Machine Learning-Based Financial Statement Analysis*

Amir Amel-Zadeh[†] Jan-Peter Calliess[‡] Daniel Kaiser[§]
Stephen Roberts[¶]

January 15, 2020

Abstract

This paper explores the application of machine learning methods to financial statement analysis. We investigate whether a range of models in the machine learning repertoire are capable of forecasting the sign and magnitude of abnormal stock returns around earnings announcements based on financial statement data alone. We find random forests and recurrent neural networks to outperform deep neural networks and linear models such as OLS and Lasso. Using the models' predictions in an investment strategy we find that random forests dominate all other models and that non-linear methods perform relatively better for predictions of extreme market reactions, while the linear methods are relatively better in predicting moderate market reactions. Analysing the underlying economic drivers of the performance of the random forests, we find that the models select as most important predictors accounting variables commonly used to forecast free cash flows and firm characteristics that are known cross-sectional predictors of stock returns.

Keywords: Financial statement analysis, fundamental value, machine learning, earnings announcement, accounting-based anomalies, prediction

JEL Codes: G12, G14, M41

*The authors would like to thank participants at the 3rd Conference on Intelligent Information Retrieval in Accounting and Finance, Shenzhen, and the Oxford-Man Institute brown-bag seminar for helpful comments. Amir Amel-Zadeh acknowledges financial support from the Saïd Business School Foundation and from the Oxford University Press John Fell Fund.

[†]Corresponding author, Saïd Business School, University of Oxford, amir.amelzadeh@sbs.ox.ac.uk.
[‡]Oxford-Man Institute of Quantitative Finance, University of Oxford, jan-peter.calliess@oxford-man.ox.ac.uk.
[§]Oxford-Man Institute of Quantitative Finance, University of Oxford, daniel.kaiser@mansfield.ox.ac.uk.
[¶]Oxford-Man Institute of Quantitative Finance, University of Oxford, steve.roberts@oxford-man.ox.ac.uk.

1 Introduction

Financial statement (or fundamental) analysis identifies information contained in financial reports, particularly the main financial statements, that is useful in assessing the value of a company. A large literature in accounting examines whether components of financial statements can help investors make better investment decisions, typically by assessing whether these components are useful in forecasting earnings or security returns (Foster et al. 1984; Ou and Penman 1989; Lev and Thiagarajan 1993; Abarbanell and Bushee 1997, 1998; Piotroski et al. 2000).[1] Generally, this research is premised on the notion that careful analysis of past financial statements is fruitful in uncovering information that is not yet reflected in stock prices. In investment practice, value investors engage in fundamental analysis to estimate the intrinsic value of a company which they then compare to market prices informing their decision to buy or sell a stock.

The main motivation for financial statement analysis research is a long-standing body of research that finds that security prices fail to immediately capture publicly available accounting information (Ball and Brown 1968; Bernard and Thomas 1989; Sloan 1996). Hence this line of accounting research (and investment practice) aims to identify mispriced securities by finding financial statement components that lead to better forecasts of earnings or stock returns. In their seminal paper, Ou and Penman (1989) examine whether past financial statements contain information that is useful in predicting future earnings. Identifying 28 financial statement ratios out of 68 potential candidates by statistical means, they use a logit model to estimate the probability of one-year ahead earnings increases for a sample of U.S. firms. The study then derives a profitable investment strategy by buying (selling) stocks based on the predictions of their probability measure that earnings will increase (decrease) in the subsequent 12 months. A flurry of subsequent studies assess the out-of-sample performance of the Ou and Penman (1989) measure (Bernard et al. 1997; Abarbanell and Bushee 1998; Bird et al. 2001), predict stock returns directly instead of earnings (Holthausen and

[1] See Kothari (2001) and Richardson et al. (2010) for reviews of the literature.

Larcker 1992; Lev and Thiagarajan 1993), and extend the analysis beyond financial statements to contextual variables (Piotroski et al. 2000; Beneish et al. 2001; Mohanram 2005).

In this paper we investigate whether machine learning methods can be employed for the analysis of past financial statements capable of forecasting the sign and magnitude of stock returns around future earnings announcements. Recent advances in the field of machine learning have led to the successful deployment of machine and deep learning algorithms for prediction tasks for a variety of applications outside of finance and accounting. Accounting researchers have only recently adopted machine learning methods for problems relating to financial reporting such as fraud detection (Perols 2011), bankruptcy prediction (Barboza et al. 2017), and textual analysis (Li 2008, 2010).[2]. Financial statement analysis research might particularly lend itself to machine learning-based approaches due to the high dimensionality of the information contained in general purpose financial statements and their inter-temporal dependencies. These features of financial statement data have been shown to present challenges to investors and financial analysts when predicting future earnings resulting in security mispricing (Bernard and Thomas 1990; Sloan 1996; Richardson et al. 2005; Wahlen and Wieland 2011).[3]

An advantage of using machine-learning methods for financial statement analysis is that these algorithms can be trained to choose the most promising accounting variables for the prediction task defined by the researcher and learn (non-linear) relationships between the variables from training on large amounts of data. Prior research has generally used statistical methods such as coefficient significance and step-wise linear regression for choosing candidate fundamental variables (Ou and Penman 1989; Holthausen and Larcker 1992; Yan and Zheng 2017) or applied economic intuition to narrow the choice to a small set of predictor variables (Lev and Thiagarajan 1993; Abarbanell and Bushee 1997; Piotroski et al. 2000). Machine learning algorithms offer variable selection and dimensionality reduction techniques

[2]Several recent studies in finance explore the asset pricing applications of machine learning models (Gu et al. 2018; Chen et al. 2019; Gu et al. 2019)

[3]Presumably, machine learning models are also capable of examining larger quantities of data in shorter time than human analysts

by reducing redundant variation among variables and contain methods to avoid overfitting problems. That is, modern machine learning methods are well suited for prediction tasks involving high-dimensional inputs with unknown functional forms.

Of particular interest to this study is the question of, whether besides the direction, also the magnitude of the future market reaction can be predicted. Skinner and Sloan (2002) and Kinney et al. (2002) find that already minor misses of expectations can yield significant negative stock reactions, and Kasznik and McNichols (2002) show that minor positive surprises lead to disproportionate positive price reactions. The prior evidence thus points to significant non-linearities suggesting that machine learning methods might outperform traditional linear models in this forecasting domain. Specifically, the prior literature has found an S-shaped relationship between unexpected earnings and abnormal returns particularly in the extreme tails of the unexpected earnings distribution (Freeman and Tse 1992; Kinney et al. 2002). This suggests that non-linear methods might perform better at predicting large absolute abnormal returns, while linear methods should perform relatively well for more moderate market reactions.

In this study, we therefore explore a range of linear and non-linear machine-learning models to examine whether the market reaction to quarterly earnings announcements is predictable using only past quarterly financial statement information. This study is exploratory in the sense that we do not test an economic theory of stock returns around earnings announcements. Instead, we test whether stock returns can be predicted from a large set of past financial statement data using machine learning methods. However, in doing so, we not only provide first evidence on the usefulness of modern machine learning methods for financial statement analysis, but also provide insights into whether the "machine-learned" relationships between the set of financial statement variables and stock returns align with established relationships and accounting-based regularities found in prior accounting research.

In contrast to the prior literature that uses fundamental analysis to predict fiscal year-end earnings or annual stock returns, this study focuses on the prediction of abnormal stock

returns around earnings announcements for several reasons.

First, the common underlying premise in fundamental analysis research is that market prices can deviate from intrinsic value as market participants either underreact or overreact to available information (Desai et al. 2004; Kothari et al. 2006), but that eventually prices revert back to their intrinsic value. Corrections of security mispricings will only occur when new information is released that causes market participants to revise their prior beliefs. This can happen either slowly as more information becomes available over time or through a catalyst such as the release of new information during an earnings announcement. Earnings announcements relate to financial events of prior periods and are thus more likely to contain information that affects investors' cash flow expectations. Price corrections are therefore more likely to occur during earnings announcements (Bernard et al. 1997).

Second, as accounting earnings are related to future dividends and are valued by investors (Collins and Kothari 1989), earnings announcements that generate surprises normally lead to large abnormal reactions (Bartov et al. 2002; Kinney et al. 2002; Skinner and Sloan 2002). These should be predictable if past financial statement information is related to future earnings surprises that signal changes in dividends. The aim of this study is thus to examine the predictability of the market's reaction to earnings announcements based on past financial statement information.

Third, a long-standing debate in the literature is whether findings of predictable stock returns based on accounting variables are evidence of mispricing, and hence a violation of efficient markets, or rather the result of omitted systematic time-varying risk factors (Fama 1970; Hirshleifer et al. 2012).[4] Concentrating our prediction on a short event-window around earnings announcements ensures that measurement errors related to risk premia are likely small.[5] Furthermore, as Bernard, Thomas, and Wahlen suggest *"if one can predict not*

[4]A third option, discussed in the finance literature, is that the findings are the result of data mining (Harvey et al. 2016; McLean and Pontiff 2016)

[5]Although risk premia can change during events such as earnings announcements (Ball and Kothari 1991; Savor and Wilson 2016), the magnitudes of abnormal returns around earnings announcements related to accounting-based anomalies are too large to be likely the result of temporary changes in discount rates (Engelberg et al. 2018).

only the signs of abnormal returns around subsequent information releases, but also their magnitude, the evidence is particularly difficult to explain except as the product of mispricing (Bernard et al. 1997, p. 96)."

We train and evaluate a range of models, including OLS, Least Absolute Shrinkage and Selection Operator (LASSO), Random Regression Forests, a Deep Neural Network and a Recurrent Neural Network on a large panel of financial statement data. Using these models, we forecast the market reaction to quarterly earnings announcements and compare their relative out-of-sample prediction performances in an expanding window setup for a comprehensive sample of publicly traded stocks in the U.S. for the quarters between 1990 and 2018. To illustrate the potential profitability of these forecasts, we back-test trading strategies based on long-short portfolio positions.

While typically key components of financial statements are released during earnings announcements at fiscal quarter-end, the complete financial statements are usually not available at that point in time. To avoid look-ahead bias we therefore allow only financial statement variables of the four quarters before the quarter for which earnings are announced to enter the prediction equations. Furthermore, at any point in time during the sample period it is important to only use past data to train the models, while prediction performance should be evaluated using 'unseen' future data points. We therefore progress through time in sliding windows of five quarters. At each point in time, we use financial statement variables from quarters t-4 to t-1 to predict the market reaction to the earnings announcement for quarter t. We train the models on the expanding window of past quarters while evaluating them on their prediction accuracy for the next upcoming quarter.

Overall, our main findings show that machine learning methods show promise for being able to forecast the sign and magnitude of abnormal returns around earnings announcements. We assess the models based on their prediction performance across various thresholds of the absolute magnitude of the market reaction. All models show better than random accuracy in their prediction of the direction and threshold magnitude. This is surprising given that

the predictions are solely based on past financial statement data. In their predictions of the magnitude of the reaction the models exhibit mean squared errors of 0.055-0.07. Among the various methods Random Forests prove to be the best performing models in terms of producing the highest prediction accuracy and trading returns. For example, when predicting absolute market reactions above a threshold of 20% the Random Forests have an average accuracy of 59% and generate average buy-and-hold abnormal returns (BHAR) of 10.4% per quarter while trading on average 31 positions (long and short) per quarter. The model performs even better with a mean quarterly BHAR of 13.2% when predicting more extreme absolute market reactions of over 50%.

The relatively high abnormal returns might not necessarily be attainable in practice, however. We find that the absolute magnitudes of abnormal returns during announcements, while being positively associated with prediction accuracy, are negatively associated with firm size and stock price. That is, a large part of the performance is concentrated among firms that likely have lower liquidity and higher transaction costs, particularly when predicting extreme market reactions. For example, we find that a large part of the positive performance when using the Random Regression Forest for predictions of more extreme BHARs is concentrated among nano-cap stocks.

We deliberately apply a "kitchen sink" approach to financial statement analysis by including a large set of balance sheet, income statement and cash flow statement variables in the prediction models without any economic rationale. In doing so we allow the models to learn from the data over a time span of three decades what variables and variable combinations work best for the predictions of stock returns around earnings announcements. Moreover, Random Forests, the best performing models in our setting, allow assessments of the importance of particular variables for the prediction enabling us to test whether the models' variable selections mirror economic intuition.

The variable selections by the Random Forests follow economic intuition surprisingly well. Specifically, the models select accounting components that are required to estimate free

cash flows, i.e., earnings, components of changes in net working capital, CAPEX and other changes in assets. That is, the models assign the highest importance to known drivers of fundamental value to make their predictions. Among these are also firm characteristics that have in the accounting-based anomalies literature been found to be associated with stock returns (Foster et al. 1984; Bernard and Thomas 1989; Lakonishok et al. 1994; Sloan 1996; Abarbanell and Bushee 1998; Richardson et al. 2005; Cooper et al. 2008; Hartzmark and Solomon 2013). Thus the models' variable selections also align with known cross-sectional predictors of stock returns. Furthermore, the Random Forests are able to predict future stock returns around earnings announcements based on past earnings and the time-series properties of earnings. Consistent with commonly practiced year-on-year and previous quarter comparisons by market participants and financial analysts, the models assign a larger weight on previous quarter and previous year financial statement components when making predictions for the current quarter. The latter findings are in line with evidence in Bernard and Thomas (1990) that financial market participants seem to trade around earnings announcements based on comparisons of year-on-year changes in quarterly earnings.

Our results also offer several interesting insights when comparing the different machine learning methods. Consistent with prior findings of a non-linear relationship between unexpected earnings and stock returns for high absolute values of unexpected earnings (Freeman and Tse 1992), the (non-linear) neural net models perform relatively stronger when predicting extreme market reactions while the linear models like OLS and LASSO peform comparably better when predicting moderate returns. Furthermore, despite generating the lowest investment returns over the sample period, the recurrent neural network produces surprisingly stable returns with low volatility and exhibits lower drawdowns during the market downturns of 2002-03 and 2008-09. We leave it for future research to examine the properties of various machine learning methods for asset allocations and portfolio construction.

To the best of our knowledge, this paper is among the first to investigate the viability and properties of modern machine learning methods for financial statement analysis. Beyond

making a methodological contribution to fundamental analysis research that began with Ou and Penman (1989), our study also contributes to the value relevance literature that examines how financial reporting outputs relate to firm value (Ohlson 1995; Feltham and Ohlson 1995) and which financial statement items are predictive of stock returns around earnings announcements (Holthausen and Larcker 1992; Lev and Thiagarajan 1993; Yan and Zheng 2017; Piotroski et al. 2000). As such, the paper also sheds light on potential accounting fundamentals associated with the earnings announcement premium and post-earnings announcement drift (Ball and Brown 1968; Bernard and Thomas 1989; Ball and Kothari 1991). The study confirms the findings in the accounting-based anomalies literature of accounting predictors of cross-sectional stock returns in a setting (i.e., earnings announcements) in which risk-based explanations are less likely. The study thus presents further evidence suggesting that market inefficiencies underlying these regularities are rooted in difficulties of investors to incorporate earnings and other financial statement information into prices.

The remainder of the paper is structured as follows. The next Section 2 presents the data and discusses the estimation set-up. Section 3 discusses the research design and provides an overview of the machine learning models as well as the metrics to evaluate them. Section 4 presents the main results. The study concludes with Section 5.

2 Data

2.1 Financial Statement Data

Our source for the financial statement data is Compustat North America. We select variables from the balance sheet, income statement, and cash flow statement in quarterly periodicity for 27,410 firms between 1987 and 2018 resulting in a total of 1,567,486 observations.

A critical problem related to using the Compustat data for machine learning tasks over long horizons is missing values. Due to changes in accounting standards over the long time span of the data set, and different accounting requirements across industries, about 63%

of values in the original data set are missing. Some variables are naturally just reported for particular industries. For instance, about a third of all variables are just used for firms that are identified as financial institutions. Furthermore, there are sets of variables that are usually reported but some industries are exempt.

Figure 1 represents a visualisation of the problem by showing the temporal and cross-sectional structure of these missing values. The vertical axis represents the cross-section of variables and the horizontal axis represents year-quarters starting in Q1 1983 and ending in Q4 2018. The year 2001 is approximately the horizontal center of the figure, when a large set of novel financial statement items have been introduced (e.g. related to other comprehensive income or options-related expenses).

For our analysis we select a set of commonly occurring financial statement variables where at least 50% of values exist from the beginning (in 1990) to the end (in 2018) of our sample period. Among the remaining observations, we then drop all firm-quarter observations for which more than 50% of the variables are missing, and also those for which the values for either *saleq* (total sales) or *atq* (total assets) are missing or zero. This initial selection leaves the sample with 868,125 firm-quarter observations (55% of the initial raw data set) and 121 of originally 720 total available financial variables in Compustat. These variables represent the most common items from the face of the financial statements and include 40 balance sheet variables, 29 cash flow statement variables, and 52 income statement variables. Table 1 lists the Compustat mnemonics.

In our final sample still about 23% of values are missing for our selection of variables. For the use of these variables as features in the machine learning models, these remaining missing values need to be addressed. A common strategy in finance and accounting studies has been to further limit the number of observations so that only observations remain where all variables of interest for the analysis exist. Given the large selection of variables in our setting, this approach is problematic as it would reduce the available sample for training our machine learning models substantially. As machine learning methods derive their excep-

tional predictive ability by being able to learn and synthesise insights from large amounts of data, such a reduction of the sample would be disproportionately severe with respect to the remaining observations relative to the low amount of selected features.

An alternative strategy to sample reduction employed in the literature is the imputation of missing values. There is a large body of literature discussing missing value imputation suggesting model based imputation to be the most promising approach. While statistical imputation imputes values like the mean or median, model based methods usually conduct a dimensionality reduction to find redundant information in variables that can be exploited. In this study we employ a matrix factorisation method called Soft-Impute.[6]

Soft-Impute uses nuclear norm regularisation for matrix completion by iteratively replacing missing values by the figures computed from a soft-thresholded singular value decomposition (Donoho 1995). The algorithm was introduced by Mazumder et al. (2010) and the implementation from the fancyimpute package for the Python programming language was used.[7]

We validate the plausibility of the imputation by comparing the mean and standard deviation of the imputed values for a particular financial statement item to the mean and standard deviation of the existing values. The distributions of the imputed values largely overlap with the distributions of the existing values of the same variable. While likely

[6]We tested several other imputation methods including a neural net-based auto-encoder, MICE and wkNN. The tested auto-encoder option consisted of a denoising auto encoder architecture with a loss function that was adapted to be invariant to missing values (Beaulieu-Jones and Moore 2017). While this approach appeared promising, it fell short as the neural network based method needed its inputs to be normalised to zero mean and unit variance. Once this operation was undone for the model prediction to arrive at the reconstructed values, the errors induced by the method were exacerbated so that not even the existing values were reconstructed to a promising degree from the lower dimensional encoding. The MICE (Azur et al. 2011) approach attempts iterative imputations of every variable through other variables filling in the mean initially. This approach did not yield satisfactory results as groups of missing variables were often replaced by the means of existing variables. The wkNN nearest neighbor imputation (Hechenbichler and Schliep 2004; Beretta and Santaniello 2016) uses values from similar samples to impute missing data. It combines these values for similar samples using a weighing calculated on the mean squared error between the existing features. The approach appeared promising as it is widely used in many studies, however the large amount of missing values and the size of the Compustat data made the computation unfeasible in finite time on the available computational infrastructure.

[7]https://github.com/iskandr/fancyimpute. Originally developed for the problem of recommender systems where unknown ratings are similar to missing values, the approach was devised for the Netflix dataset where the inputs have a dimensionality of 10^6 rows with 10^6 colums where only 0.001% of values are known.

introducing noise into the data, we take comfort in the similar realistic magnitudes of means and variances for the imputations suggesting that these are viable inputs for the remaining 23% of missing values.

To prevent look-ahead bias, the imputation is conducted for every calendar quarter separately by only taking into account data samples that were published before the imputation period. For example, to impute Q1 2001 the only other values used for the imputation are observations in the Compustat dataset belonging to earlier quarters. After each of these quarterly imputation steps, the imputed observations belonging to the quarter are saved and finally combined to form the final sample of 545,387 firm-quarters spanning 107 quarters between Q1 1991 and Q4 2017. Table 2 summarises the final sample construction.

Since neural net based models might have problems with the heterogeneity of variables and to prevent the over-fitting of models on firms of a particular size, something that has been critizised by Greig (1992) with regards to the Ou and Penman (1989) study, we normalise our data based on total sales (*saleq*) and total assets (*atq*). All values associated with the balance sheet are divided by total assets and all values in the income and cash flow statement by total sales. This step not only normalises our values but also controls for (firm) size effects that are identified in financial research as a major factor explaining the cross-section of stock returns (Banz 1981; Fama and French 1992).

2.2 Abnormal Returns

We compute abnormal Buy- and Hold Abnormal Returns (BHARs) using stock price data from CRSP that serve as our dependent variable for the market reaction to earnings announcements in quarter Y_Q. They span the period of one day before the announcement day T denoted as $(T-1)$ up until 30 days after the announcement denoted as $(T+30)$. BHARs are a measure of the actual excess returns an investor would earn over that period and are calculated as

$$\text{BHAR}_{i,(-1;+30)} = \prod_{t=-1}^{+30}(1+R_{i,t}) - \prod_{t=-1}^{+30}(1+R_{m,t}), \quad (1)$$

where $R_{i,t}$ is the stock return of company i and $R_{m,t}$ is the return of the value-weighted CRSP market index on day t.

2.3 Data Window Construction

The training and test samples for our models are constructed using a sliding window of 5 quarters. This window traverses the history of records of all firms in our sample period. Let X_{Q-n} denote a report of firm f that is filed in calendar quarter $Q-n$ where $Q \in \mathbb{N}$ represents an index that pertains to a unique year and calendar quarter combination and n denotes the offset of how many quarters before Q the report was filed. The set of reports spanning the four previous quarters (i.e. X_{Q-4}, X_{Q-3}, X_{Q-2}, X_{Q-1}) act as the independent variables to construct one sample for our models while Y_{Q-0} (i.e., Y_Q) denotes the market reaction in quarter Q that is associated with the dependent variable. The set of all samples associated with quarter Q over all firms constitutes the training and test set of quarter Q. This setup ensures that the financial statements published in Q, which are typically published after the earnings announcement, are not used in the model, constraining the setup to base the prediction of the market reaction on the past point-in-time available financial statements.[8]

2.4 Training and Test Set

To prevent look-ahead bias, it is paramount to use only past data to train models, and unknown future data to evaluate them. The test set is therefore constructed to be out-of-sample and out-of-time. This is achieved by applying an expanding window where by

[8]Since fiscal reporting periods are different across firms, we remap the fiscal quarters to calendar quarters by associating a fiscal quarter with the calendar quarter where at least two months of the operations overlap. If for instance the fiscal quarter Q3 2009 of some firm ranges from November 1st 2008 to January 31st 2009, we would classify the timing as the calendar quarter Q4 2008 filing, as two of the operating months (i.e. November and December) in this fiscal quarter fall into this calendar quarter which runs from October 1st 2009 to December 31st 2009.

traversing the history of quarters from past to present, upcoming quarters are initially used to evaluate the past model, before including them in the training data for models evaluated on future quarters.

For calendar quarter Q4 2008 for instance, only data up to Q3 2008 is used to train the models, whereas the data pertaining to Q4 2008 is taken to evaluate the performance of the models. For the subsequent quarter Q1 2009, the data of quarter Q4 2008 is included in the training data together with the data from earlier quarters.

This approach implies that the models are tested under conditions that prevailed at that point in time and so that the performance can possibly improve over time as more training data becomes available over the sample period of 107 quarters. As a result models trained for later year-quarters, e.g., Q4 2017, can benefit from substantially more training data than models trained for earlier year-quarters, e.g., Q1 1991.[9]

3 Research Design

3.1 Set up

We deliberately apply a "kitchen sink" approach to financial statement analysis by including a large set of balance sheet, income statement and cash flow statement variables in the prediction models without applying any economic rationale to the selection. In doing so we allow the models to learn from the data over a time span of three decades which variables and variable combinations work best for the predictions of stock returns around earnings announcements over a period of 30 days (i.e., the immediate earnings response and the post-

[9]While in general, it can be assumed that more training data leads to better model performance, the time decay of the training data might also play a role. The further in the future the evaluated quarters are, the more weight is put on less recent training data potentially negatively affecting the predictive performance. In robustness tests we therefore varied the time-series length of the training data for the random forest models to include only the previous 4, 12, 20, or 40 quarters and attempted a similar limitation for the neural networks by reusing the trained weights of a past quarter for the subsequent quarter and biasing the weights towards the more recent quarter. Both tests yielded inferior performance than the unconstrained models.

earnings announcement drift). Our setup implies that we only use quantitative information contained in financial statements as independent variables to build the model. As a consequence we ignore other known market-based stock return predictors (e.g. momentum or volatility) and macroeconomic variables (e.g. interest rates, GDP growth, unemployment). Machine learning models such as neural networks are designed to learn combinations and interactions of the raw input variables and potentially synthesise latent economic variables based on the time-series behaviour of the accounting variables. It is nevertheless likely that providing the additional variables or applying a variable selection ex-ante based on economic theory might improve the models' predictive power. However, the purpose of the study is to explore whether the models are able to systematically predict returns even on the basis of raw and unfiltered financial statement data alone.

As the aim is to predict the sign and magnitude of the market reaction we formulate the machine learning task as the prediction of a continuous scalar (i.e. a regression task) that represent the abnormal returns (i.e. BHARs) around earnings announcements using a mean squared error (MSE) as the loss function during training.[10] The benefit of the regression setup is that the models learn from the entire information value of the market reaction by learning to distinguish the whole spectrum of major and minor reactions in both directions.

The regression problems in the various models are set up as the mapping $f : x_Q \mapsto y_Q$, where $x_Q = (X_{Q-4}, .., X_{Q-1})$ with x_Q representing the set of financial statements 4 quarters before quarter Q, and $y_Q = BHAR_Q$ denoting the market reaction in quarter Q. In our setup the learning ability is subsequently evaluated based on the performance of the models in predicting the unknown reactions y_{Q+1} from the future quarter $Q+1$ based on the known financial statements $x_{Q+1} = (X_{Q-3}, .., X_Q)$.

[10] The MSE for the prediction $\hat{y}_i$ and ground truth y_i of n training samples is defined as $\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$

3.2 Machine Learning Models

The machine learning methods we employ in our study represent a repertoire of models that have gained increasing popularity in recent years. They include a Deep Neural Network, a Recurrent Neural Network, and a CART regression forest. We hypothesise that these methods are able to take advantage of the possible non-linear relationship between the range of input variables and the the outcome variables. To benchmark the non-linear machine learning methods, we also model our data with a (linear) ordinary least squares (OLS) regression and a least absolute shrinkage and selection operator (LASSO).

3.2.1 Deep Neural Network

By a Deep Neural Network (DNN) we refer to a feed-forward artificial neural network (ANN) that has more than one hidden layer trained in a supervised fashion.

Similar to other types of models, its purpose is to approximate a function f^* that maps inputs x to a prediction $\hat{y}$ in the form of $\hat{y} = f^*(x_i)$. The functional specification is usually extended by a set of parameters θ of a network $f^*(x;\theta)$ that determine how the f^* conducts this mapping. Subsequently, the learning process also known as *training* of a DNN consists of finding a set of optimal values for θ so that $f^*(x;\theta)$ results in the best functional approximation (Goodfellow et al. 2016). A good function approximation usually means that the function output $\hat{y}$ is close to what is considered the true value y if it is observed, relative to some choice of distance.

In the context of neural networks, θ usually refers to the set of weights w_i and biases b of a model so that the elementary component of a neural net, known as a neuron (Rosenblatt 1958), can be formulated as:

$$\hat{y} = f(b + \sum_{i=1}^{n} x_i w_i) \tag{2}$$

where $\hat{y}$ is the prediction/output, and f is a non-linear activation function (e.g. Sigmoid,

tanh, ELU (see equation 3 below), x_i are the n inputs of the perceptron, w_i are the weights by which the inputs are transformed, and b is the bias of a unit. The intuition for the perceptron has loosely been inspired by biological insight of how synapses work in the brain.

The measure of how well the approximation succeeds is referred to as the loss function of a network. This loss function typically guides the learning process to find the optimal set of parameters through the backpropagation algorithm and a choice of local optimizer (Rumelhart et al. 1986).

The parameters of the DNN are found during the training process employing the Adam optimizer (Kingma and Ba 2014). Also known as Adaptive Moment Estimation, Adam is an extension of RMSProp that uses running averages of the first and second moments of the gradient to adapt the learning rate for each weight of the neural network. The learning rate is a hyperparameter of the training process that controls how much the weights are updated with respect to the loss gradient. We use a learning rate value of 0.00005.[11]

A layer of a DNN consists of multiple neurons (see eq. 2) with non-linear activation functions that implement a function mapping of the outputs of the previous layer to the inputs of the subsequent layer. Generally, layers take the outputs of the previous layers as inputs while providing their own output for a subsequent layer as demonstrated in figure 2. A technique that is applied between the layers of the DNN used in this study is batch-normalisation (Ioffe and Szegedy 2015). It addresses the problem of internal covariance shift in which inputs of hidden layers follow different distributions. This is of concern due to the heterogeneity of our sample firms that make up the training set. For every batch during training, batch-normalisation standardises the inputs to zero mean and unit variance.

Having tried various numbers of layers and sizes of layers empirically, we decided on a deep neural net consisting of three hidden layers that follow the input layer of dimension 484

[11] As the loss function that is optimised is usually a non-convex function of the DNN parameters θ, it is important to note that it can have many local minima. Therefore, θ as found through the training process of the model, are not necessarily the best parameters. Choromanska et al. (2015) address this issue and assert that this is not a major problem as the found local minima are usually of high quality and finding the real global minimum of the training data would be over-fitting.

(i.e. 4 concatenated quarters of financial statement variables) with 100, 50, and 33 hidden units, respectively. We use an Exponential Linear Unit (ELU) (Clevert et al. 2015) as the activation function in each layer. The ELU used as the activation function f in equation 2 is defined as:

$$\text{ELU}(x) = \max(0, x) + \min(0, \alpha * (\exp(x) - 1)), \tag{3}$$

where $\alpha = 1$, and x is the input.

In contrast to the popular rectified linear unit (ReLU) (Nair and Hinton 2010), the ELU function can have negative values and therefore is able learn on examples for which the activation is zero. The data for the DNN is passed with a batch size of 256 and a total training period of 10 epochs per training set. The entire implementation is done in Python using PyTorch (Paszke et al. 2017).

3.2.2 Recurrent Neural Network - Gated Recurrent Unit

The distinct characteristic of a Recurrent Neural Networks (RNNs) compared to a traditional feed-forward DNN is that it is designed to process sequential data $x_1, x_2, ..., x_t$ by sharing a tensor, called *hidden state* h, between all sequence steps x_t. Such a standard RNN model can be described as:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}X_t + b_h) \tag{4}$$

$$y_t = W_{hy}h_t + b_y \tag{5}$$

where t denotes the sequence step, X_t the input at step t , h_t denotes the hidden state at step t, tanh the tanh non-linearity, with $\tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, and W denotes the weight matrices that are randomly initialised. Concretely the weight matrix W_{hh} is used to transform the past hidden state h_{t-1} to h_t, W_{xh} is used when transforming the input X_t at step t to h_t, and

W_{hy} is used when transforming the computed hidden state h_t to the output y_t. b represents randomly initialised column matrices added as biases to the calculation of h_t (b_h) and y_t (b_y).[12]

We employ the RNN architecture because the financial statement data used as inputs in the models follows a temporal sequence. With other machine learning models the variables of all inputs are concatenated to form a vector of 484 elements (i.e., 121 variables * 4 sequence steps). In the case of the RNN, this concatenation is not necessary as the model is designed to take a sequence of four quarters (with 121 variables) as an input so that one quarter represents to one sequence step. In comparison to many published applications of RNNs where the type of neural network is applied to settings with inputs of a relatively long sequence length, the input sequence in this application is just 4 steps (i.e., 4 quarters) long.

Multiple types of RNNs exist in the literature and the particular one chosen for this task is a Gated Recurrent Unit (GRU) which is functionally very similar to the popular Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997). These architectures were developed to deal with the exploding and vanishing gradient problem in traditional RNNs by introducing memory cells and forget gates. The benefit of the GRU over the LSTM is that it has a smaller number of parameters and gates to be learned by combining the forgetting gate and the decision to update the unit state into a single update unit. Prior research suggests that GRUs are similar to LSTMs in applied settings (Chung et al. 2014).

The mathematical properties of a GRU are described as the following set of equations:

$$h_t = (1 - z_t)n_t + z_t h_{(t-1)} \tag{6}$$

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \tag{7}$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}) \tag{8}$$

[12]These type of neural network models have become state of the art in a range of natural language processing (NLP) tasks (e.g., Mikolov 2012).

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t(W_{hn}h_{(t-1)} + b_{hn})) \tag{9}$$

with h_t being the hidden state at time t, x_t the input at time t, $h_{(t-1)}$ the hidden state of the layer at time $t-1$ or the initial hidden state at time 0, and z_t, r_t, n_t being the update, reset, and new gates. σ represents the sigmoid function $\sigma(x) = \frac{e^x}{e^x+1}$, W and b are learned weight matrices and biases belonging to various gates as denoted in the second letter of the subscript. The weights and biases where the first letter of the subscript is h (e.g. W_{hr}, b_{hr}) transform the hidden state h in a gate, whereas a subscript starting with the letter i (e.g. W_{ir}, b_{ir}) transform the input x.

The GRU initially computes the hidden state h_t according to the current input vector x_t, and then uses this information to compute the update gate z_t and reset gate r_t. Then, it uses current reset gate r_t, input x_t and previous hidden state $h_{(t-1)}$ to calculate new memory content n_t. The previous hidden state $h_{(t-1)}$ and new memory content n_t are then combined to form the final hidden state h_t. At the first iteration h and n are initialised as zeros.

The employed GRU architecture follows the custom of stacking multiple GRU cells (i.e., 10 GRUs in this setting) on top of each other. Stacking means that one GRU takes the output of a GRU below as input until the GRU on top of the stack computes the final output. Figure 3 demonstrates how this stacking computes the hidden state and output. The blue cells in the figure represent w GRU cells that are stacked for n sequence steps. It demonstrates how the input sequence x and the hidden state h is transformed through the network to form the final hidden state $h_n^{(w)}$ and cell state $c_n^{(w)}$ (the hidden state of the bottom GRU would be $h_n^{(0)}$).

Each GRU in the setup has a hidden state dimension of 20 units, and the hidden state of the top most GRU $h_n^{(w)}$ is linked to a fully connected linear unit for prediction. We use the RMSProp optimizer to train the model with a learning rate of 0.001 training for 5 epochs with a batch-size of 128 elements.

3.2.3 Random Regression Forest

A random forest is a supervised ensemble learning approach that combines multiple classification and regression trees (CART) for a non-linear prediction. It has been introduced by Breiman (2001) and we use the scikit-learn implementation that is based on Pedregosa et al. (2011).

A CART tree is a hierarchical structure with every "node" representing a binary split of the data space into pieces based on the value of a variable. During the construction of the tree starting from the root node, a split is chosen among all possible splits so that the resulting node becomes the "purest". In the case of regression trees this impurity measure refers to the mean squared error (MSE) so the split with the lowest possible MSE is chosen.[13]

The two hyperparameters for the random forest are the number of regression trees that it consists of and their maximum depth. The use of many trees makes the forest more robust as the dimensionality of the inputs increases as every regression tree gets assigned a random set of inputs. A higher number of trees therefore means that variables get reused often in permutations with other variables. The computational intensity of training the model scales linearly with the number of trees it consists of. We decided to use 200 trees per forest.[14]

One benefit of using a random forest is that it provides a variable importance measure. This measure allows us to identify important variables based on which the predictions are made. Neural net based machine learning models lack such a measure making the output less interpretable. The mathematical construction of this variable importance measure is explained in Breiman et al. (1984).

[13] The algorithm for the construction of trees is more complex in detail than outlined here. (Breiman et al. 1984) and more recently (Loh 2011) explain further details of the CART algorithm.

[14] The maximum depth induces a trade-off between overfitting and modelling capacity. A deeper tree can model a more complex and intricate combination of inputs with a hypothetically unlimited depth of a tree perfectly learning the entire training data to the point where one branch just contains one observation. Since such over-fitting might be problematic for the prediction of unseen observations, and as the memory requirements of the trees grows as well, we limit the depth to 10 splits.

3.2.4 Linear Models

Linear regression still represents the standard forecasting methods in econometrics and financial research. To investigate the benefits of utilising non-linear learning algorithms, we benchmark the machine learning models against a linear *ordinary least-squares (OLS)* regression and a regularising *least absolute shrinkage and selection operator (Lasso)* regression (Tibshirani 1996).

3.3 Ordinary Least Squares (OLS)

The most basic linear regression model is estimated via ordinary least squares (OLS). It assumes that the target variable y can be approximated through a linear combination of the independent variables x by a set of coefficients β. It can be written in matrix notation as:

$$y = X\beta + \varepsilon \tag{10}$$

where y and ε are vectors of length n of the dependent variables and error terms, and X is a matrix of dimension $n \times d$ that contains the explanatory variables.

The best set of estimates $\hat{\beta}$ for β is found by solving the minimisation problem

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\, g(\beta) \tag{11}$$

for $g(\beta) = \left\|y - X\beta\right\|^2$.

3.4 Least absolute shrinkage and selection operator (Lasso)

The least absolute shrinkage and selection operator (Lasso) (Tibshirani 1996) is a linear regression and was introduced to improve the model fitting by selecting only a subset of coefficients in the final prediction model to avoid overfitting. Lasso aims to solve the quadratic

minimisation problem

$$\min_{\beta_0,\beta} \left\{ \frac{1}{N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 \right\} \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t \tag{12}$$

with the tuning parameter $t \geq 0$.

Rewriting this optimisation using the matrix notation in the Lagrangian form gives

$$min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \alpha \|\beta\|_1 \right\} \tag{13}$$

where a relationship exists between the respective tuning parameters t and α.

In a simple interpretation, the parameter α and t control the number of selected variables in the model. If $\alpha = 0$ the Lasso regression finds the same coefficients as the OLS regression, and as α becomes larger fewer independent variables are selected as more coefficients become zero.

3.5 Evaluation metrics

3.5.1 Training loss function

All models are trained using the mean squared error (MSE) as the loss function. For the prediction $\hat{y}_i$ and the ground truth label y_i of observation i the MSE is defined as:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{14}$$

Raising the error $e = y_i - \hat{y}_i$ to the second power not only has the effect of making the term positive, but tunes the model to give more weight towards learning from large errors.

3.5.2 Predictive performance

The predictive ability of the models is evaluated using the errors e that are calculated for all predictions $\hat{y}_i$ and observed market reactions y_i in the (out-of-sample) test set as $e_i = y_i - \hat{y}_i$.

One way of comparing the predictive ability of the models is to compare the value of the loss function and the mean squared error (MSE) for the test set predictions.[15]

While the mean squared error is a meaningful metric to compare the predictive performance across models for a regression task, it lacks the economic interpretability of accuracy measures used classification tasks such as the percentage of correctly predicted outcomes. For this reason, we develop another metric, Percentage Correct (PC), which quantifies in how many cases the respective models predict the correct sign of market reactions of various thresholds. The models are still trained using the mean squared error as the loss function during training but are subsequently evaluated also on the PC metric in addition to the mean squared error.[16]

For small BHARs close to zero the meaningfulness of the sign is ambiguous. A reaction of -1% might be driven by a very similar information content as a reaction of $+1\%$ so that the sign might be the result of noise in the price movement. Therefore a set of threshold values (ε) are created within which the absolute values of the predictions and ground truths are set to zero. That is, if the absolute value of the reaction r is smaller than a given ε so that $|r| < \varepsilon$ then, in the context of evaluating the prediction, it is assigned the value zero $r = 0$.

$$r = \begin{cases} 0 & \text{if } |r| < \varepsilon \\ r & \text{otherwise} \end{cases} \tag{15}$$

This adjustment separates the BHARs into significantly negative $r < -\varepsilon$, ambiguous $-\varepsilon < r < \varepsilon$, and significantly positive reactions $r > \varepsilon$.

[15] We also compute the root mean squared error (RMSE), the mean absolute error (MAE), and the median absolute error (MedAE) for easier interpretation (See Appendix A).

[16] Using the PC metric for evaluation is a compromise between evaluating the different models purely based on a directional prediction performance and using the information contained in the return magnitudes. It is also possible to train the models on the PC metric as the loss, which likely should improve the PC test results. However, training using the MSE has computational advantages. Thus, our findings based on the PC metric are likely conservative.

Seven epsilon thresholds are created $\varepsilon \in \{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ and applied to both the prediction of models $\hat{y}$ and the actual ground truth y. This is done after the prediction so that if $\hat{y}$ and/or the ground truth y, falls into the respective epsilon threshold (e.g. $|\hat{y}| < \varepsilon$, $|y| < \varepsilon$) we manually set it to 0. This way the choice of ε, does not impact training as the epsilon threshold check occurs after $\hat{y}$ is predicted.[17]

Table 3 summarises how many BHAR observations fall inside and outside the particular ε thresholds. For example, at $\varepsilon = 0$ all observations fall outside the threshold and thus all available market reactions are used to evaluate the prediciton performance, while at $\varepsilon = 0.5$ only 5% of the observations fall outside of the threshold.

The introduction of ε thresholds complicates the calculation of correct predictions. As Table 3 demonstrates, a growing ε leads to an increasing amount of observations to fall within the ε threshold. This leads to a metric that measures the correctly predicted BHARs withing or outside the ε thresholds that is not comparable across the different levels of ε. We therefore define the PC such that (a) it quantifies the rate of correctly predicted directions of stock market reactions outside of the dedicated ε threshold (e.g., predicting a BHAR of 20% at a $\varepsilon = 0.1$ where the true BHAR has a positive sign would be a *correct* prediction and (b) the metric is penalised for the cases where a true BHAR is outside of the ε threshold and predicted to be within.[18]

[17] Rectifying the ground truth value to zero based on ε before the training of the models might be preferable as models would only be explicitly exposed to significant reactions (outside of ε) and minor reactions (inside of ε) as training inputs. Models would not need to learn to distinguish ambiguous minor reactions as these would be set to zero. This would however drastically increase the amount of necessary computation as instead of merely fitting one model, it would be necessary to train one model per ε-threshold.

[18] See Appendix B for a formal definition of the PC metric for predictions across epsilon thresholds. It is important to note that the PC metric is merely used as an evaluation metric for the test loss, and not during the training of the models.

4 Empirical Results

4.1 Prediction Performance

Table 4 presents the results of the evaluation metrics calculated over the predictions of the entire test samples across all periods. The results suggest that the Random Forest is the best performing model with a MSE of 0.055 and a mean absolute error of 0.153, closely followed by the Lasso regression ($MSE = 0.055$ and $MAE = 0.154$), and the deep neural net ($MSE = 0.058$ and $MAE = 0.16$). The ordinary least squares regression and the recurrent neural net are the poorest performing models. Overall, the differences between the linear and non-linear methods are not large. Based on these evaluation metrics non-linear machine learning models do not seem to outperform traditional linear models significantly in predicting the magnitude of stock returns around earnings announcements.

To evaluate the prediction performance of the different methods in more detail over various epsilon thresholds Table 5 lists the mean and standard deviation of the PC measures as well as the mean proportion of predictions outside the epsilon threshold. The Random Forest has the highest percentage of correct predictions averaging 55% of the cases correctly predicted over the entire return distribution (with an epsilon $\varepsilon = 0$), and an average of 56% correctly predicted at the other end of the spectrum with $\varepsilon = 0.5$. The model performs best with an average of 59% correctly predicted at $\varepsilon = 0.3$, suggesting that market reactions of medium severity can be most confidently predicted. The standard deviation of these predictions increases monotonically with increasing epsilon threshold.

Interestingly, the Lasso regression performs equally well on more moderate absolute market reactions up to 10% but experiences an significant decrease in the average of correctly predicted cases and significant increase in the standard deviation of the predictions at higher thresholds. This finding also applies to the neural net based models (i.e. DNN, RNN) and OLS. For small values of $\varepsilon \in \{0, 0.05, 0.1\}$ the performance of the linear models is on par and, in the case of Lasso, even superior to any neural network. While the RNN is about

1% better at the mean PC than the DNN, Lasso is about an additional 1% better than the RNN. However, at higher levels of $\varepsilon \in \{0.2, 0.3, 0.4, 0.5\}$ the neural net based models (as well as the Random Forests) outperform the linear models clearly. At $\varepsilon = 0.5$ for instance, Lasso and OLS are barely more accurate than a random guess with a mean PC of 51% and 52%. At these levels, the DNN and RNN are performing substantially better with a mean PC of 54 and 55%. These results suggest that the neural net based models are better at predicting larger reactions than linear models. This divergence in performance between non-linear and linear models with increasing epsilon threshold is consistent with findings in the prior literature of a S-shape relationship of the earnings response coefficient (Freeman and Tse 1992; Kinney et al. 2002) and point to non-linear relationships between the accounting predictors and stock returns at larger market reaction thresholds.

The finding of the monotonically increasing variance of the quarterly PC measure for all models further suggests that market reactions at larger ε levels are harder to predict.[19] However, across all thresholds of ε, Lasso and OLS exhibit higher standard deviations than neural net based models. The difference is most significant at higher values of ε, but is also exhibited at the lower thresholds where the linear models have a better mean PC. The mean proportion of samples based on which the PC metric has been computed might be another contributing factor for this phenomenon, however. Quarters with just a few predictions outside of a high ε threshold, would inadvertently yield a higher standard deviation than lower levels of ε where the majority of the observations are used to compute a relatively *stable* level of PC. For most models only about 5% of predictions fall into the $\varepsilon = 0.5$ case.

4.2 Trading Strategy

We next test the profitability of an investment strategy that takes long and short positions based on the predictions of the different models that abnormal returns fall on the positive or negative side outside the epsilon band. That is, the strategy ignores predicted magnitudes

[19] It is however also possible that at larger market reactions a few outliers affect the PC metric

that fall within the respective epsilon band. The positions are taken one day before the earnings announcement and are held until 30 days after. More formally, the back-tested trading strategy assumes that for every quarter Q there exist a set of indices I_Q of a test set that contains observations $i \in I_Q$ which refer to pairs of predictions $\hat{y}_i$ and ground truth y_i. The trading positions are constructed based on the value of the prediction $\hat{y}$ as:

$$T_{Q,\varepsilon} = \{i \in I_Q | \hat{y}_i > \varepsilon \vee \hat{y}_i < -\varepsilon\} \tag{16}$$

Every trading position $t \in T_{Q,\varepsilon}$ is weighed equally using a nominal quarterly portfolio size s.

$$w_t = \frac{s}{|T_{Q,\varepsilon}|} \tag{17}$$

with w_t denoting the weight of the position t. Based on the sign of $\hat{y}_t$ either a long position when $\hat{y}_t > 0$ or a short position when $\hat{y}_t < 0$ is taken. Depending on the sign of the observed market reaction y_t these positions yield a profit or loss of p_t depending on whether the correct sign was predicted.

$$p_t = \begin{cases} 1 + y_t & \text{if } \hat{y}_t > 0 \\ 1 - y_t & \text{if } \hat{y}_t < 0 \end{cases} \tag{18}$$

This profit or loss p_t of the individual position is then multiplied by the weight w_t and summed across position in that quarter to determine the overall quarterly profit $P_{Q,\varepsilon}$

$$P_{Q,\varepsilon} = \sum_{t \in T_{Q,\varepsilon}} w_t p_t \tag{19}$$

Because of the use of BHARs as the dependent variable y, the quarterly profit $P_{Q,\varepsilon}$ can be directly interpreted as the abnormal return of a given quarter and ε threshold.

We backtest the performance of this trading strategy over the entire sample period by

compounding the returns starting at the portfolio size of 1 in Q1 1991, and reinvesting the proceeds in subsequent quarters. In order to impose somewhat more realistic diversification and trade size restrictions the trading strategy requires at least three position per quarter. Furthermore, extreme BHAR reactions that are greater than +100% or smaller than −100% are censored to these limits.

The results are presented in Table 6. The table summarises the mean and standard deviation of the the quarterly abnormal returns per model by ε as well as the Information Ratio (before trading costs). The figures in italics show the *total number of trades* executed for the strategy over the entire backtesting period Q1 1991 to Q4 2017.

Consistent with results of the PC metric the best performing strategy overall is based on the Random Forest exhibiting the highest mean excess return across the various levels of ε (except for $\varepsilon = 0.05$, in which case the LASSO model dominates). In case of $\varepsilon = 0$, equivalent to a model where every position is traded, the mean quarterly excess return is 2.1%. The highest mean return of 13.2% is found at $\varepsilon = 0.5$. The monotonically increasing abnormal returns across the ε levels suggest that the model is able to distinguish between market reactions of higher magnitude and smaller market reactions. The high number of trades at lower levels of ε suggests that the strategy is impracticable and unprofitable at these lower thresholds in the presence of transaction costs. The model has the highest Information Ratio (IR) of 1.13 at an epsilon of 0.1.

Comparing the DNN and the Lasso models, the Lasso model performs favourably at lower thresholds of $\varepsilon \in \{0, 0.05, 0.1, 0.2\}$ while the DNN outperforms at higher thresholds again suggesting that the DNN is better at predicting large reactions than the linear Lasso model. This difference between neural networks and linear methods can also be observed in the comparison of the RNN and the OLS models: at lower ε thresholds, the OLS model is superior in terms of excess returns and IR, while at higher thresholds the RNN is.

The poor performance of the linear models at the higher ε thresholds is consistent with the poor performance in the PC metric and confirms the hypothesis that linear models are

inferior at predicting market reactions of large magnitudes. The relatively lower number of trades (at a higher profitability) in comparison to the neural net models at $\varepsilon \in \{0.05, 0.1, 0.2\}$ indicate that the linear models are better at distinguishing the cases with ambiguous reactions close to zero. While the phenomenon deserves a more thorough investigation, this evidence suggests that minor reactions follow linear statistical relationships, while the major reactions are more likely the result of complex non-linear dynamics of financial statement components.

Table 7 presents the terminal wealth in dollars $P_{Q,\varepsilon}$ at the end of the backtesting period (Q4 2017) if a nominal portfolio of size \$1 was invested at the beginning of the test period (i.e. Q1 1991). The random forest generates the highest terminal wealth of $\$71,527$ in the case of $\varepsilon = 0.5$. The DNN outperforms the RNN among all ε thresholds except for the highest $\varepsilon = 0.5$ case. The LASSO model performs stronger than the plain OLS model with the highest terminal wealth in the $\varepsilon = 0.1$ scenario.

While the findings presented in Table 3 suggested no major differences in the predictive ability between the RF and LASSO at lower epsilon levels, and an equivalent performance of OLS to the neural nets, the trading results in this section indicate that it makes a difference whether the machine learning methods are evaluated based on trading profits, or based on the common precision metrics that traditionally have been used in the machine learning literature.

Figure 4 charts the compounded excess returns over the study period with an ε of 0. The figure suggests that the non-linear machine learning models were less sensitive to economic downturns than the linear models. During the global financial crisis from Q3 2008 to Q2 2009 as well as the Dot.com crash the neural networks are relatively stable, while OLS and Lasso suffer significant losses. Interestingly, the Random Forest returns remain relatively stable during the Global Financial Crisis of 08-09. This ability of the neural networks to have steady (albeit lower) returns in this setup is consistent with the relatively lower mean standard deviation of quarterly returns reported in table 6. This might be due to the explicit

memory cells the GRU architecture allows, and warrants further investigation into why the model was better able to anticipate and navigate these crisis periods.

In untabulated results we further find that the quarterly returns diminish over time across all models and particularly for larger epsilon values. This return pattern is consistent with the notion of financial markets having become more efficient over time and accounting-based regularities to have been increasingly arbitraged away (McLean and Pontiff 2016).

4.3 Economic drivers of prediction

In this section we examine which accounting variables drive the Random Forest predictions. Random forests allow quantifying the relative importance of an input variable for the prediction by accumulating the improvement in the splitting criterion (i.e., the sum of squares). Computed for every tree that constitutes the random forest, the variable importance for the whole forest is the mean of these metrics in the same way the prediction of the forest is the mean of the tree predictions. Being normalised to sum to one over all features that constitute the input vector, the metric helps to assess how much a particular feature improved the prediction ability in comparison to the others. Variable importance can be computed for every individual quarter as well as aggregated over the entire sample period. To rank the most important accounting variables the variable importance measures are added up per variable over all quarters of the study period. A higher number, therefore, indicates a higher importance.

Table 8, Panel A shows the results of the analysis of the most important variables overall over the entire sample period (aggregated over 4 quarters, i.e., out of 121 variables). To aggregate we compute the sum of variable importance measures per financial statement variable over the sample period. Panel B shows the results by quarter and presents the top 10 variables of the 484 quarterly variables (121 for 4 quarters). The findings summarised in the table provide several insights.

First, Panel A suggests that the model selects accounting components that are required

to estimate free cash flows, i.e., earnings, components of changes in net working capital and CAPEX. That is, out of 121 potential candidates the model assigns the highest importance to known drivers of fundamental value to minimise its prediction error. Second, the model's selection of variables (overall and by quarter) is consistent with commonly known accounting-based cross-sectional return predictors. Specifically, we find earnings before extraordinary items, accruals, asset growth, CAPEX, cash flows and cash dividends to be the most important variables the model selects for the prediction. These firm characteristics have in the accounting-based anomalies literature been found to be associated with stock returns (Foster et al. 1984; Bernard and Thomas 1989; Lakonishok et al. 1994; Sloan 1996; Abarbanell and Bushee 1998; Richardson et al. 2005; Cooper et al. 2008; Hartzmark and Solomon 2013).

Third, Panel B reveals that the quarters in which these variables are important are one and four quarters before the announcement. Consistent with commonly practiced year-on-year and previous quarter comparisons by market participants and financial analysts, the model assigns higher importance to the variables in the previous one-quarter and previous one-year financial statements when making predictions for the current quarter. Thus, the model's selection of variables is also consistent with findings from a long-standing literature in accounting on the time-series properties of cash and accrual components of earnings and investors' extrapolation of these components (Bernard and Thomas 1990; Sloan 1996; Abarbanell and Bushee 1998).

Overall, the results in this section indicate that the models did not overfit to a special case with idiosyncratic financial statement items, but rather that they select variables that are useful for forecasting future free cash flows and that correspond to known predictors of earnings and stock returns.

4.4 The effect of firm size

The large number of trades executed by some of the strategies at lower ε values indicates that trading costs could thwart profitability. Another concern is whether the exceptionally large returns at higher ε values stem from trades in smaller companies. In robustness test we therefore investigate the role of firm size. For this we bin companies into size groups along their market capitalisation. The market capitalisation of a firm is calculated by multiplying the share price (`prccq`) with the total amount of shares outstanding (`cshoq`) at the beginning of the quarter. We call these bins micro caps, small caps, medium caps, and large caps.

Table 9 depicts the selected thresholds for the bins. For 1.3% of observations the market capitalisation could not be calculated because of missing values.

We examine the Random Forest for this analysis at various ε thresholds. To conduct the analysis we construct portfolios that include all firms and differ from each other by leaving out a particular market cap bin. Through this 'leave one out' approach, the return of the *standard portfolio*, which includes all firms, can be compared to the returns of portfolios with omitted bins. The relative difference in the overall returns between these portfolios indicates the contribution that the bin made to the trading strategy that trades all companies. For each of these 'leave one out' portfolios the total compounded return relative to the standard portfolio is calculated using the previously defined trading strategy in section 4.

The results are presented in Table 9. The results indicate that for the high ε strategies, which have achieved the largest abnormal returns, firms of the smallest market capitalisation are very important. The total return of the $\varepsilon = 0.5$ scenario is reduced by 97% if micro cap firms were excluded. Small cap firms have a similarly negative, albeit less dramatic, effect on the profitability of the random forest strategy. However, the effect of micro and small cap firms on the profitability of the strategy declines monotonically with smaller epsilon. This suggests that smaller firms are more important for predictions of large market reactions. The results further suggest that the investment strategy at epsilon of 0.1 still generates large returns even when excluding micro and small cap stocks. The latter findings are consistent

with the increasing volatility of the abnormal returns at increasing epsilon values and the highest IR value at epsilon of 0.1 reported in table 6.

In contrast, the profits were 62% larger when mid caps were excluded, indicating that they diminish profits. However the large caps contribute positively as the total profits were 57% smaller when they were excluded. The analysis in this section confirms that firms of smaller market capitalisation play an important role in the abnormal profits achieved by the learning-based trading strategy.

5 Conclusion

This study explores the use of machine learning algorithms for financial statement analysis. We investigate whether machine learning methods are capable of forecasting the sign and magnitude of stock returns around earnings announcements based on financial statement data alone. We test and compare various models from the machine learning repertoire including simple OLS, LASSO, Random Regression Forests, Deep Neural Networks and Recurrent Neural Networks over a period from Q1 1991 to Q4 2018.

Despite relatively large average forecasting errors the non-linear methods are able to predict the direction and various thresholds of the absolute magnitude of the market reaction to earnings announcements correctly in 53% to 59% of the cases on average. Among the various methods Random Forests provide the best performing models. However, we also find that the (non-linear) neural net models perform relatively stronger when predicting extreme market reactions, while the linear models like OLS and LASSO perform comparably better when predicting moderate returns consistent with a non-linear earnings response relationship.

We further provide evidence on the investment performance of signals based on the predictions of the various models. Consistent with the results on their prediction accuracy, we find Random Forest-based strategies to outperform other learning-based strategies. However, we find some variation in the investment performance depending on different thresholds

of the absolute magnitude of the abnormal returns around the announcements. The Random Forests attain the highest information ratio of 1.13 for predictions of abnormal returns of absolute magnitude of 10% of higher, generating abnormal returns of 10.2% per quarter. We find largest abnormal returns of 13.2% per quarter for predictions of the most extreme reactions above absolute abnormal returns of 50%, but also find that these tend to be concentrated among small-cap stocks. Furthermore, despite generating the lowest investment returns over the sample period, the recurrent neural network produces surprisingly stable returns with low volatility and exhibits lower drawdowns during the market downturns of 2002-03 and 2008-09.

We then analyse the underlying economic drivers of the performance of the random forests. We find that the models tend to select as the most important predictors those accounting variables that are components of free cash flows and known predictors of stock returns. Specifically, we find that the models base their predictions on the time-series and cross-sectional properties of earnings and accruals, as well as asset growth, CAPEX, cash flows and cash dividends - all firm characteristics that have been found in the accounting-based anomalies literature to predict stock returns. Furthermore, consistent with commonly practiced year-on-year and previous quarter comparisons by market participants and financial analysts, the models rely predominantly on previous quarter and previous year reported values of these variables when making predictions for the current quarter.

While this study does not test or develop a new economic theory of stock returns around earnings announcements, it adds to our understanding of the role of fundamental analysis and the behavior of stock returns around earnings announcements by providing first evidence on the usefulness of modern machine learning methods for financial statement analysis. We document that "machine-learned" relationships between the set of financial statement variables and stock returns follow fundamental valuation intuition for predicting free cash flows and are consistent with established return relationships of accounting-based regularities found in prior accounting research. We further show that the earnings-return

relationship is likely nonlinear particularly for extreme events. We hope that our findings will encourage further research into whether and how machine learning methods can further our understanding of how financial reporting outputs relate to firm value and their relationships to accounting-based anomalies.

Appendix A - Alternative Evaluation Metrics

In addition to the MSE we also compute the root mean squared error (RMSE), the mean absolute error (MAE), and the median absolute error (MedAE) for easier interpretation as follows:

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{20}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{21}$$

$$\text{MedAE} = \text{median}(|y_i - \hat{y}_i|_{i \in I}) \tag{22}$$

These metrics differ slightly in their interpretation. The MSE is generally used during the training of a model because the squaring operation disproportionally penalises large errors in comparison to small ones. This way the models are tuned to be more sensitive to learn from large errors than from minor errors. However, because of this squaring operation the MSE is not clearly interpretable with regards to the original unit of measure. Therefore the RMSE is also computed which undoes the squaring operation by taking the square root.

The MAE on the other hand, directly computes the mean absolute error of the predictions. This makes the MAE the simplest metric to interpret as it denotes how much the predictions deviate from the ground truths on average. As the mean could be skewed by a few extremely bad predictions, the MedAE is also included in the array of metrics as it that represents the median of the mean absolute errors.

Preliminary test statistics indicate some extreme values and outliers among the OLS model that might potentially bias the performance statistics. Therefore the evaluation met-

rics are computed based on the 99.99^{th} percentile of the absolute value of errors per model type. The percentile is constructed using the ordered set E that is composed of the absolute values of the errors $E = \{|e_0|, .., |e_n|\}$ of the n predictions contained in the complete test set I per model type. The set E is sorted (from smallest to largest) so that for every monotonically rising integer index j of the elements $k \in E$ it is true that $k_j \leq k_{j+1}$. Then the ordinal rank r for percentile p is computed as:

$$r = \left\lceil \frac{p}{100} \times |E| \right\rceil \tag{23}$$

So that taking the value k_r from E yields the number that is used to limit the selection of predictions and ground truth labels used in the computation of the subsequent statistics by redefining the set of test sample indices I of every model to samples where

$$|y_i - \hat{y}_i| \leq k_r \tag{24}$$

This operation excludes 0.0001% of observations with the most extreme outlier errors which are not representative of the general predictive ability of the models.

Appendix B - PC Metric for Predictions Outside Epsilon Thresholds

With $i \in I_t$ being the index of the ground truth value y_i and predicted value $\hat{y}_i$ in the set of test sample indices I_t of a quarter t, $\varepsilon \in \{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$ we define the helper functions $sign(x)$, $g(x, \varepsilon)$, $h(x, z, \varepsilon)$

$$sign(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{25}$$

$$g(x, \varepsilon) = \begin{cases} 1 & \text{if } x < \varepsilon \wedge x > -\varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

$$h(x, z, \varepsilon) = g(x, \varepsilon) * g(z, \varepsilon) \tag{27}$$

so that the PC metric can be computed as

$$PC = \frac{\sum_{i \in I_t} (|h(y_i, \hat{y}_i, \varepsilon) - 1| \, \frac{|sign(y_i) + sign(\hat{y}_i)|}{2})}{|I_t| - \sum_{i \in I_t} h(y_i, \hat{y}_i, \varepsilon)} \tag{28}$$

Based on this definition, a perfect PC metric of 1 indicates a case where the models predicted significant (i.e. outside of ε) BHARs, the predictions have the correct sign, and there are no observed BHARs that were predicted to be not significant (i.e. inside of ε).

References

Abarbanell, J. S. and Bushee, B. J. (1997). Fundamental analysis, future earnings, and stock prices, *Journal of Accounting Research* **35**(1): 1–24.

Abarbanell, J. S. and Bushee, B. J. (1998). Abnormal returns to a fundamental analysis strategy, *The Accounting Review* pp. 19–45.

Azur, M. J., Stuart, E. A., Frangakis, C. and Leaf, P. J. (2011). Multiple imputation by chained equations: what is it and how does it work?, *International Journal of Methods in Psychiatric Research* **20**(1): 40–49.

Ball, R. and Brown, P. (1968). An empirical evaluation of accounting income numbers, *Journal of Accounting Research* pp. 159–178.

Ball, R. and Kothari, S. P. (1991). Security returns around earnings announcements, *The Accounting Review* pp. 718–738.

Banz, R. W. (1981). The relationship between return and market value of common stocks, *Journal of Financial Economics* **9**(1): 3–18.

Barboza, F., Kimura, H. and Altman, E. (2017). Machine learning models and bankruptcy prediction, *Expert Systems with Applications* **83**: 405–417.

Bartov, E., Givoly, D. and Hayn, C. (2002). The rewards to meeting or beating earnings expectations, *Journal of Accounting and Economics* **33**(2): 173–204.

Beaulieu-Jones, B. K. and Moore, J. H. (2017). Missing data imputation in the electronic health record using deeply learned autoencoders, *Pacific Symposium on Biocomputing 2017*, World Scientific, pp. 207–218.

Beneish, M. D., Lee, C. M. and Tarpley, R. L. (2001). Contextual fundamental analysis through the prediction of extreme returns, *Review of Accounting Studies* **6**(2-3): 165–189.

Beretta, L. and Santaniello, A. (2016). Nearest neighbor imputation algorithms: a critical evaluation, *BMC Medical Informatics and Decision Making* **16**(3): 74.

Bernard, V. L. and Thomas, J. K. (1989). Post-earnings-announcement drift: delayed price response or risk premium?, *Journal of Accounting Research* **27**: 1–36.

Bernard, V. L. and Thomas, J. K. (1990). Evidence that stock prices do not fully reflect the implications of current earnings for future earnings, *Journal of Accounting and Economics* **13**(4): 305–340.

Bernard, V., Thomas, J. and Wahlen, J. (1997). Accounting-based stock price anomalies: Separating market inefficiencies from risk, *Contemporary Accounting Research* **14**(2): 89–136.

Bird, R., Gerlach, R. and Hall, A. D. (2001). The prediction of earnings movements using accounting data: an update and extension of ou and penman, *Journal of Asset Management* **2**(2): 180–195.

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

Breiman, L., Friedman, J., Olshen, R. and Stone, C. (1984). *Classification and regression trees*, Taylor & Francis.

Chen, L., Pelger, M. and Zhu, J. (2019). Deep learning in asset pricing, *Available at SSRN 3350138* .

Choromanska, A., Henaff, M., Mathieu, M., Arous, G. B. and LeCun, Y. (2015). The loss surfaces of multilayer networks, *Artificial Intelligence and Statistics*, pp. 192–204.

Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. (2014). Empirical evaluation of gated

recurrent neural networks on sequence modeling, *arXiv preprint arXiv:1412.3555* .

Clevert, D.-A., Unterthiner, T. and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus), *arXiv preprint arXiv:1511.07289* .

Collins, D. W. and Kothari, S. (1989). An analysis of intertemporal and cross-sectional determinants of earnings response coefficients, *Journal of Accounting and Economics* **11**(2-3): 143–181.

Cooper, M. J., Gulen, H. and Schill, M. J. (2008). Asset growth and the cross-section of stock returns, *The Journal of Finance* **63**(4): 1609–1651.

Desai, H., Rajgopal, S. and Venkatachalam, M. (2004). Value-glamour and accruals mispricing: One anomaly or two?, *The Accounting Review* **79**(2): 355–385.

Donoho, D. L. (1995). De-noising by soft-thresholding, *IEEE Transactions on Information Theory* **41**(3): 613–627.

Engelberg, J., McLean, R. D. and Pontiff, J. (2018). Anomalies and news, *The Journal of Finance* **73**(5): 1971–2001.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work, *The Journal of Finance* **25**(2): 383–417.

Fama, E. F. and French, K. R. (1992). The cross-section of expected stock returns, *the Journal of Finance* **47**(2): 427–465.

Feltham, G. A. and Ohlson, J. A. (1995). Valuation and clean surplus accounting for operating and financial activities, *Contemporary Accounting Research* **11**(2): 689–731.

Foster, G., Olsen, C. and Shevlin, T. (1984). Earnings releases, anomalies, and the behavior of security returns, *Accounting Review* pp. 574–603.

Freeman, R. N. and Tse, S. Y. (1992). A nonlinear model of security price responses to unexpected earnings, *Journal of Accounting Research* **30**(2): 185–209.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep learning*, MIT press.

Greig, A. C. (1992). Fundamental analysis and subsequent stock returns, *Journal of Accounting and Economics* **15**(2-3): 413–442.

Gu, S., Kelly, B. T. and Xiu, D. (2019). Autoencoder asset pricing models, *Available at SSRN* .

Gu, S., Kelly, B. and Xiu, D. (2018). Empirical asset pricing via machine learning, *Technical report*, National Bureau of Economic Research.

Hartzmark, S. M. and Solomon, D. H. (2013). The dividend month premium, *Journal of Financial Economics* **109**(3): 640–660.

Harvey, C. R., Liu, Y. and Zhu, H. (2016). and the cross-section of expected returns, *The Review of Financial Studies* **29**(1): 5–68.

Hechenbichler, K. and Schliep, K. (2004). Weighted k-nearest-neighbor techniques and ordinal classification, *LMU Munich Working Paper* .

Hirshleifer, D., Hou, K. and Teoh, S. H. (2012). The accrual anomaly: risk or mispricing?, *Management Science* **58**(2): 320–335.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural Computation* **9**(8): 1735–1780.

Holthausen, R. W. and Larcker, D. F. (1992). The prediction of stock returns using financial statement information, *Journal of Accounting and Economics* **15**(2-3): 373–411.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift, *arXiv preprint arXiv:1502.03167* .

Kasznik, R. and McNichols, M. F. (2002). Does meeting earnings expectations matter? evidence from analyst forecast revisions and share prices, *Journal of Accounting Research* **40**(3): 727–759.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* .

Kinney, W., Burgstahler, D. and Martin, R. (2002). Earnings surprise materiality as measured by stock returns, *Journal of Accounting Research* **40**(5): 1297–1329.

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

Kothari, S., Lewellen, J. and Warner, J. B. (2006). Stock returns, aggregate earnings surprises, and behavioral finance, *Journal of Financial Economics* **79**(3): 537–568.

Lakonishok, J., Shleifer, A. and Vishny, R. W. (1994). Contrarian investment, extrapolation, and risk, *The Journal of Finance* **49**(5): 1541–1578.

Lev, B. and Thiagarajan, S. R. (1993). Fundamental information analysis, *Journal of Accounting Research* **31**(2): 190–215.

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 filingsa naïve bayesian machine learning approach, *Journal of Accounting Research* **48**(5): 1049–1102.

Loh, W.-Y. (2011). Classification and regression trees, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **1**(1): 14–23.

Mazumder, R., Hastie, T. and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices, *Journal of Machine Learning Research* **11**(Aug): 2287–2322.

McLean, R. D. and Pontiff, J. (2016). Does academic research destroy stock return predictability?, *The Journal of Finance* **71**(1): 5–32.

Mikolov, T. (2012). Statistical language models based on neural networks, *Presentation at Google, Mountain View, 2nd April* **80**.

Mohanram, P. S. (2005). Separating winners from losers among lowbook-to-market stocks using financial statement analysis, *Review of Accounting Studies* **10**(2-3): 133–170.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines, *Proceedings of the 27th International Conference n Machine Learning*, pp. 807–814.

Ohlson, J. A. (1995). Earnings, book values, and dividends in equity valuation, *Contemporary Accounting Research* **11**(2): 661–687.

Ou, J. A. and Penman, S. H. (1989). Financial statement analysis and the prediction of stock returns, *Journal of Accounting and Economics* **11**(4): 295–329.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L. and Lerer, A. (2017). Automatic differentiation in pytorch, *NIPS-W*.

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

Perols, J. (2011). Financial statement fraud detection: An analysis of statistical and machine

learning algorithms, *Auditing: A Journal of Practice & Theory* **30**(2): 19–50.

Piotroski, J. D. et al. (2000). Value investing: The use of historical financial statement information to separate winners from losers, *Journal of Accounting Research* **38**: 1–52.

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.

Richardson, S., Tuna, I. and Wysocki, P. (2010). Accounting anomalies and fundamental analysis: A review of recent research advances, *Journal of Accounting and Economics* **50**(2-3): 410–454.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain., *Psychological Review* **65**(6): 386.

Rumelhart, D. E., Hinton, G. E. and Williams, R. J. (1986). Learning representations by back-propagating errors, *Nature* **323**(6088): 533–536.

Savor, P. and Wilson, M. (2016). Earnings announcements and systematic risk, *The Journal of Finance* **71**(1): 83–138.

Skinner, D. J. and Sloan, R. G. (2002). Earnings surprises, growth expectations, and stock returns or don't let an earnings torpedo sink your portfolio, *Review of Accounting Studies* **7**(2-3): 289–312.

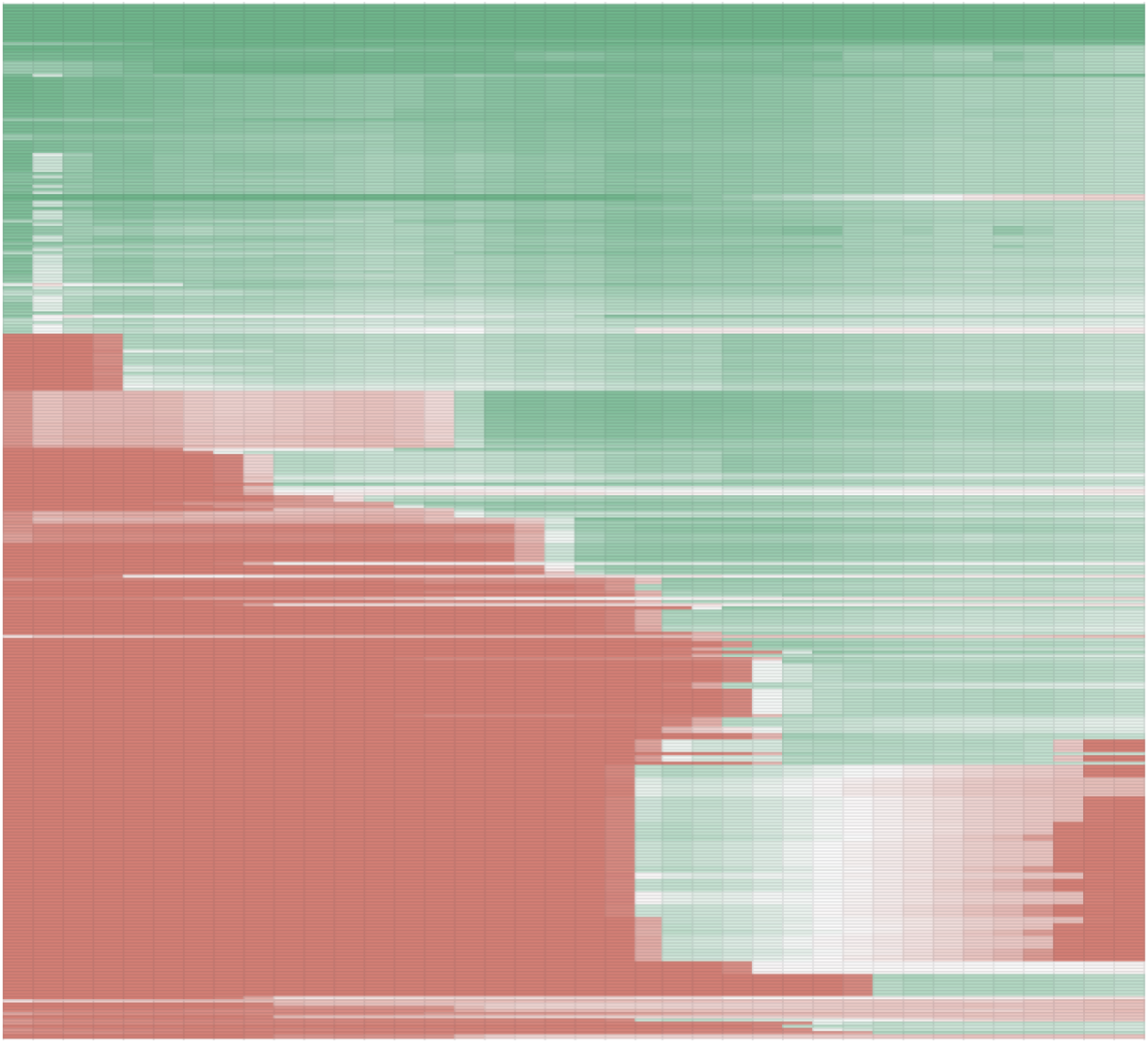
Sloan, R. G. (1996). Do stock prices fully reflect information in accruals and cash flows about future earnings?, *The Accounting Review* pp. 289–315.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society: Series B (Methodological)* **58**(1): 267–288.

Wahlen, J. M. and Wieland, M. M. (2011). Can financial statement analysis beat consensus analysts recommendations?, *Review of Accounting Studies* **16**(1): 89–115.

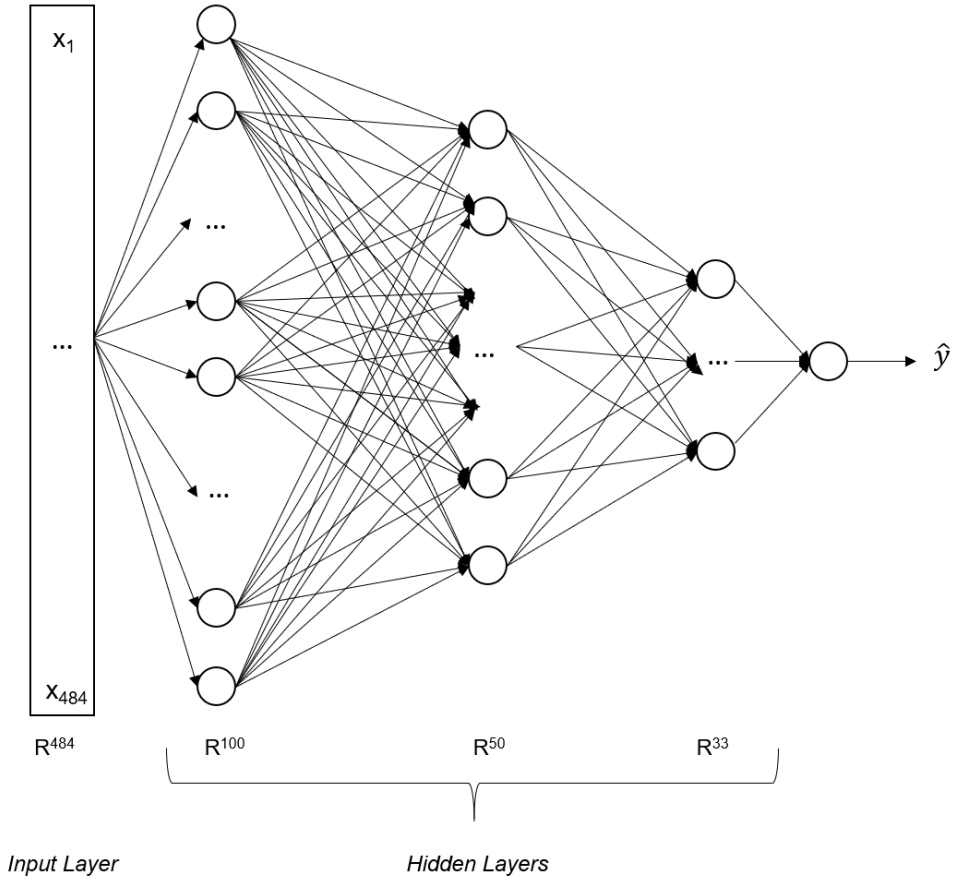
Yan, X. and Zheng, L. (2017). Fundamental analysis and the cross-section of stock returns: A data-mining approach, *The Review of Financial Studies* **30**(4): 1382–1423.

Figure 1: Visualisation of missing values in Compustat 1983-2018



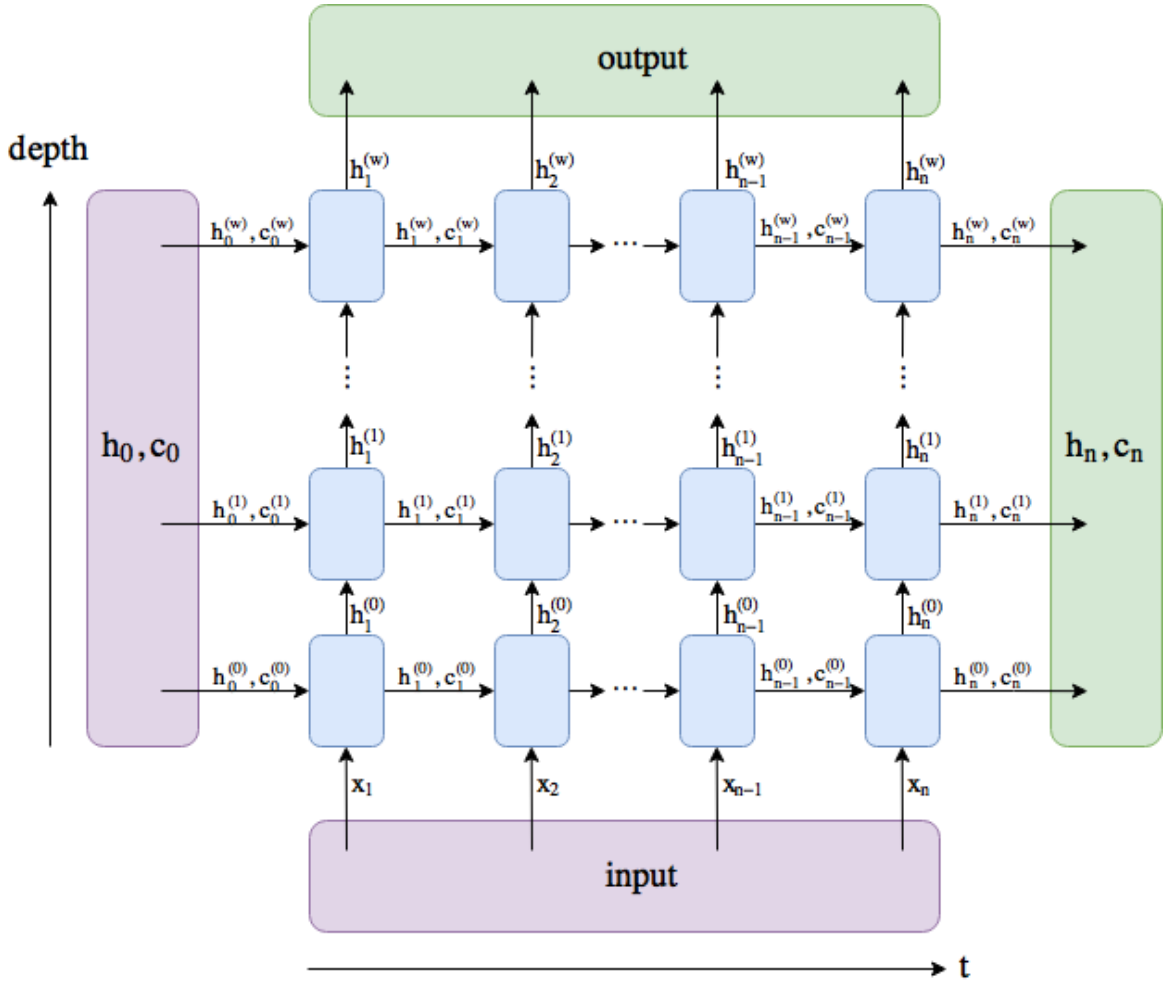
Note: This figure visualises the structure of missing values in the Compustat FUNDQ data set. The rows correspond to a particular variable (e.g. total assets) and the columns divide years from 1983 to 2018. The colour of the cells indicates the rate of missing values of a particular variable in a year. A dark red cell represents a rate of 100% missing, a green cell a rate of 0% missing, and a white cell 50% missing. The rows are sorted based on their overall rate of missing values. .

Figure 2: Visualisation of the neural network architecture



Note: This figure illustrates the architecture of the Deep Neural Network in this study. The circles denote a neuron where inputs are combined and run through a non-linear activation function with a neuron specific bias. The lines represents the weighted connections between nodes that turn the outputs of one layer to the inputs of a subsequent layer. For simplicity not all neurons and weight connections are depicted.

Figure 3: Recurrent Neural Network Architecture



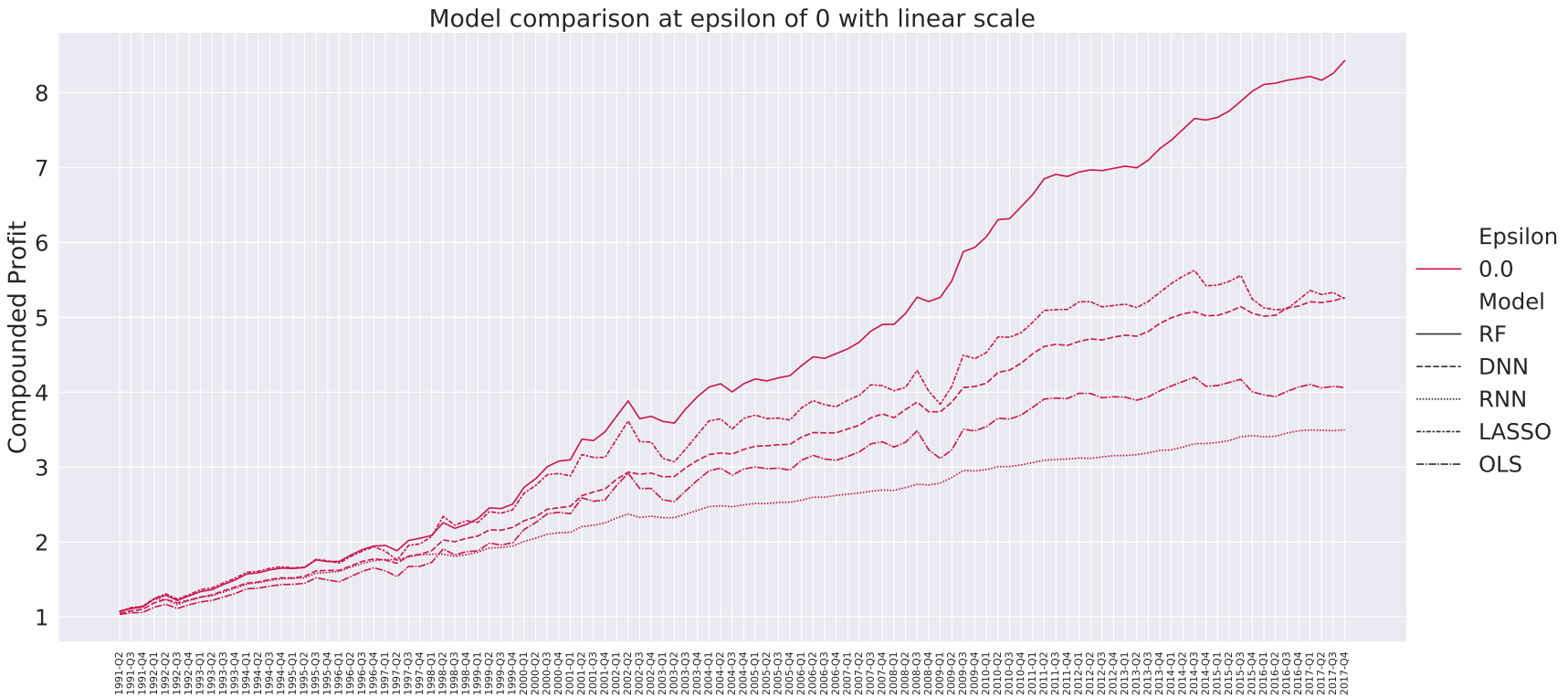
Note: This figure represents the typical architecture of a Gated Recurrent Unit (GRU), a specific type of Recurrent Neural Network, following the Pytorch notation with h_t being the hidden state, x_t the input and c_t the cell state at time t.

Table 1: Selected Compustat Variables

Category	Compustat Variables
Balance Sheet	acoq, actq, ancq, aoq, apq, atq, capsq, ceqq, cheq, csh12q, cshoq, cshprq, cshpry, cstkq, dlcq, dlttq, dpactq, icaptq, invtq, lcoq, lctq, lltq, loq, lseq, ltmibq, ltq, mibq, mibtq, ppegtq, ppentq, pstknq, pstkq, pstkrq, rectq, req, seqq, tstkq, txditcq, txpq, wcapq
Cashflow Statement	aolochy, apalchy, aqcy, capxy, chechy, dltisy, dltry, dpcy, dvy, esubcy, exrey, fiaoy, fincfy, fopoy, ibcy, intpny, invchy, ivacoy, ivchy, ivncfy, ivstchy, oancfy, prstkcy, recchy, sivy, sppivy, sstky, txdcy, xidocy
Income Statement	acchgq, cogsq, cogsy, cstkeq, doq, doy, dpq, dpy, dvpq, dvpy, epsfiq, epsfiy, epsfxq, epsfxy, epspiq, epspiy, epspxq, epspxy, epsx12, ibadjq, ibadjy, ibcomq, ibq, iby, miiq, miiy, niq, niy, nopiq, nopiy, oiadpq, oiadpy, oibdpq, opepsq, piq, piy, revtq, revty, saleq, saley, spiq, spiy, txtq, txty, xidoq, xidoy, xintq, xiq, xiy, xoprq, xopry, xsgaq

Note: This table shows the Compustat variable mnemonics for the selected variables used as features in the machine learning models.

Figure 4: Compounded quarterly returns from 1991 to 2017



Note: This figure the shows the compounded quarterly returns from 1991 to 2017 for the different machine learning models denoted in the legend using no threshold for the BHARs (i.e., $\epsilon = 0$.

Table 2: Sample Selection

Reduction Step	Rows (= firm-quarters)	Columns (= variables)	Missing Values
1. Original data set	1,567,486	720	64%
2. Selection of the most populated columns	1,567,486	121	46%
3. Dropping all rows where more than 50% of values are missing	870,492	121	23%
4. Dropping all rows where SALEQ or ATQ is missing	868,125	121	23%
5. Imputation via SoftImpute	868,125	121	0%
6. Dropping all rows where SALEQ or ATQ is 0	810,407	121	0%
7. Excluding quarters where Y_{Q-0} has no announcement date and limiting the data to events between 1991 and 2017	545.387	121	0%

Note: This table summarises the sample selection and imputation steps.

Table 3: BHAR values inside and outside of epsilon thresholds

ε	# outside ε	# inside ε	% inside	% outside
0	545,387	0	0.00	1.00
0.05	391,567	153,820	0.28	0.72
0.1	272,912	272,475	0.50	0.50
0.2	137,070	408,317	0.75	0.25
0.3	73,771	471,616	0.86	0.14
0.4	41,904	503,483	0.92	0.08
0.5	25,270	520,117	0.95	0.05

Note: This table presents the number and proportion of ground truth BHAR values inside and outside of the selected epsilon thresholds

Table 4: Model evaluation

Model	MSE	RMSE	MAE	MedAE
RF	0.055	0.234	0.153	0.099
DNN	0.058	0.241	0.16	0.106
RNN	0.07	0.265	0.174	0.111
Lasso	0.055	0.235	0.154	0.100
OLS	0.062	0.250	0.158	0.101

Note: This table shows the mean squared error, root mean squared error, mean absolute error, and median absolute error of the employed model types presented in the rows. The metrics have been computed for the collection of the quarterly test sets on a sliding window basis over the entire study period between 1991 Q2 and 2017 Q4.

Table 5: PC measure of the different models

Epsilon	**0**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
RF							
Mean	0.55	0.56	0.57	0.59	0.59	0.58	0.56
SD	(0.06)	(0.05)	(0.06)	(0.09)	(0.11)	(0.13)	(0.14)
Proportion	*1.0*	*0.74*	*0.5*	*0.25*	*0.13*	*0.08*	*0.05*
DNN							
Mean	0.53	0.54	0.55	0.56	0.56	0.56	0.54
SD	(0.03)	(0.04)	(0.05)	(0.07)	(0.09)	(0.11)	(0.12)
Proportion	*1.0*	*0.77*	*0.52*	*0.26*	*0.14*	*0.08*	*0.05*
RNN							
Mean	0.54	0.55	0.56	0.57	0.57	0.56	0.55
SD	(0.05)	(0.06)	(0.07)	(0.09)	(0.11)	(0.12)	(0.13)
Proportion	*1.0*	*0.75*	*0.51*	*0.25*	*0.13*	*0.08*	*0.05*
Lasso							
Mean	0.55	0.56	0.57	0.57	0.56	0.54	0.51
SD	(0.07)	(0.08)	(0.1)	(0.14)	(0.16)	(0.17)	(0.19)
Proportion	*1.0*	*0.72*	*0.49*	*0.25*	*0.13*	*0.07*	*0.04*
OLS							
Mean	0.54	0.55	0.56	0.56	0.55	0.54	0.52
SD	(0.05)	(0.06)	(0.08)	(0.11)	(0.13)	(0.15)	(0.16)
Proportion	*1.0*	*0.75*	*0.51*	*0.25*	*0.14*	*0.08*	*0.05*

Note: This table shows mean and standard deviation (SD) of the PC measure across models and epsilon values. The figure in italics represents the mean *proportion* of all predictions outside the given ε threshold of the entire test sample.

Table 6: Average quarterly BHAR

Epsilon	**0**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
Random Forest							
Excess Return	2.1%	6.7%	10.2%	10.4%	11.3%	11.9%	13.2%
SD	(3%)	(8%)	(9%)	(12%)	(14%)	(18%)	(22%)
IR	0.70	0.84	1.13	0.87	0.81	0.66	0.60
No. of Trades	*540,995*	*90,447*	*21,308*	*3,345*	*1,371*	*766*	*515*
DNN							
Excess Return	1.6%	3.0%	4.8%	7.0%	9.9%	8.9%	6.5%
SD	(2%)	(3%)	(5%)	(9%)	(12%)	(16%)	(17%)
IR	0.80	1.00	0.96	0.78	0.83	0.56	0.38
No. of Trades	*540,995*	*211,504*	*79,233*	*14,981*	*3,853*	*1,307*	*584*
RNN							
Excess Return	1.2%	2.5%	3.4%	4.2%	4.9%	5.2%	5.7%
SD	(2%)	(3%)	(5%)	(6%)	(6%)	(6%)	(7%)
IR	0.60	0.83	0.68	0.70	0.82	0.87	0.81
No. of Trades	*540,995*	*210,689*	*119,791*	*54,055*	*27,148*	*14,291*	*7,733*
LASSO							
Excess Return	1.7%	7.1%	9.6%	9.4%	7.6%	5.7%	2.8%
SD	(4%)	(9%)	(11%)	(13%)	(17%)	(17%)	(18%)
IR	0.43	0.79	0.87	0.72	0.45	0.34	0.16
No. of Trades	*540,995*	*41,334*	*7,900*	*1,685*	*761*	*425*	*277*
OLS							
Excess Return	1.4%	5.2%	7.3%	7.7%	6.5%	5.3%	4.3%
SD	(3%)	(6%)	(8%)	(8%)	(9%)	(10%)	(11%)
IR	0.47	0.87	0.91	0.96	0.72	0.53	0.39
No. of Tradess	*540,995*	*87,536*	*24,877*	*7,167*	*3,837*	*2,529*	*1,894*

Note: This table shows mean quarterly excess returns, standard deviation of returns (in parenthesis) and the Information Ratio for a long-short strategy that takes positions based on the sign of the predicted market reactions per model and epsilon. The last row in italics shows the *total number of trades* executed over the entire sample period.

Table 7: Final compounded value of $1 from 1991 and 2017

Epsilon	**0**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
RF	8.77	797.2	23,396	22,203	40,782	42,083	71,527
DNN	5.34	23.11	127.47	957.16	11,747	3087	239.84
RNN	3.53	13.15	32.79	69.21	136.09	189.22	287.32
LASSO	5.62	994.86	10,617	6,784	403.05	86.88	2.4
OLS	4.31	184.09	1,380	2,156	587.69	155.89	48.55

Note: This table shows the final compounded values of nominal portfolios (of size $1) invested in the respective strategies (i.e., per epsilon and model) from 1991 and 2017.

Table 8: Top 10 important variables in the random regression forest

Panel A: Overall

Variable	Importance	Description
RECCHY	223.0%	ΔAccounts Receivable
EPSX12	201.0%	EPS Excluding Extraordinary Items
CHEQ	203.0%	Cash and Short-Term Investments
INVCHY	208.0%	ΔInventory
APALCHY	206.0%	ΔAccounts Payable
CHECHY	194.0%	ΔCash and Cash Equivalents
AOLOCHY	178.0%	ΔAssets and Liabilities Other
CAPXY	174.0%	Capital Expenditures
WCAPQ	165.0%	Working Capital
DVY	121.0%	Cash Dividends

Panel B: By relative quarter

Variable	Quarter	Importance	Description
RECCHY	-1	62%	ΔAccounts Receivable
EPSX12	-4	59%	EPS Excluding Extraordinary Items
CHEQ	-1	57%	Cash and Short-Term Investments
CHEQ	-4	56%	Cash and Short-Term Investments
EPSX12	-1	56%	EPS Excluding Extraordinary Items
RECCHY	-4	55%	ΔAccounts Receivable
INVCHY	-1	54%	ΔInventory
APALCHY	-1	54%	ΔAccounts Payable
RECCHY	-3	54%	ΔAccounts Receivable
INVCHY	-4	53%	ΔInventory
RECCHY	-2	53%	ΔAccounts Receivable

Note: This table shows the ten most important variables selected by the random regression forest models in their predictions. Panel A shows the ten most important variables for the predictions measured over the entire sample period. Panel B shows the ten most important variables for the predictions by input quarter. The Δ prefix indicates the variable is measured in changes from the prior period.

Table 9: The effect of firm size

Panel A: Market capitalisation bins in million USD

Bin Name	Lower bound	Upper bound	% Observations
Micro Cap		10	4.7%
Small Cap	10	100	27.8%
Mid Cap	100	1,000	37.4%
Large Cap	1,000		28.8%

Panel B: Relative performance of size portfolios

	ε threshold						
Bin (b)	0	0.05	0.1	0.2	0.3	0.4	0.5
Micro cap	-0.12	-0.02	-0.19	-0.51	-0.81	-0.98	-0.97
Small cap	-0.21	0.49	0.11	-0.39	-0.49	-0.7	-0.77
Mid cap	0.06	-0.2	0.04	-0.14	-0.01	-0.32	0.62
Large cap	0.39	-0.04	0.01	0.11	0.32	-0.4	-0.57

Note: Panel A shows market capitalisation bins and the percentage of observations falling into those bins. The percentages do not add up to 1 as 1.3% of observations had missing values for the calculation of the market capitalisation. Panel B shows total compounded return of the $LP_{b,\varepsilon}$ portfolios relative to the total compounded return of the SP_{ε} portfolio. A positive number indicates that $LP_{b,\varepsilon}$ performs x% better than SP_{ε}, while a negative number indicates a poorer performance if the market capitalisation bin b is excluded.