Chen, J., Hong, H., & Stein, J. C. (2001). Forecasting crashes: Trading volume, past returns, and conditional skewness in stock prices. *Journal of Financial Economics*, *61*(3), 345–381. https://doi.org/10.1016/S0304-405X(01)00066-6

Chen, X., Cho, Y. H. (Tony), Dou, Y., & Lev, B. (2022). Predicting future earnings changes using machine learning and detailed financial data. *Journal of Accounting Research*, *60*(2), 467–515. https://doi.org/10.1111/1475-679X.12429

Cheng, C., Jones, S., & Moser, W. J. (2018). Abnormal trading behavior of specific types of shareholders before US firm bankruptcy and its implications for firm bankruptcy prediction. *Journal of Business Finance & Accounting*, *45*(9-10), 1100–1138. https://doi.org/10.1111/jbfa.12338

Dechow, P. M., Ge, W., Larson, C. R., & Sloan, R. G. (2011). Predicting material accounting misstatements. *Contemporary Accounting Research*, *28*(1), 17–82. https://doi.org/10.1111/j.1911-3846.2010.01041.x

DeLong, E. R., DeLong, D. M., & Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, *44*(3), 837–845.

Du, Z., Huang, A. G., Wermers, R., & Wu, W. (2022). Language and domain specificity: A Chinese financial sentiment dictionary. *Review of Finance*, *26*(3), 673–719. https://doi.org/10.1093/rof/rfab036

Eaglesham, J. (2013, May 27). SEC refocuses on accounting fraud. *Wall Street Journal*. http://online.wsj.com/article/SB10001424127887324125504578509241215284044.html

El-Haj, M., Alves, P., Rayson, P., Walker, M., & Young, S. (2020). Retrieving, classifying and analysing narrative commentary in unstructured (glossy) annual reports published as PDF files. *Accounting and Business Research*, *50*(1), 6–34. https://doi.org/10.1080/00014788.2019.1609346

El-Haj, M., Rayson, P., Walker, M., Young, S., & Simaki, V. (2019). In search of meaning: Lessons, resources and next steps for computational analysis of financial discourse. *Journal of Business Finance & Accounting*, *46*(3-4), 265–306. https://doi.org/10.1111/jbfa.12378

Ertugrul, M., Lei, J., Qiu, J., & Wan, C. (2017). Annual report readability, tone ambiguity, and the cost of borrowing. *The Journal of Financial and Quantitative Analysis*, *52*(2), 811–836. https://doi.org/10.1017/S0022109017000187

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367–378. https://doi.org/10.1016/S0167-9473(01)00065-2

Fu, X., Wu, X., & Zhang, Z. (2021). The information role of earnings conference call tone: Evidence from stock price crash risk. *Journal of Business Ethics*, *173*(3), 643–660. https://doi.org/10.1007/s10551-019-04326-1

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223–2273. https://doi.org/10.1093/rfs/hhaa009

Hinton, G. E., Osindero, S., & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, *18*(7), 1527–1554.

Hong, H. A., Kim, J. B., & Welker, M. (2017). Divergence of cash flow and voting rights, opacity, and stock price crash risk: International evidence. *Journal of Accounting Research*, *55*(5), 1167–1212. https://doi.org/10.1111/1475-679X.12185

Hutton, A. P., Marcus, A. J., & Tehranian, H. (2009). Opaque financial reports, R^2, and crash risk. *Journal of Financial Economics*, *94*(1), 67–86. https://doi.org/10.1016/j.jfineco.2008.10.003

Jin, L., & Myers, S. C. (2006). R^2 around the world: New theory and new tests. *Journal of Financial Economics*, *79*(2), 257–292. https://doi.org/10.1016/j.jfineco.2004.11.003

Jin, S. (2023). More than words: Can tone of consumer product reviews help predict firms' fundamentals? *Journal of Business Finance & Accounting*, *50*(9-10), 1910–1942. https://doi.org/10.1111/jbfa.12680

Jones, S., & Hensher, D. A. (2004). Predicting firm financial distress: A mixed logit model. *The Accounting Review*, *79*(4), 1011–1038. https://doi.org/10.2308/accr.2004.79.4.1011

Jones, S., & Hensher, D. A. (2007). Modelling corporate failure: A multinomial nested logit analysis for unordered outcomes. *The British Accounting Review*, *39*(1), 89–107. https://doi.org/10.1016/j.bar.2006.12.003

Jones, S., Johnstone, D., & Wilson, R. (2015). An empirical evaluation of the performance of binary classifiers in the prediction of credit ratings changes. *Journal of Banking & Finance*, *56*, 72–85. https://doi.org/10.1016/j.jbankfin.2015.02.006

Jones, S., Johnstone, D., & Wilson, R. (2017). Predicting corporate bankruptcy: An evaluation of alternative statistical frameworks. *Journal of Business Finance & Accounting*, *44*(1-2), 3–34. https://doi.org/10.1111/jbfa.12218

Jones, S., Moser, W. J., & Wieland, M. M. (2023). Machine learning and the prediction of changes in profitability. *Contemporary Accounting Research*, *40*(4), 2643–2672. https://doi.org/10.1111/1911-3846.12888

Kaya, D., & Seebeck, A. (2019). The dissemination of firm information via company register websites: Country-level empirical evidence. *Journal of Accounting & Organizational Change*, *15*(3), 382–429. https://doi.org/10.1108/JAOC-03-2018-0023

Kim, C. (Francis), Wang, K., & Zhang, L. (2019). Readability of 10-K reports and stock price crash risk. *Contemporary Accounting Research*, *36*(2), 1184–1216. https://doi.org/10.1111/1911-3846.12452

Kim, J.-B., Li, Y., & Zhang, L. (2011a). CFOs versus CEOs: Equity incentives and crashes. *Journal of Financial Economics*, *101*(3), 713–730. https://doi.org/10.1016/j.jfineco.2011.03.013

Kim, J.-B., Li, Y., & Zhang, L. (2011b). Corporate tax avoidance and stock price crash risk: Firm-level analysis. *Journal of Financial Economics*, *100*(3), 639–662. https://doi.org/10.1016/j.jfineco.2010.07.007

Kuncheva, L. I., Bezdek, J. C., & Duin, R. P. W. (2001). Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition*, *34*(2), 299–314. https://doi.org/10.1016/S0031-3203(99)00223-X

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

Lewis, C., & Young, S. (2019). Fad or future? Automated analysis of financial text and its implications for corporate reporting. *Accounting and Business Research*, *49*(5), 587–615. https://doi.org/10.1080/00014788.2019.1611730

Li, K., Mai, F., Shen, R., & Yan, X. (2021). Measuring corporate culture using machine learning. *The Review of Financial Studies*, *34*(7), 3265–3315. https://doi.org/10.1093/rfs/hhaa079

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, *18*(1), 6765–6816.

Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *The Journal of Finance*, *66*(1), 35–65. https://doi.org/10.1111/j.1540-6261.2010.01625.x

Loughran, T., & McDonald, B. (2014). Measuring readability in financial disclosures. *The Journal of Finance*, *69*(4), 1643–1671. https://doi.org/10.1111/jofi.12162

Mai, F., Tian, S., Lee, C., & Ma, L. (2019). Deep learning models for bankruptcy prediction using textual disclosures. *European Journal of Operational Research*, *274*(2), 743–758. https://doi.org/10.1016/j.ejor.2018.10.024

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient estimation of word representations in vector space*. https://doi.org/10.48550/arxiv.1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). *Distributed representations of words and phrases and their compositionality*. https://doi.org/10.48550/arxiv.1310.4546

Peat, M., & Jones, S. (2012). Using neural nets to combine information sets in corporate bankruptcy prediction. *Intelligent Systems in Accounting, Finance and Management*, *19*(2), 90–101. https://doi.org/10.1002/isaf.334

Rapach, D. E., Strauss, J. K., & Zhou, G. (2010). Out-of-sample equity premium prediction: combination forecasts and links to the real economy. *Review of Financial Studies*, *23*(2), 821–862. https://doi.org/10.1093/rfs/hhp063

Reichmann, D., & Reichmann, M. (2022). Predicting firm-specific stock price crashes. In D. Reichmann, *Qualitative disclosures and capital market consequences—A textual analysis approach*. [Doctoral dissertation, Ruhr University Bochum].

Reichmann, D. (2023). Tone management and stock price crash risk. *Journal of Accounting and Public Policy*, *42*(6), 107155. https://doi.org/10.1016/j.jaccpubpol.2023.107155

Reichmann, D., Möller, R., & Hertel, T. (2022). Nothing but good intentions: The search for equity and stock price crash risk. *Journal of Business Economics*, *92*(9), 1455–1489. https://doi.org/10.1007/s11573-022-01085-w

Schmidt, P. S., von Arx, U., Schrimpf, A., Wagner, A. F., & Ziegler, A. (2019). Common risk factors in international stock markets. *Financial Markets and Portfolio Management*, *33*, 213–241. https://doi.org/10.1007/s11408-019-00334-3

Shumway, T. (2001). Forecasting bankruptcy more accurately: A simple hazard model. *The Journal of Business*, *74*(1), 101–124. https://doi.org/10.1086/209665

Vapnik, V. N. (1998). *Statistical learning theory*. Wiley-Interscience.

Wu, K., & Lai, S. (2020). Intangible intensity and stock price crash risk. *Journal of Corporate Finance*, *64*, 101682. https://doi.org/10.1016/j.jcorpfin.2020.101682

How to cite this article: Kaya, D., Reichmann, D., & Reichmann, M. (2025). Out-of-sample predictability of firm-specific stock price crashes: A machine learning approach. *Journal of Business Finance & Accounting*, *52*, 1095–1115. https://doi.org/10.1111/jbfa.12831

APPENDIX

A.1 | Word2vec validation

In this section, we ensure the validity of our word2vec embeddings (Reichmann & Reichmann, 2022). Specifically, we provide descriptive evidence on how different word groups occupy different locations in the vector space. Figure A1 shows word2vec embeddings in a three-dimensional scatter plot, which suggests that closely related words such as "profit," "margin" and "sales" occupy close locations. Further, word groups related to governance, such as "independent auditor" and "internal control," tend to be closer in the vector space. We infer that word2vec embeddings identify reasonable semantic similarities (Reichmann & Reichmann, 2022).

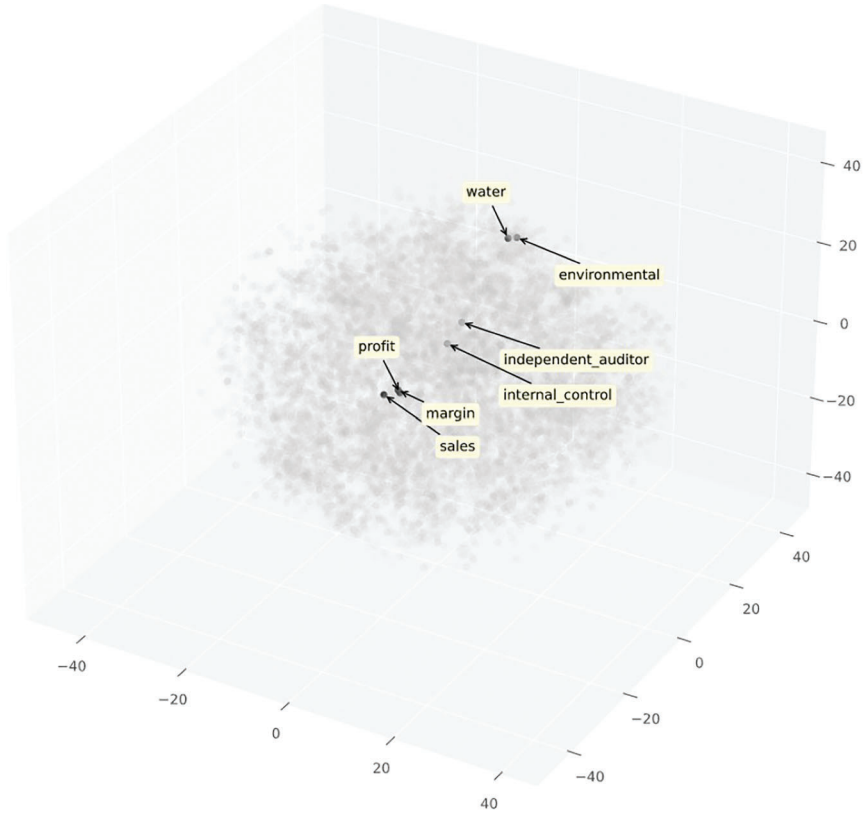

FIGURE A1 Visualization of word2vec embeddings. This figure presents a visualization of our word2vec embeddings. We employ t-distributed stochastic neighbor embedding techniques to visualize the 300-dimensional word2vec embeddings in a three-dimensional scatter plot. The dots represent the reduced vector representations of words and phrases contained in our sample of MD&A of 10-K filings ($n = 39{,}583$).

A.2 | Model hyperparameters

This section presents the hyperparameters tested to train our machine learning models. For the LOGIT and SVM, we follow the specifications of Mai et al. (2019) without performing further optimization. For the decision trees, RF and SGB, we choose similar hyperparameters as in Chen et al. (2022) and use grid search to identify optimal hyperparameters. Similar to Gu et al. (2020), we test different layer depths and the number of neurons per layer for both the NN and CNN and implement early-stopping to prevent overfitting during training. In addition, we also test a range of activation functions and hyperparameters that are specific to the CNN. Because training the neural networks is computationally

expensive, we use the Hyperband approach to tune hyperparameters that aims to speed up random search through adaptive resource allocation and early-stopping (e.g., Li et al., 2017).

Model	Hyperparameters	Optimization
Logistic regression (LOGIT)	L1 regularization	–
Support vector machines (SVM)	Radial basis function kernel	–
Random forest (RF)	# trees: 500, 600, 700, …, 2,000	Grid search
	Max features: 110, 111, 112, …, 120	
	Min. # of obs. in a leaf: 10	
	Bagging: 0.5	
Stochastic gradient boosting (SGB)	# trees: 500, 600, 700, …, 2,000	Grid search
	Learning rate: 0.005, 0.01, 0.05	
	Max. depth: 1, 2, 3, 4	
	Min. # of obs. in a leaf: 10	
	Bagging: 0.5	
Neural network (NN)	# hidden layer: 1,2,3,4,5	Hyperband
	# neurons in hidden layers: 50, 100, 200, 300 400, 500	
	Activation function (every node except final): relu, tanh, sigmoid	
	Regularizer: float in [0, 0.0001]	
Convolutional neural network (CNN)	# hidden layer: 1,2,3,4,5	Hyperband
	# neurons in hidden layers: 50, 100, 200, 300 400, 500	
	# filter: 200, 250, 300, …, 500	
	kernel size: 2,4,6,8, …, 20	
	Embedding dimension (word2vec): 50, 100, 200, 300	
	Activation function: relu, tanh, sigmoid	
	Regularizer: float in [0, 0.0001]	
NN-CNN (concat layer)	# neurons in hidden layer: 50, 100, 200, 300 400, 500	Hyperband

Note: This table presents the hyperparameters and optimization techniques used to train our machine learning models.