Machine Learning-Based Financial Statement Analysis*

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

November 25, 2020

Abstract

This paper explores the application of machine learning methods to financial statement analysis. We compare a range of models in the machine learning repertoire in their ability to predict the sign and magnitude of abnormal stock returns around earnings announcements based on past financial statement data alone. Random Forests produce the most accurate forecasts and the highest abnormal returns. (Nonlinear) neural network-based models perform relatively better for predictions of extreme market reactions, while the linear methods are relatively better in predicting moderate market reactions. Long-short portfolios based on model predictions generate sizable abnormal returns, which seem to decay over time. Abnormal returns are robust to various risk factors and load in expected ways on size, value and accruals. Analysing the underlying economic drivers of the performance of the Random Forests, we find that the models select as most important predictors financial variables required to forecast free cash flows and firm characteristics that are known cross-sectional predictors of stock returns.

Keywords: Financial statement analysis, machine learning, earnings announcement accounting-based anomalies, random forest, neural networks

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, the Chicago Quantitative Alliance Fall Conference, 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 information in financial statements can help investors make better investment decisions, typically by assessing whether specific 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 investigates whether careful analysis of past financial statements is fruitful in uncovering information 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 investment decisions.

In this paper we apply modern machine learning methods to fundamental analysis. Although there is a long-standing literature in the engineering and decision sciences that explores machine learning for financial market predictions (e.g. Trafalis and Ince 2000; Dhar and Chou 2001; Enke and Thawornwong 2005; Tsai et al. 2011; Lee et al. 2019), accounting researchers have only fairly recently adopted machine learning methods for problems related to financial reporting such as fraud and misstatement detection (Perols 2011; Bao et al. 2020; Bertomeu et al. 2020; Brown et al. 2020), bankruptcy prediction (Barboza et al. 2017), and the forecasting of earnings and other accounting estimates (Frankel et al. 2017; Ding et al. 2020).[2]

In this paper we aim to contribute to this burgeoning field in accounting by examining the usefulness of machine learning methods for financial statement analysis. Specifically, we compare a range of linear and non-linear machine-learning models in their ability to predict

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

[2] An earlier study by Callen et al. (1996) applies a neural network for time-series forecasting of earnings. Furthermore, 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).

the market reaction to quarterly earnings announcements using only past quarterly financial statement information. The goal of the paper is to understand whether and how machine learning methods can successfully be applied to financial statement analysis; and whether these methods are able to uncover fundamental information from past financial statement data that is not fully impounded into stock prices. In the process we compare different methods and provide evidence on their investment properties. We then examine whether the selection of fundamental variables of the best performing models are consistent with economic intuition and valuation theory.

Machine learning-based approaches particularly lend themselves to fundamental analysis as they are geared towards forecasting tasks and designed to analyse highly dimensional information contained in general purpose financial statements that is inter-temporally dependent. These features of financial statement data have been shown to present challenges to investors and financial analysts when predicting future earnings and valuing securities (Bernard and Thomas 1990; Sloan 1996; Richardson et al. 2005; Wahlen and Wieland 2011).

While the application of machine learning to financial market predictions is not new outside of finance and accounting, prior work in this area has mainly focused on time-series predictions of stock returns using technical and economic indicators (see Henrique et al. 2019, for a review of the literature). Prior accounting research on fundamental analysis, on the other hand, exclusively uses linear methods over relatively short time periods and largely applies economic intuition and common statistical methods (e.g., step-wise regressions) to reduce the number of potential candidate fundamental variables in their tests (Ou and Penman 1989; Holthausen and Larcker 1992; Lev and Thiagarajan 1993; Abarbanell and Bushee 1997; Piotroski et al. 2000; Yan and Zheng 2017).

In contrast, in this paper we deliberately apply a "kitchen sink" approach to fundamental analysis by including a large set of balance sheet, income statement and cash flow statement variables in the prediction models. Machine learning algorithms offer dimensionality reduction and variable selection techniques that allow us to find the most informative

predictors from large sets financial variables and avoid common overfitting problems. Therefore, instead of narrowing the selection of variables ex-ante, we allow the models to learn from three decades of quarterly financial data what variables are most informative for forecasting stock returns around earnings announcements. This allows us to explore whether the "machine-learned" selection of financial statement variables is consistent with economic theory and with the prior evidence on accounting regularities.

We use abnormal return reactions to quarterly earnings announcements as our empirical setting 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 and earnings announcements likely contain information that affects investors' cash flow expectations resulting in such price corrections (Bernard et al. 1997).

Second, as accounting earnings inform investors' expectations of future dividends (Collins and Kothari 1989), earnings surprises normally lead to large stock price reactions (Bartov et al. 2002; Kinney et al. 2002; Skinner and Sloan 2002). These should be predictable if past financial statement data contains information about future changes in dividends. Furthermore, already minor positive or negative surprises often result in significant market reactions (Kasznik and McNichols 2002; Kinney et al. 2002; Skinner and Sloan 2002). The prior evidence thus points to potential non-linearities suggesting that machine learning methods might outperform traditional linear models in this forecasting domain. Specifically, Freeman and Tse (1992) and Kinney et al. (2002) document an S-shaped relationship between unexpected earnings and abnormal returns particularly in the extreme tails of the unexpected earnings distribution.

Third, a long-standing debate in the literature continues whether findings of predictable stock returns based on accounting variables are evidence of mispricing, and hence a violation

of efficient markets, or the result of omitted systematic time-varying risk factors (Fama 1970; Hirshleifer et al. 2012).[3] Concentrating our prediction on a short event-window around earnings announcements ensures that measurement errors related to risk premia are likely small.[4] Furthermore, as Bernard, Thomas, and Wahlen suggest *"if one can predict not 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)." Our study thus also tests whether widely available past financial statement information is fully priced by market participants in a setting in which risk-based explanations are less likely.

We train and evaluate a range of models, including OLS, Least Absolute Shrinkage and Selection Operator (LASSO), Random Regression Forests, a Deep Neural Network (DNN) and a Recurrent Neural Network (RNN) on a large panel of financial statement data from 1990 to 2017. We train the models on an expanding window of past quarters while evaluating them on their prediction accuracy for the next upcoming quarter. To avoid look-ahead bias, we only use financial statement variables from quarters t-4 to t-1 to predict the market reaction to the earnings announcement for quarter t and compare the relative out-of-sample prediction performance of the models on 'unseen' future quarters.

Prior research finds a S-shaped relationship between unexpected earnings and stock returns (Freeman and Tse 1992; Kinney et al. 2002) suggesting that the functional relationship between financial statement amounts and stock returns might be non-linear in the extreme tails. We therefore test the models' ability to predict market reactions over varying thresholds across the return reaction distribution. To illustrate the potential profitability of these forecasts, we back-test trading strategies based on long-short portfolio positions.

Overall, our findings show that machine learning methods are able to forecast the sign

[3] 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).

[4] 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).

and magnitude of abnormal returns around earnings announcements at better than random prediction accuracy. Long-short portfolios based on the model predictions earn moderate abnormal returns when trading all predicted market reactions and, given the large number of trades, are likely overall unprofitable in practice. However, when limiting trading to predictions of more extreme surprises the models earn sizable abnormal returns in a relatively moderate number of trades. This is surprising given that the predictions are solely based on commonly available past financial statement data. Random Forests perform best, on average, with the highest prediction accuracy and trading returns. When predicting absolute market reactions above a threshold of 30% the Random Forests have an average accuracy of 59% and long-short portfolios based on the models' predictions generate annual 3-factor abnormal returns of 25%. The neural network models are less accurate in their predictions, but also generate large abnormal returns particularly when predicting even more extreme market reactions. All models exhibit a higher volatility of their prediction accuracy and of their trading returns for predictions of larger market reactions suggesting that larger surprises are more difficult to predict from past performance alone, but given the size of the reactions lead to large trading profits when correctly predicted.

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, 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. In fact, the linear models perform poorly when predicting extreme market reactions producing negative returns. Furthermore, the non-linear models, particularly the neural network-based models, exhibit low volatility of their predictions and returns; and long-short portfolios based on their predictions show favorable risk characteristics. Compared to OLS the RNN and DNN produce returns with almost half the volatility, double the Sharpe ratio and significantly more positive skewness.

The abnormal returns of the quarterly long-short portfolios of the nonlinear machine learning models are robust to controls for various risk factors in asset pricing regressions including controls for size, value, momentum, investment, profitability, accruals and liquidity. Quarterly alphas range from 5.3% to 6.7% (p-values<0.01) for the Random Forest for predictions of market reactions above the 20% threshold. The Deep Neural Network generates similar abnormal returns. The long-short portfolio excess returns load in expected ways on size, value and accruals factors consistent with a value investment strategy. We further find that the abnormal returns to these strategies decay over time suggesting that the efficiency increase of the market over time with respect to the information contained in financial statements outweighs the benefit of learning from more data.

One advantage of the Random Forest is that it allows us to generate metrics to assess the importance of a particular variable for its predictions. This enables us to test whether the models' variable selections mirror economic intuition. We find that the best performing models assign the highest importance in their prediction to variables necessary to estimate free cash flows, such as earnings, changes in net working capital, CAPEX and other changes in assets. That is, the "learned" variable selection is consistent with known drivers of fundamental value and 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). Consistent with commonly practiced year-on-year and previous quarter comparisons by market participants and financial analysts, the models assign a larger weight to these variables from previous quarter and previous year financial statements when making predictions for the current quarter. These 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.

Overall, our study provides several novel insights into the use(fulness) of machine learn-

ing for financial statement analysis. First, modern machine learning models seem to be able to learn the relation between fundamental value and accounting amounts from investor reactions to accounting information releases. Second, the models select for their predictions fundamental variables consistent with investment theory. Third, the profitability of the strategies suggest that even the most basic and widely available components of past financial statements commonly used to predict fundamental value are not fully impounded into stock prices. And fourth, non-linear machine learning models seem to outperform traditional linear models particularly for predictions of more extreme, and hence more profitable, market reactions.

This study contributes to the literature on fundamental analysis in accounting and to an earlier machine learning literature for earnings and returns predictions in decision sciences in several ways.[5] We present first evidence on the viability of modern machine learning methods for financial statement analysts. We show that these models are able to learn the relation between value relevant financial statement information and stock returns and fit flexible functional forms, automate variable selection, address missing values and avoid overfitting. Beyond making a methodological contribution, our findings have several implications for fundamental analysis research that began with Ou and Penman (1989).

Our study extends the findings in prior research that pre-selects expected value relevant variables based on economic theory (Holthausen and Larcker 1992; Lev and Thiagarajan 1993; Yan and Zheng 2017; Piotroski et al. 2000) by allowing the models to learn which financial statement items are predictive of stock returns around earnings announcements for a large panel of US stocks spanning almost three decades. In doing so, the paper also sheds light on potential accounting fundamentals associated with the earnings announcement premium and the post-earnings announcement drift (Ball and Brown 1968; Bernard and Thomas 1989; Ball and Kothari 1991). The study further contributes to the related literature on accounting-based anomalies (Sloan 1996; Richardson et al. 2005) documenting that past

[5]We discuss the prior literature and our contribution in more detail in the next section.

accrual components in earnings are not fully impounded into stock prices resulting in the predictability of stock returns around earnings announcements. Our findings confirm prior findings of earnings and accrual-based predictors in the cross-section of 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 investors failing to fully incorporate commonly available financial statement information into prices. We therefore also extend the earlier literature in decision sciences that so far has reported mixed result on the usefulness of machine learning for earnings and return prediction (Falas et al. 1994; Callen et al. 1996; Dhar and Chou 2001).

We next discuss the prior literature and highlight our contribution to the field in more detail. Section 3 presents the data and discusses the estimation set-up. Section 4 discusses the research design and provides an overview of the machine learning models as well as the metrics to evaluate them. Section 5 presents the main results. The study concludes with Section 6.

2 Related Literature and Contribution

2.1 Fundamental Analysis

Fundamental analysis research is motivated by evidence that security prices fail to immediately capture publicly available accounting information (Ball and Brown 1968; Bernard and Thomas 1989; Sloan 1996). In their seminal paper, Ou and Penman (1989) (O&P) find that a range of financial ratios constructed from financial statements are informative for predictions of one-year ahead earnings changes. The study employs a logit model to estimate the probability of a positive or negative earnings change and, based on the predicted probability, constructs portfolios, which earn positive returns over the subsequent 12 and 24 month period between 1973 and 1983.

Holthausen and Larcker (1992) similarly apply a logit model using the same set of financial statement ratios as in O&P, but instead of predicting unexpected earnings, they attempt to directly model the sign of excess returns over 12 months following the release of the annual financial statement. Using step-wise regressions to select a subset of the variables from O&P, the study documents annual excess returns of 4.3% to 9.5% for a long-short trading strategy that is based on the predicted probability score of their model for the period of 1978 to 1988. The study also replicates the O&P study out-of-sample and finds no significant excess returns.

Consistent with the findings in Holthausen and Larcker (1992), that fundamental accounting variables are predictive of future abnormal returns, Lev and Thiagarajan (1993) find that their selection of fundamentals adds about 70% to the explanatory power of an earnings-based prediction model for the period of 1974 to 1988. In contrast to the previous studies they select their set of value relevant fundamental accounting variables not through a statistical search but guided by their use in financial publications such as the Wall Street Journal and Barron's, professional commentaries on corporate financial reporting, and newsletters of securities firms. Similarly, Abarbanell and Bushee (1998) also demonstrate that the information contained in financial statements provides strong signals that yield abnormal returns of 13.2% over 12 months. They further find that abnormal returns are concentrated around a three-day window around quarterly earnings announcements.[6]

More recently, Yan and Zheng (2017) use a data mining approach to examine the ability of 18,000 fundamental signals from financial statements to predict the cross-section of stock returns. The study creates various accounting ratios and combinations of about 220 variables from financial statements and finds that a large number of the generated fundamental signals are significant predictors of cross-sectional stock returns.

By applying modern machine learning methods to fundamental analysis our study con-

[6]Several other studies assess the out-of-sample validity of the original O&P study during other time periods and countries (Stober 1992; Bernard et al. 1997; Bird et al. 2001) and extend prior research adding macroeconomic and contextual variables beyond financial statements to their prediction models (Piotroski et al. 2000; Beneish et al. 2001; Mohanram 2005).

tributes to this strand of the accounting literature in several ways. First, the prior literature commonly pre-selects promising fundamental variables ex-ante based on economic intuition (Abarbanell and Bushee 1997, 1998), their use in practice (Lev and Thiagarajan 1993) or out of necessity because of too many missing values (Ou and Penman 1989). One advantage of machine learning methods is that they offer variable selection techniques that allow us to find the most informative predictors from large sets of 'unfiltered' financial variables, while at the same time avoiding common overfitting problems inherent in models that include a large number of predictors. Moreover, we employ dimensionality reduction techniques to address the missing variables problem that would otherwise prohibit the use of large numbers of variables.[7]

Second, in contrast to the prior literature our prediction models go beyond linear logit models and also exploit a variety of non-linear models available in the machine learning repertoire. Prior research points to significant non-linearities in the market response to earnings surprises (Freeman and Tse 1992; Kasznik and McNichols 2002; Kinney et al. 2002; Skinner and Sloan 2002) suggesting that machine learning methods are particularly well-suited to uncover significant relationships between fundamental variables associated with future earnings and stock returns. Instead of trying to specify the functional form between fundamental variables and stock returns, we allow the machine learning models to identify the data generating process by learning from past data.

Third, in contrast to the prior literature that generally aims to predict annual stock returns following the publication of financial statements, we focus on the market reaction to quarterly earnings announcement. Price corrections to fundamental information are more likely to occur during earnings releases. Consistent with this, Abarbanell and Bushee (1998) find that a large fraction of the abnormal returns to a fundamental strategy are observable around earnings announcements. Furthermore, concentrating our predictions on shorter

[7]Pre-selecting variables to a small number of variables with non-missing values might also reduce the generalizability of the results if variable availability coincides with particular industries. For example, most selected predictors in the Ou and Penman (1989) and subsequent studies are not available for financial services firms.

return windows compared to one or two-year stock returns makes risk-based explanations related to measurement errors or shifts in risk premia less likely.

Lastly, in our tests we use common financial statement variables that are widely available in machine readable form. Any evidence that applying such common fundamental accounting data leads to abnormal returns, would be strong sign that markets do not fully impound past fundamental information into prices. Moreover, prior research generally has estimated and tested the models over relatively short sample and hold-out periods. In contrast, in this study we provide evidence over almost three decades, training and testing the models during rolling windows with data available at that point in time. This allows us to examine whether the predictive ability diminishes over time.

2.2 Machine Learning for Earnings and Returns Predictions

A series of studies in the engineering and decision sciences apply machine learning methods to the prediction of quarterly earnings. In earlier work Falas et al. (1994) and Callen et al. (1996) compare artificial neural networks (ANN) to logit and time-series models in their ability to predict earnings per share (changes) and do not find that the ANNs significantly outperform the linear models. In contrast, Zhang et al. (2004) find that neural net based forecasts of earnings are more accurate than those of traditional linear models. The study also finds that the market reaction to earnings announcements is associated more strongly with linear EPS forecasts suggesting that market participants do not employ non-linear information processing methods to form their expectations about future earnings even though these are more accurate.[8] Similarly, Dhar and Chou (2001) compares four nonlinear models (i.e. an artificial neural network, a genetic algorithm, CART, and Näive Bayes) to linear regressions in their ability to predict earnings surprises. The study selects predictor variables from 18 technical and fundamental variables using a correlation-based heuristic. The study finds that the genetic algorithm delivers the best prediction results, followed by Näive Bayes, CART,

[8] Cao and Parry (2009) improve upon the neural network approach with a genetic algorithm that is significantly more accurate.

Neural Nets, and the linear model.

More recently, in accounting Frankel et al. (2017) train support vector regression, supervised latent Dirichlet allocation and random regression forests on the text of quarterly earnings call transcripts to predict earnings surprises. The study then uses the trained coefficients to predict analyst earnings revisions to measure the informativeness of earnings calls to analysts. The study finds that the narrative content in earnings calls contains incremental information beyond concurrent earnings, changes in cash flows and managerial guidance for future earnings. However, the study fails to find strong evidence that the earnings predictions lead to positive returns when applied to an investment strategy.

Several studies also attempt to predict stock returns directly applying various machine learning methods to accounting variables. Among earlier studies Sorensen et al. (2000) apply a classification and regression tree (CART) to explain return differences among technology stocks using a handful of accounting valuation ratios such as sales-to-price, cash flow-to-price and forward earnings-to-price. The study finds that the CART successfully discriminates between outperforming and under-performing stocks.

Using the same set of accounting ratios as the Ou and Penman (1989) study, Olson and Mossman (2003) compare OLS, logit and neural network models in their ability to predict one-year ahead cumulative abnormal returns for Canadian stocks from 1976 to 1993. Similar to Holthausen and Larcker (1992) the study uses step-wise regressions to select a subset of the accounting ratios. Due to missing accounting ratios, particularly for smaller firms, the study's sample only covers between 80 to about 200 companies each year. Nevertheless, Olson and Mossman (2003) show that neural networks generate higher long-short portfolio returns than traditional OLS and logit models in multi-class classification predictions across different return percentiles.

Subsequent studies also use ensemble methods combining different classification techniques to predict future stock returns on the basis of accounting ratios (Albanis and Batchelor 2007; Tsai et al. 2011). These studies generally find ensemble methods to produce higher

excess returns than single classification methods.

The literature in this sub-field of engineering generally uses fairly short training and test periods and fairly small sets of data largely because of computational and data constraints during the early days of this line of research. Similar to the earlier accounting studies, this literature also pre-selects a handful of accounting variables. It is therefore unclear whether the mixed results are due to the short sample periods, small samples (often from one industry) or variable selection. In contrast, in this study we examine a relatively long time-period between 1990-2017, using a comprehensive set of balance sheet, income statement and cash flow statement variables of a large sample of listed companies. Computational power and advancements in machine learning methods since the earlier studies allow us to address the variable selection and missing data problem. Furthermore, prior studies in this line of research often also do not investigate whether their best performing machine learning models in terms of prediction accuracy, also provide sufficiently strong results for a profitable investment strategy.

Similar to concurrent work in the asset pricing literature in finance that compares various machine learning methods for predictions of risk premia and monthly returns using asset pricing factors and firm characteristics (Gu et al. 2018; Chen et al. 2019), our study provides a comprehensive comparison of the usefulness of several common modern machine learning methods for financial statement analysis.

3 Data

3.1 Financial Statement Data

We begin with all quarterly variables from the Compustat North America FUNDQ file which contains about 720 identifying and balance sheet, income statement, and cash flow statement variables for 27,410 firms. We obtain all available variables between 1990 and 2017 resulting in a total of 1,567,486 observations.

Missing values of input variables pose a major problem for machine learning methods, particularly for those based on neural networks. This issue is particularly prevalent when using Compustat data for machine learning tasks over long horizons as about 63% of values in the data set are missing. Figure 1 represents a visualisation of the problem by showing the temporal and cross-sectional structure of these missing values. The rows in the figure correspond to the availability a Compustat variable over time from 1980 to 2018. The color indicates the rate of missing values for each variable in a particular year where dark red means all observations are missing, a white cell means about 50% of observations are missing and a dark green shading means all observations are available. The figure illustrates that for a large number of variables a considerable proportion of observations are missing during our sample period.

For our analysis, we first select financial statement variables where at least 50% of values exist throughout our sample period. Among the remaining observations, we then drop all firm-quarter observations for which the values for either *saleq* (total sales) or *atq* (total assets) are missing or zero. This initial selection leaves 868,125 firm-quarter observations (55% of the initial raw data set) and 121 of originally 720 available financial variables. These variables represent the most common items from the face of the financial statements and include 40 balance sheet, 29 cash flow statement, and 52 income statement variables (see Table 1).

A common strategy in the finance and accounting literature is to limit the variables of interest to those for which all observations exist. Given the large selection of variables in our setting, this approach is problematic as about 23% of values are still missing for our selection. Deleting variables associated with these remaining missing values would reduce the available sample for training our machine learning models substantially. As machine learning methods derive their 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.

An alternative strategy commonly used in the machine-learning literature is therefore

to use imputation techniques. Specifically, model based imputation methods conduct a dimensionality reduction to find redundant information in existing variables that can be exploited. We employ a matrix factorisation method called Soft-Impute.[9] Soft-Impute uses nuclear norm regularisation for matrix completion by iteratively replacing missing values by figures computed from a soft-thresholded singular value decomposition (Donoho 1995; Mazumder et al. 2010).[10]

We validate the plausibility of the imputation by comparing the distributions of the imputed values to the those of the existing values for a particular variable. While likely 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 considering variables published before the imputation period. Table 2 summarises the sample construction. The final sample consists of 545,387 firm-quarters between Q1 1990 and Q4 2017.

To avoid over-fitting the models to firms of a particular size (something that had been criticised by Greig (1992) with regards to the Ou and Penman (1989) study), we normalise our data by dividing all balance sheet items by total assets and income and cash flow statement items 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).

3.2 Data Window Construction

The training and test samples are constructed using a sliding window of five quarters. Let $X_{c,Q-n}$ denote a report of a company c filed in calendar quarter $Q-n$ where $n \in \mathbb{N}$ denotes the offset of how many quarters before Q the report was filed. The sequence of financial

[9] We also tested several other imputation methods including a neural net-based auto-encoder, MICE and wkNN, which performed inferior to Soft-Impute.

[10] We us an implementation from the *fancyimpute* package for the Python programming language. https://github.com/iskandr/fancyimpute.

statement variables spanning the four previous quarters (i.e. $X_{c,Q-4}$, $X_{c,Q-3}$, $X_{c,Q-2}$, $X_{c,Q-1}$) are the independent variables while the dependent variable is the buy- and hold abnormal return, $BHAR_{c,Q}$, in quarter Q calculated as

$$\text{BHAR}_{c,Q} \quad = \quad \prod_{k=-1}^{30}(1+R_{c,T+k}) \; - \; \prod_{t=-1}^{30}(1+R_{m,T+k}), \tag{1}$$

where $R_{c,t}$ is the stock return and $R_{m,t}$ is the return of the value-weighted CRSP market index on day t. Returns are compounded from one day before the announcement day $(T-1)$ until 30 days after the announcement $(T+30)$.

Thus, the feature vector

$$x_i = \big(X_{c,Q-4}, X_{c,Q-3}, X_{c,Q-2}, X_{c,Q-1}\big).$$

consists of the sequence of past financial statement variables and the corresponding response variable is

$$y_i = BHAR_i.$$

Our 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 predictions on the past point-in-time available financial statements.[11]

3.3 Training and Test Set

The test set is constructed to be out-of-sample and out-of-time, i.e., we use past data to train the models, and unknown future data to evaluate them. This is achieved by applying an expanding window where by 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

[11] 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.

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 that their performance can possibly improve over time as more training data becomes available. As a result models trained for later year-quarters, e.g., Q4 2017, benefit from substantially more training data than models trained for earlier year-quarters, e.g., Q1 1991.[12]

4 Research Design

4.1 Setup

Instead of biasing our selection of variables a priori informed by past findings reported in the literature, we include a large set of financial statement variables to train the prediction models. In doing so, we allow the machine learning models to learn from the data from almost three decades and to independently identify variable combinations that work best for predicting stock returns around earnings announcements. The models solely take advantage of quantitative information contained in financial statements as independent variables and ignore other commonly known market-based stock return predictors (e.g. momentum or volatility), macroeconomic variables (e.g. interest rates, GDP growth, unemployment), or textual data.

Since the aim is to predict the sign and magnitude of the market reaction, we pose the

[12] While it is likely 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 and neural network models to include the previous 4, 12, 20, or 40 quarters 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.

machine learning problem a regression task using a mean squared error (MSE) as the loss function during training. Here, for the set of training indices $\mathfrak{I}^{\text{train}}$ the training loss for the prediction sequence $\left(\hat{y}_i(\theta)\right)_{i\in\mathfrak{I}^{\text{train}}}$ and ground truth values $\left(y_i\right)_{i\in\mathfrak{I}^{\text{train}}}$ is defined as

$$\lambda^{\text{train}}(\theta, \mathfrak{I}^{\text{train}}) = \frac{1}{|\mathfrak{I}^{\text{train}}|} \sum_{q\in\mathfrak{I}^{\text{train}}} |y_i - \hat{y}_i(\theta)|^2$$

where θ is a parameter of the prediction model.

The prediction problem can be stated as the computation of a function of the form $\hat{f}_\theta : x_i \mapsto \hat{y}_i$, where $i = (c, Q)$, $x_i = (X_{c,Q-4}, .., X_{c,Q-1})$ represent the financial statement data four quarters preceding quarter Q, and $y_i = BHAR_i$ denotes the market reaction in quarter Q for company c. In this setting, learning is the task of finding the parameter θ to minimise the training loss. The difference between the various machine learning methods lies in (i) the specification of model $\hat{f}_\theta$ and (ii) the manner they tune the parameter θ of the model.

While there is a multitude of potential machine learning methods one could adopt, we have selected the following well-known supervised learning methods:[13]

a) A feed-forward Deep Neural Network (DNN) for regression (Goodfellow et al. 2016).

b) A Recurrent Neural Network (RNN) based on gated recurrent units (GRU) (Chung et al. 2014).

c) A random forest type approach (CART) (Breiman 2001).

d) Ordinary least squares (OLS) regression with a linear prediction model.

e) LASSO with a linear prediction model (Tibshirani 1996).

4.2 Training and Forecasting with a Rolling-Window

Our approach is to incrementally forecast one quarter into the future based on all the historical data available up to last quarter. This motivates our choice of training and assessment

[13]We present a more detailed explanation of how we train these models in the Online Appendix A.

of the methods on a rolling-window basis. Let $\mathfrak{I}^{\text{test}}$ denote the set of all quarters for which we desire to generate a forecast. In our models these quarters range from Q1 1991 to Q4 2017 as described in section 3.1.

For a given model, at $i \in \mathfrak{I}^{\text{test}}$, we compute a forecast

$$\hat{y}_i(\theta_i) = \hat{f}_{\theta_i}(x_i)$$

where the parameter was chosen by attempting to minimise the training loss on all available historical data up to that point, where

$$\theta_i \approx \arg\min_{\theta} \lambda^{\text{train}}(\theta, \mathfrak{I}_i^{\text{train}}) \tag{2}$$

4.3 Test Loss Evaluation Metrics

Once sequences $\left(\hat{f}_{\theta_i}(x_i)\right)_{i \in \mathfrak{I}^{\text{test}}}$ of forecasts are generated, we assess the quality of the predictions and compare them with those generated by competing methods. The machine learning literature discusses various metrics to quantify the "quality" of forecasts. We first focus on test losses measuring the prediction inaccuracy and in Section 5.2 test how the models' prediction accuracy translate to returns of a simple forecasting-based trading strategy.

We define the prediction test loss as follows: Firstly, we define the *stage-loss* $\ell : (y_i, \hat{y}_i) \mapsto l_i$ quantifying the discrepancy $l_i \in \mathbb{R}$ between an ex-post ground truth BHAR y_i and the pertaining ex-ante prediction $\hat{y}_i$. Secondly, we define a test loss function λ^{test} that converts a trajectory of observed prediction losses into a summary statistic $\lambda^{\text{test}}(\mathfrak{I}^{\text{test}})$ on a sequence of test quarters $\mathfrak{I}^{\text{test}}$:

$$\lambda^{\text{test}}(\mathfrak{I}^{\text{test}}) = \psi\big(\left(l_i\right)_{i \in \mathfrak{I}^{\text{test}}}\big) \tag{3}$$

where ψ is some function.

As a straight-forward choice we consider the square loss stage function $l_i = \ell(y_i, \hat{y}_i) =$

$|y_i - \hat{y}_i|^2$ and $\psi\Big(\Big(l_i\Big)_{i\in\mathfrak{I}^{\text{test}}}\Big) = \frac{1}{|\mathfrak{I}^{\text{test}}|}\sum_{i\in\mathfrak{I}^{\text{test}}} l_i$. In this case, the test loss simply coincides with the mean squared error (MSE).[14]

While the MSE is a common metric to compare the predictive performance across models for a regression task, it lacks the economic interpretability of accuracy measures used for classification tasks such as the percentage of correctly predicted outcomes. For this reason, we develop a new test metric, Percentage Correct (PC), which assess the degree to which significant reactions are correctly anticipated by our models. PC quantifies in how many cases the respective models predict the correct sign of the market reaction. The models are still trained using the MSE as the loss function but are subsequently evaluated also on the PC metric.[15]

We evaluate the PC metric over various abnormal return thresholds. As a large fraction of the abnormal return reactions are generally dominated by small returns, these might be the product of noise rather than systematic patterns in the data. Therefore, it is useful to distinguish market reactions by magnitude. Furthermore, from a trading perspective, prediction accuracy below certain return thresholds might not be of value if the returns do not cross the bid-ask spread or are below commissions or other transaction costs.

With this in mind, we devise our PC metric as follows:

For a set S, let $\mathbf{1}_S(x) = \begin{cases} 1, x \in S \\ 0, x \notin S \end{cases}$ be the indicator function. For a given threshold ε, $y, y' \in \mathbb{R}$, we define the stage loss of the PC metric to be

$$\ell^{\varepsilon}(y, y') = 1 - \mathbf{1}_{(-\infty,-\varepsilon]}(y)\,\mathbf{1}_{(-\infty,-\varepsilon]}(y') + \mathbf{1}_{[\varepsilon,\infty)}(y)\,\mathbf{1}_{[\varepsilon,\infty)}(y') + \mathbf{1}_{(-\varepsilon,\varepsilon)}(y)\,\mathbf{1}_{(-\varepsilon,\varepsilon)}(y'). \tag{4}$$

The stage loss is zero iff both y and y' jointly reside in either of the intervals $(-\infty, -\varepsilon], (-\varepsilon, \varepsilon)$

[14] Apart from the MSE we also compute the root mean squared error (RMSE), the mean absolute error (MAE) and the median absolute error (MedAE) (see Appendix Table IA-1)

[15] 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.

or $[\varepsilon, \infty)$ and is one, otherwise.

We then define ψ to ensure we compute the fraction of all misclassified instances where at least one value is outside of the interval $(-\varepsilon, \varepsilon)$ of negligible BHARs, yielding

$$\lambda_{PC}^{\text{test}}(\mathfrak{I}^{\text{test}}) = \frac{\sum_{i \in \mathfrak{I}^{\text{test}}} \ell^{\varepsilon}(y_i, \hat{y}_i)}{|\mathfrak{I}^{\text{test}}| - \sum_{i \in \mathfrak{I}^{\text{test}}} \mathbf{1}_{(-\varepsilon,\varepsilon)}(y_i)\, \mathbf{1}_{(-\varepsilon,\varepsilon)}(\hat{y}_i)} \tag{5}$$

where the denominator divides by the number of samples where a predicted or ground truth value is not contained in the epsilon ball around zero.

We create seven epsilon thresholds $\varepsilon \in \{0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$.[16] As a growing ε leads to an increasing amount of observations to fall within the ε threshold, it results in a test loss statistic that is not comparable across the different levels of ε. We have therefore defined the *PC* test metric 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) it is penalised for the cases where a true BHAR is outside of the ε threshold and predicted to be within.

5 Empirical Results

5.1 Prediction Performance

Table 3 presents the evaluation metrics for the predictions over the entire sample period. The results suggest that the Random Forest is the best performing model with a MSE of 0.055 and MAE of 0.153, 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

[16] We present in the Online Appendix in Table IA-2 how many BHAR observations fall inside and outside the particular threshold. For example, at $\varepsilon = 0$ all observations fall outside the threshold and thus all available market reactions are used to evaluate the prediction performance, while at $\varepsilon = 0.5$ only 5% of the observations fall outside of the threshold.

between the linear and non-linear methods are not large. Based on these evaluation metrics non-linear machine learning models do not seem to significantly outperform traditional linear models in predicting the magnitude of stock returns around earnings announcements.

Table 4 summarises the mean and standard deviation of the PC measure across various epsilon thresholds and shows the mean proportion of predictions outside the threshold. Again the differences between the linear and non-linear models are overall not large. For smaller values of $\varepsilon \in \{0, 0.05, 0.1\}$ the performance of the linear models (OLS and Lasso) is on par and, in the case of Lasso, superior to the neural networks. However, at higher levels of $\varepsilon \in \{0.2, 0.3, 0.4, 0.5\}$ the non-linear methods outperform the linear models. The Random Forest overall has the highest PC averaging 55% of the cases correctly predicted over the entire return distribution (with an epsilon $\varepsilon = 0$), and an average of 56% correctly predicted with $\varepsilon = 0.5$. At $\varepsilon = 0.2$ and $\varepsilon = 0.3$ the model correctly predicts returns to fall above these thresholds in 59% of the cases suggesting it performs best at predicting fairly large market reactions. The standard deviation of these predictions increases monotonically with increasing epsilon threshold.

The Lasso regression performs equally well to the Random Forest on more moderate market reactions up to 10% but experiences a significant decrease in PC and a significant increase of its standard deviation at higher thresholds. 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 perform better with a mean PC of 54% and 55%. This divergence in performance between non-linear and linear models with increasing epsilon threshold points to non-linear relationships between accounting amounts and stock returns in the extreme regions, 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).

The monotonically increasing variance of the PC measure for all models suggests that market reactions at larger ε levels are harder to predict. 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.

5.2 Trading Strategy

We next test the profitability of an investment strategy that takes long and short positions based on the predictions that abnormal returns fall on the positive or negative side outside the epsilon band. The positions are held from one day before the earnings announcement until 30 days after, weighted equally using a nominal quarterly portfolio size. The strategy takes a long position when $\hat{y}_t > 0$ and a short position when $\hat{y}_t < 0$. Depending on the sign of the observed market reaction y_t, these positions yield a positive or negative return, which is then summed across all positions (equally weighted) in that quarter and ε threshold to determine the overall quarterly portfolio profit.

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.[17] Figure 3 shows the mean BHAR and standard deviation of each model by epsilon threshold. Consistent with the results of the prediction accuracy over various epsilon thresholds, the top left figure shows returns to the RF model strategy peaking at $\varepsilon = 0.3$ and Lasso and OLS performing poorly when predicting more extreme market reactions at $\varepsilon > 0.2$. The remaining sub-graphs in the Figure show the average abnormal returns and standard deviations of the different models across the various thresholds. All models exhibit increasing volatility of returns with increasing epsilon, however, the neural network models seem to experience less variation around the mean. We examine the return characteristics in more detail next.

[17] 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.

5.2.1 Portfolio Characteristics

Table 5 summarises the return characteristics of the trading portfolios constructed based on the different models' predictions. Panels A to G show the results across epsilon thresholds from zero to 0.5. The return characteristics we focus on are the (annualised) raw and excess returns and volatility, Sharpe ratio, skewness, VaR, maximum drawdown, the CAPM beta and alpha as well as abnormal returns based on the Fama-French 3-factor model. Return measures do not include trading costs, but the table also reports the average number of trades per year for the respective strategy to get an initial indication of potential trading costs.

For $\varepsilon = 0$, equivalent to an environment in which every quarterly earnings announcement is traded (about 20,000 trades a year), the highest mean excess return is attained by the Random Forest at 7.3% and the lowest by the RNN at 4.2% suggesting that such strategy is likely unprofitable at these lower thresholds given the high number of trades. The returns seem to be increasing with the ε threshold, but only up to a point that varies by model, suggesting that the models have varying ability to distinguish between market reactions of higher magnitude and smaller market reactions. For example, in terms of returns the Random Forest is the best performing model producing the highest mean excess return of 27.2% and 3-factor alpha of 25.2% globally at $\varepsilon = 0.3$. These are sizeable returns coming from an average of 51 trades a year.[18]

At $\varepsilon = 0.4$ the DNN produces comparably higher excess and abnormal returns and at $\varepsilon = 0.5$ the DNN also performs relatively better. Average returns thus exhibit a concave relation over epsilon thresholds with the maximum being pushed further to the right of the epsilon distribution for the non-linear models. This return behavior confirms the findings of the accuracy metrics above that the neural network-based models perform relatively better

[18] The relatively high abnormal returns might not necessarily be attainable in practice, however. In robustness tests discussed in section 5.4 below we find that 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.

at predicting more extreme stock market reactions.

The poor performance of the linear models at the higher ε thresholds is consistent with the poor performance in the PC metric and confirms 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 at lower return thresholds.

As the volatility of returns is monotonically increasing with ε for every model the Sharpe ratios peak at lower epsilon thresholds whereby the DNN attains the highest Sharpe ratio globally of 1.5 at $\varepsilon = 0.1$. Overall, the RNN exhibits comparably lower volatility in returns across all epsilon thresholds and therefore has the relatively higher Sharpe ratios compared to the other models at the highest epsilon thresholds. The neural network models, and particularly the RNN, also show the lowest maximum drawdowns.[19] The portfolio returns are overall positively skewed at lower epsilon thresholds, but particularly for the linear models and the Random Forest become increasingly negatively skewed at higher ε.

While the previous findings presented in Table 4 suggested no major differences in the predictive ability between the RF and LASSO at lower epsilon levels, the trading results in Table 5 show that these models differ significantly in their profitability, risk characteristics, Sharpe ratio and number of executed trades.

5.2.2 Asset Pricing Regressions

To further explore the return characteristics of the model strategies we next run quarterly regressions of the long-short portfolio excess returns on several factor mimicking portfolio returns. We augment the Fama-French 5-factor model that contains the market factor (Mkt-RF), size (SMB), value (HML), profitability (RMW) and investment (CMA) (Fama and French 1992, 2015) with a momentum factor (MOM) (Jegadeesh and Titman 1993; Chan

[19] In untabulated results we also find 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 produce relatively stable returns, while OLS and Lasso suffer significant losses.

et al. 1996), liquidity (LIQ) (Pástor and Stambaugh 2003), accruals (ACC) (Sloan 1996) and several additional value factors such as cash flow-to-price (CFP), dividend-to-price (DP) and earnings-to-price (EP) (Piotroski et al. 2000).[20]

Table 6 summarizes the results for the different epsilon thresholds across Panels A to F. Several results are worth highlighting. The abnormal returns conditional on controlling for various risk factors to the strategies that base their predictions on non-linear models are highly statistically significant and monotonically increasing with increasing epsilon (except for the Random Forest). For example, the coefficients on the constant of the asset pricing regressions (alphas) for RNN and DNN with $\varepsilon = 0$ (Panel A) are 1.1% and 1.5% (p-values<0.001) and increase to 4.8% and 6.4% (p-values<0.01) for $\varepsilon = 0.5$ (Panel F), respectively. The Random Forest seems to generate the highest alpha overall of 6.7% per quarter at $\varepsilon = 0.4$. The alphas of the linear models (OLS and Lasso), on the other hand, become insignificant for $\varepsilon > 0.2$, but are positive and significant, albeit slightly smaller than for the non-linear models, at lower epsilon thresholds.

All strategies have significant exposures to the size and value factor, both of which are increasing with increasing epsilon threshold. These findings are consistent with the notion that smaller firms on average experience more extreme market reactions around earnings announcements and that an investment strategy based on financial statement data is fundamentally a value investment strategy with firms with higher book-to-market ratios being more important in predictions for larger market reactions.[21] Furthermore, the neural network models also have significantly positive coefficients on the accruals factor suggesting that accounting accruals explain part of the steady returns to the neural network models.

Overall, the machine-learned investment strategies based on fundamental analysis load in expected ways on size, value and accruals factors, but are not fully explained by their

[20] All factor mimicking portfolio returns, except for the liquidity factor, are obtained from Ken French's website. Returns to a traded liquidity factor are obtained from Robert Stambaugh's website. Factor mimicking portfolio returns except for size and liquidity are formed based on bivariate sorts on size and the respective factor.

[21] In additional results tabulated in section 5.4 we further show that firm size explains a large fraction of the abnormal returns particularly when predicting larger market reactions.

exposure to these. We next examine the persistence of alphas over time and examine in more detail in the next section what fundamental information is important in the best performing model, the Random Forest, in making profitable predictions.

5.2.3 Abnormal Returns over Time

As we train our models with more data as we move forward in time throughout our sample period, one would expect the model predictions to become more accurate and thus more profitable. On the other hand, prior work shows that abnormal returns and particularly those based on accounting-based regularities decay over time as the market learns about their discovery (McLean and Pontiff 2016). To examine the time-series profitability of our investment strategies we therefore divide the sample period into three periods, 1991-2001, 2002-2009, and 2010-2017, and compare the profitability of the investment strategies across these time periods. Figure 4 shows the hypothetical compound returns of the different investment strategies reset at the beginning of each of the time periods above. The graphs show that the long-short return series become flatter over time suggesting the market becomes more and more efficient with respect to the information contained in a fundamental analysis investment strategy.

Figure 5 further confirms the decay of the 5-factor alphas over time. The figure shows a particularly large decline during the period 2002-2009 compared the period 1991-2001 across all different models and epsilon.

5.3 Variable Importance for the Predictions

In this section we further examine which accounting variables drive the Random Forest predictions, overall the best performing models. Random Forests allow us to quantify the relative importance of an input variable for the prediction by accumulating the improvement in the splitting criterion (i.e., the sum of squares). The variable importance for the whole forest is computed as the mean of the metrics of every tree. 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. We compute the variable importance for every quarter and aggregate it over the entire sample period.

Table 7, Panel A shows the top 10 most important variables (of the 121) overall over the entire sample period (aggregated over 4 quarters) and Panel B shows the results by quarter. To aggregate we sum the importance measure of each variable over the sample period. The results in Panel A suggest that the model selects accounting variables helpful in estimating 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. The model's selection of variables (overall and by quarter) is also consistent with firm characteristics that have in the accounting-based anomalies literature been found to be associated with the cross-section of 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). 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.

Consistent with commonly practiced year-on-year and previous quarter comparisons by market participants and financial analysts, Panel B reveals that 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. These findings are in line with evidence in Bernard and Thomas (1990) that financial market participants trade around earnings announcements comparing year-on-year changes in quarterly earnings and with the time-series properties of cash and accrual components of earnings (Bernard and Thomas 1990; Sloan 1996; Abarbanell and Bushee 1998).

Overall, the results in this section reveal two novel findings. First, modern machine learning models seem to be able to learn the relation between fundamental value and ac-

counting amounts from investor reactions to accounting information releases and select for their predictions fundamental variables consistent with investment theory. Second, the profitability of the strategies suggest that even the most basic and widely available components of past financial statements commonly used to predict fundamental value are not fully impounded into stock prices. This is somewhat surprising given that these relationships are known since at least Graham and Dodd (1940).

5.4 Robustness: 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 Table 6 we find significant exposure of all strategies to a size factor that increases for predictions of more extreme market reactions. In robustness test we therefore investigate the role of firm size. For this we bin companies into size groups along their market capitalisation. We calculated the market capitalisation 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 8 depicts the selected thresholds for the bins.[22] 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 5.

[22]For 1.3% of observations the market capitalisation could not be calculated because of missing values.

The results are presented in Table 8. The results indicate that for the high ε strategies, which have achieved some of the largest abnormal returns, firms of the smallest market capitalisation are very important. The total return of the Random Forest strategy at $\varepsilon = 0.3$ scenario, which is the one with the highest excess returns is reduced by 81% if micro cap firms were excluded. Small cap firms have a similarly negative, albeit less dramatic, effect on the profitability of this strategy, cutting the returns by half if left out. 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 consistent with the increasing betas for the SMB factor presented in 6. The results also suggest, however, 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 Sharpe ratios at epsilon of 0.1.

The analysis in this section further confirms that firms of smaller market capitalisation play an important role in the abnormal profits achieved by the learning-based trading strategy.

6 Conclusions

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 past financial statement data alone. We compare various models from the machine learning repertoire including OLS, LASSO, Random Regression Forests, Deep Neural Networks and Recurrent Neural Networks over a period from Q1 1990 to Q4 2017.

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 in terms of prediction accuracy. 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 S-shaped earnings response relationship.

We further provide evidence on the investment performance of signals based on the predictions of the various models. The Random Forests produce the highest abnormal returns of 25% for predictions of market reactions of absolute magnitude of 30% or higher whereas the neural network-based models outperform the Random Forest when predicting even more extreme market reactions. The neural network-based models also experience lower volatility in their predictions and portfolio returns and thus produce the highest Sharpe ratios. The abnormal returns are robust to controlling for a variety of risk factors. Overall, these results are surprising given that the models only use widely available and machine readable past financial statement data.

Examining the robustness of the abnormal returns we find that their magnitudes decrease over time suggesting that the market has become more efficient with respect to the return predictability of fundamental data. We also find that a large fraction of the more extreme abnormal returns can be explained by very small stocks. Consistent with a value investment strategy portfolio excess returns load in expected ways on size, value and accruals factors.

We then analyse which financial variables in particular drive 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. That is, the models learn economically sensible fundamental associations between accounting amounts and stock prices. Surprisingly, the relatively large abnormal trading returns suggest that even these fundamental and widely know accounting relations are not fully impounded in stock prices.

Overall this study 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. Our findings contribute to the nascent academic literature on machine learning in accounting research by adding to our understanding of how these methods can be applied to examining how financial reporting outputs relate to firm value and their relationships to accounting-based anomalies. The study also provides insights into practical applications of machine learning techniques for financial statement analysis and as such is relevant to investment practice.

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.

Albanis, G. and Batchelor, R. (2007). Combining heterogeneous classifiers for stock selection, *Intelligent Systems in Accounting, Finance & Management: International Journal* **15**(1-2): 1–21.

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.

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

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.

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.

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.

Bertomeu, J., Cheynel, E., Floyd, E. and Pan, W. (2020). Using machine learning to detect misstatements, *Available at SSRN 3496297* .

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.

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

Callen, J. L., Kwan, C. C., Yip, P. C. and Yuan, Y. (1996). Neural network forecasting of quarterly accounting earnings, *International Journal of Forecasting* **12**(4): 475–482.

Cao, Q. and Parry, M. E. (2009). Neural network earnings per share forecasting models: A comparison of backward propagation and the genetic algorithm, *Decision Support Systems* **47**(1): 32–41.

Chan, L. K., Jegadeesh, N. and Lakonishok, J. (1996). Momentum strategies, *The Journal of Finance* **51**(5): 1681–1713.

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.

Dhar, V. and Chou, D. (2001). A comparison of nonlinear methods for predicting earnings surprises and returns, *IEEE Transactions on Neural networks* **12**(4): 907–921.

Ding, K., Lev, B., Peng, X., Sun, T. and Vasarhelyi, M. A. (2020). Machine learning improves accounting estimates: evidence from insurance payments, *Review of Accounting Studies* pp. 1–37.

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.

Enke, D. and Thawornwong, S. (2005). The use of data mining and neural networks for forecasting stock market returns, *Expert Systems with applications* **29**(4): 927–940.

Falas, T., Charitou, A. and Charalambous, C. (1994). The application of artificial neural networks in the prediction of earnings, *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, Vol. 6, IEEE, pp. 3629–3633.

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.

Fama, E. F. and French, K. R. (2015). A five-factor asset pricing model, *Journal of financial economics* **116**(1): 1–22.

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

Frankel, R. M., Jennings, J. N. and Lee, J. A. (2017). Using natural language processing to assess text usefulness to readers: The case of conference calls and earnings prediction, *Available at SSRN 3095754* .

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.

Henrique, B. M., Sobreiro, V. A. and Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction, *Expert Systems with Applications* **124**: 226–251.

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* .

Jegadeesh, N. and Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency, *The Journal of finance* **48**(1): 65–91.

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.

Lee, T. K., Cho, J. H., Kwon, D. S. and Sohn, S. Y. (2019). Global stock market investment strategies based on financial network indicators using machine learning techniques, *Expert Systems with Applications* **117**: 228–242.

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

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.

Olson, D. and Mossman, C. (2003). Neural network forecasts of canadian stock returns using accounting ratios, *International Journal of Forecasting* **19**(3): 453–465.

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.

Pástor, L. and Stambaugh, R. F. (2003). Liquidity risk and expected stock returns, *Journal of Political economy* **111**(3): 642–685.

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.

Sorensen, E. H., Miller, K. L. and Ooi, C. K. (2000). The decision tree approach to stock selection, *The Journal of Portfolio Management* **27**(1): 42–52.

Stober, T. L. (1992). Summary financial statement measures and analysts' forecasts of earnings, *Journal of Accounting and Economics* **15**(2-3): 347–372.

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

Trafalis, T. B. and Ince, H. (2000). Support vector machine for regression and applications to financial forecasting, *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, Vol. 6, IEEE, pp. 348–353.

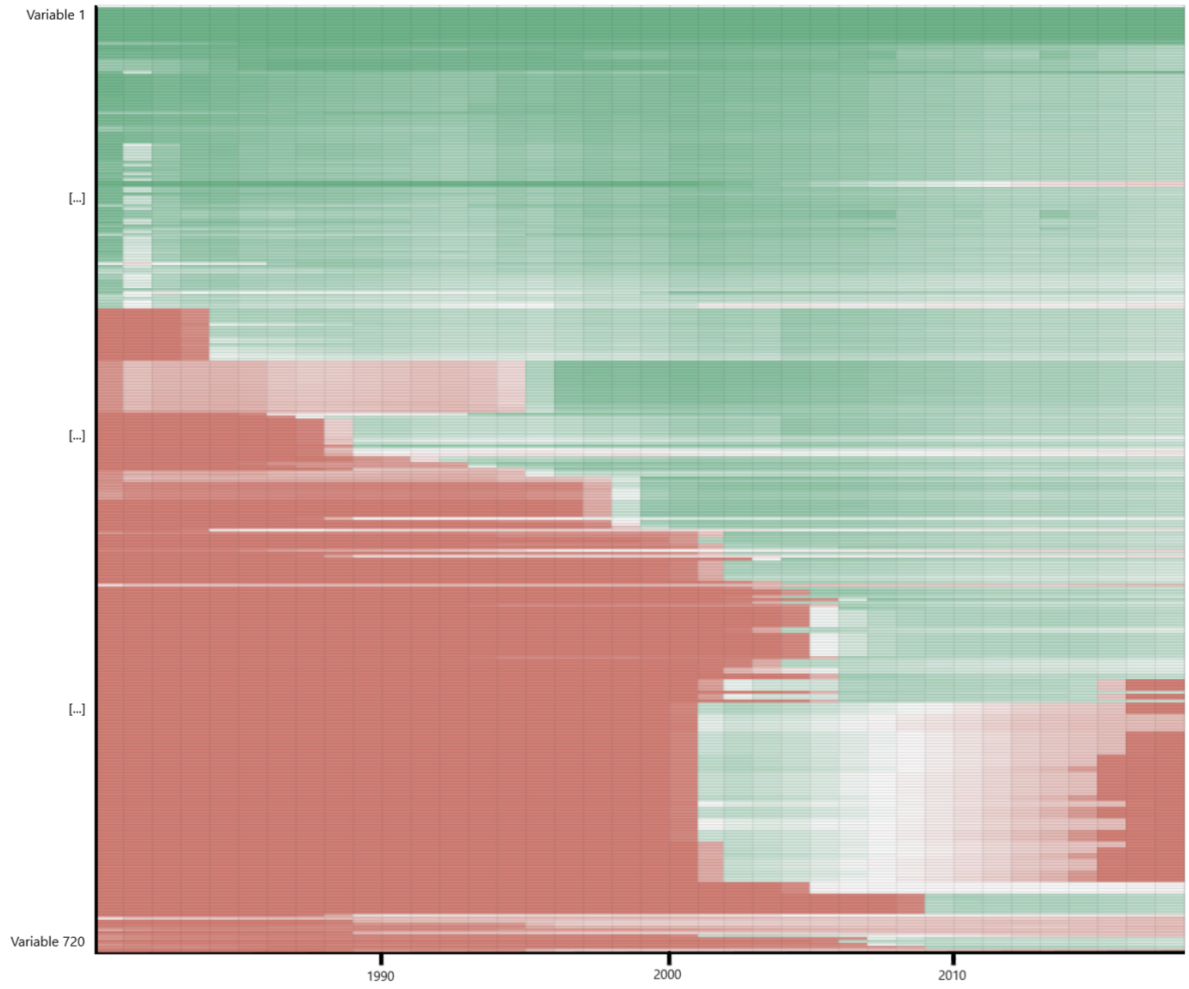
Tsai, C.-F., Lin, Y.-C., Yen, D. C. and Chen, Y.-M. (2011). Predicting stock returns by classifier ensembles, *Applied Soft Computing* **11**(2): 2452–2459.

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.

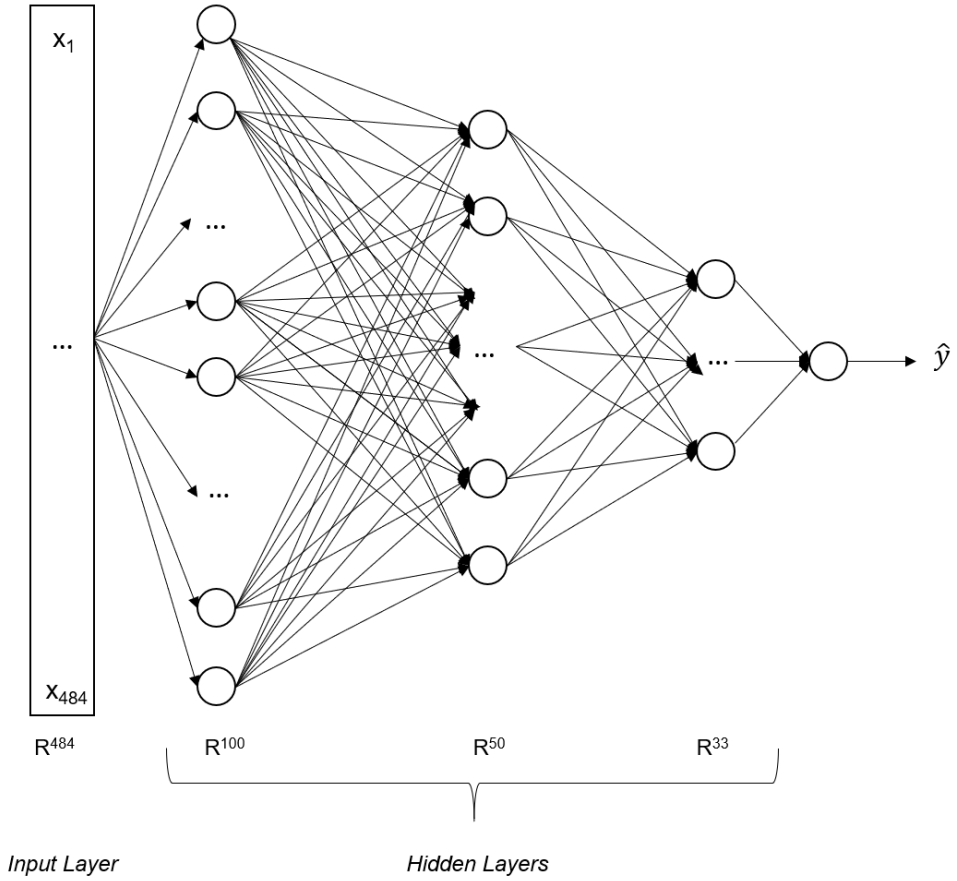
Zhang, W., Cao, Q. and Schniederjans, M. J. (2004). Neural network earnings per share forecasting models: a comparative analysis of alternative methods, *Decision Sciences* **35**(2): 205–237.

Figure 1: Visualisation of missing values in Compustat 1980-2018



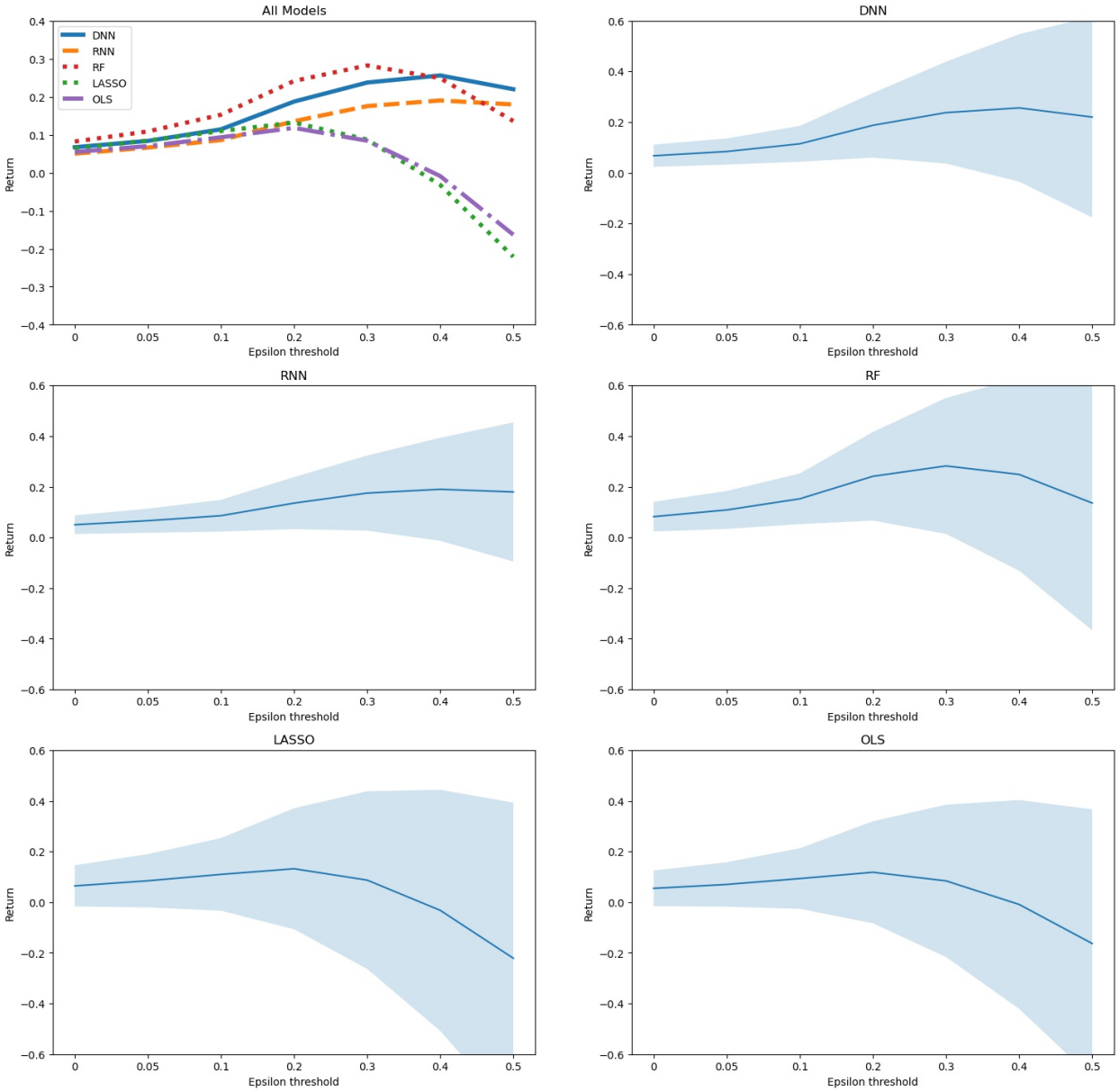
This figure visualises the structure of missing values in the Compustat FUNDQ data set. The rows correspond to a particular variable (e.g. total assets) and the columns divide years from 1980 to 2018. The colour of the cells indicates the rate of missing values of a particular variable in a year. A dark 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 in descending order based on their overall rate of missing values.

Figure 2: Visualisation of the neural network architecture



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: Model Returns and Volatility by Epsilon

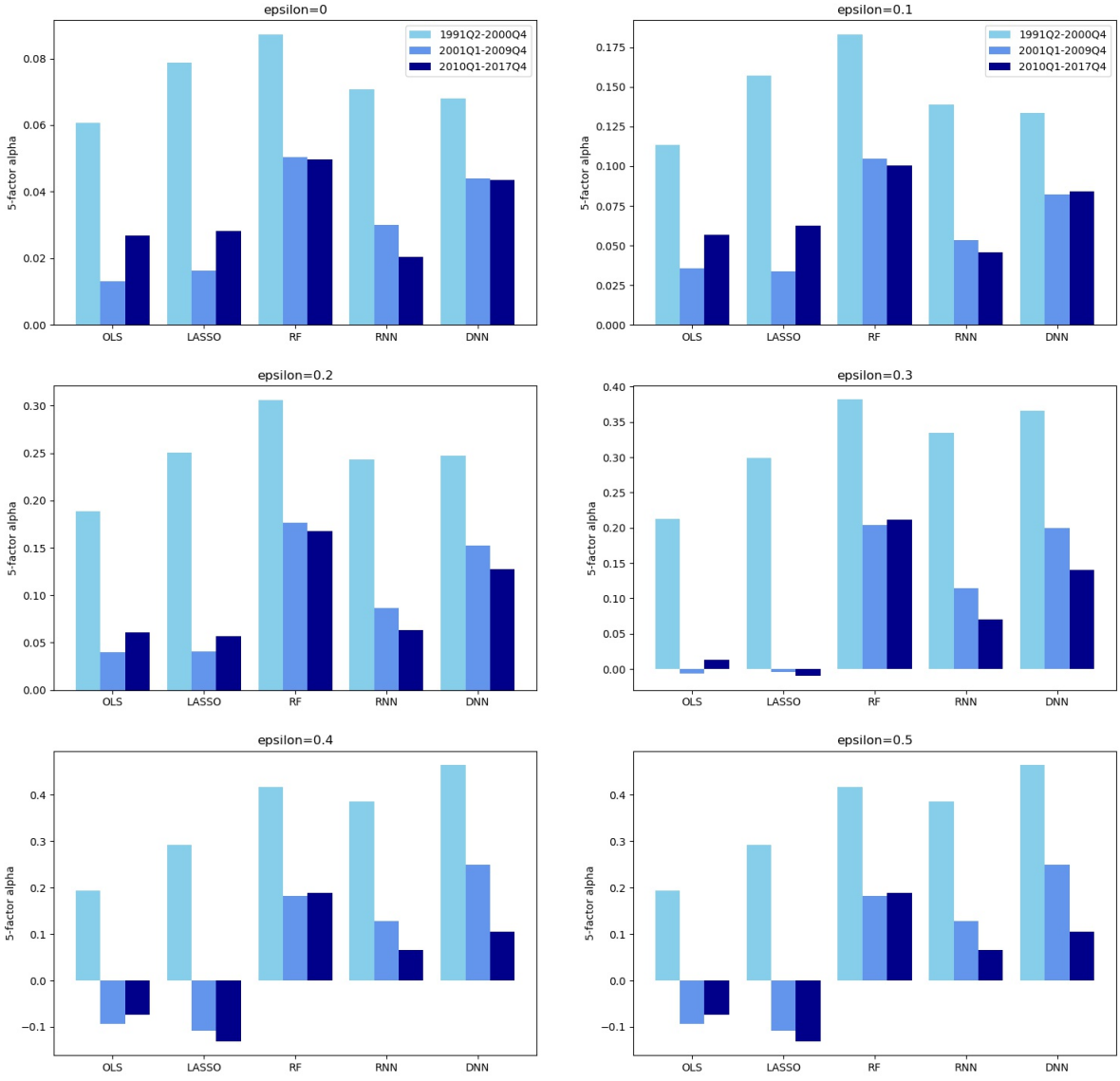


This figure shows mean returns and one standard deviation around the mean (light blue shaded areas) for each model over various epsilon thresholds.

Figure 4: Compound Returns 1991Q2-2000Q4, 2001Q1-2009Q4, and 2010Q1-2017Q4

This figure shows compound returns of $1 invested using the model predictions and reset to $1 in Q1 2001 and again in Q12010.

Figure 5: 5-Factor Alphas over Time



This figure shows abnormal returns over the periods 1991-2000, 2001-2009 and 2010-2017 from 5-factor asset pricing regressions of excess returns using predictions of the models stated on the x-axis on the market factor, size, book-to-market, momentum and liquidity. Returns to factor mimicking portfolios are from Ken French's website.

Table 1: Selected Compustat Variables Used as Features

Balance Sheet
acoq (other current assets), actq (total current assets), ancq (total noncurrent assets), aoq (other assets), apq (accounts payable),atq (total assets), capsq (share premium reserve), ceqq (total common equity), cheq (cash and short-term investments), cshoq (common shares outstanding), cstkq (ordinary stock), dlcq (debt in current liabilities), dlttq (total long-term debt),dpactq (accumulated depreciation), icaptq (total invested capital), invtq (total inventories), lcoq (other current liabilities), lctq (total current liabilities), lltq (total long-term liabilities), loq (other liabilities), lseq (total liabilities and stockholders' equity),ltmibq (total liabilities and minority interest), ltq (total liabilities), mibq (redeemable noncontrolling interest), mibtq (total noncontrolling interest, ppegtq (gross property, plant and equipment), ppentq (net property, plant and equipment), pstknq (nonredeemable preferred stock), pstkq (total preferred stock), pstkrq (redeemable preferred stock), rectq (total receivables), req (retained earnings), seqq (total parent stockholders' equity), tstkq (treasury stock), txditcq (deferred taxes), txpq (income taxes payable), wcapq (working capital)
Cashflow Statement
aolochy (net change in other assets and liabilities), apalchy (change in accounts payables and accrued liabilities, aqcy (acquisitions), capxy (capital expeditures), chechy (change in cash), dltisy (long-term debt issuance), dltry (long-term debt reduction), dpcy (depreciation and amortization), dvy (cash dividends), esubcy (equity in net loss), fiaoy (other financing activities), fincfy (net cash flow from financing), fopoy (other funds from operations), ibcy (income before extraordinary items, intpny (net interest paid), invchy (change in inventory), ivacoy (other investing activities), ivchy (increase in investments), ivncfy (net cash flow from investing activities), ivstchy (change in short-term investments, oancfy (net cash flow from operating activities, prstkcy (change in cash and cash equivalents), recchy (change in accounts receivable), sivy (sale of investment), sppivy (gain/loss on sale of PP&E), sstky (sale of common and preferred stock, txdcy (deferred taxes), xidocy (extraordinaty items and discontinued operations)
Income Statement
acchgq (cumulative effect of accounting changes), cogsq (cost of goods sold), cogsy (cost of goods sold - year to date),csh12q (common shares - 12 month moving average), cshprq (common shares - basic), cshpry (common shares - basic, year-to-date), cstkeq (dollar savings of common stock equivalent of option and warrant conversion), doq (income from discontinued operations), doy (income from discontinued operations year-to-date), dpq (depreciation and amortization), dpy (depreciation and amortization - year-to-date), dvpq (preferred dividends), dvpy (preferred dividends - year-to-date), epsfiq (diluted earnings per share), epsfiy (diluted earnings per share - year-to-date), epsfxq (diluted earnings per share excluding extraordinary items), epsfxy (diluted earnings per share excluding extraordinary items - year-to-date), epspiq (basic earnings per share), epspiy (basic earnings per share - year-to-date), epspxq (basic earnings per share excluding extraordinary items), epspxy (basic earnings per share excluding extraordinary items - year-to-date), epsx12 (basic earnings per share excluding extraordinary items - 12 month moving average), ibadjq (adjusted income before extraordinary items), ibadjy (adjusted income before extraordinary items - year-to-date), ibcomq (income before extraordinary items available to common), ibq (net income), iby (net income - year-to-date), miiq (noncontrolling interest (income account), miiy (noncontrolling interest (income account - year-to-date), niq (net income), niy (net income - year-to-date), nopiq (nonoperating income), nopiy (nonoperating income, year-to-date), oiadpq (operating income after depreciation), oiadpy (operating income after depreciation, year-to-date), oibdpq (operating income before depreciation), opepsq (earnings per share from operations), piq (pre-tax income), piy (pre-tax income, year-to-date), revtq (total revenue), revty (total revenue, year-to-date), saleq (net sales), saley (net sales, year-to-date), spiq (special items), spiy (special items, year-to-date), txtq (income taxes), txty (income taxes - year-to-date), xidoq (extrordinary items and discontinued operations), xidoy (extrordinary items and discontinued operations, year-to-date), xintq (total interest and related expenses), xiq (extraordinary items), xiy (extraordinary items, year-to-date), xoprq (total operating expense), xopry (total operating expense, year-to-date), xsgaq (selling, general and administrative expense)

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%

This table summarises the sample selection and imputation steps.

Table 3: Model evaluation

Model	**MSE**	**RMSE**	**MAE**	**MedAE**
OLS	0.062	0.250	0.158	0.101
LASSO	0.055	0.235	0.154	0.100
RF	0.055	0.234	0.153	0.099
RNN	0.070	0.265	0.174	0.111
DNN	0.058	0.241	0.160	0.106

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 4: PC measure of the different models

Epsilon	**0**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
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*
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*
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*

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 5: Return Characteristics

Panel A: Epsilon=0											
	Raw Return	Volatility	Excess Return	Sharpe Ratio	Skewness	VaR (5%)	Max Drawdown	Beta	CAPM Alpha	3-Factor Alpha	Trades per year
OLS	0.055	0.070	0.046	0.654	0.166	0.042	-0.126	0.372	0.038	0.029	20,037
LASSO	0.064	0.081	0.056	0.686	0.274	0.047	-0.135	0.405	0.048	0.037	20,037
RF	0.082	0.059	0.073	1.249	0.305	0.025	-0.088	0.222	0.069	0.063	20,037
RNN	0.050	0.037	0.042	1.134	1.264	0.009	-0.050	0.089	0.040	0.036	20,037
DNN	0.067	0.044	0.058	1.333	0.509	0.016	-0.040	0.163	0.055	0.050	20,037
Panel B: Epsilon=0.05											
	Raw Return	Volatility	Excess Return	Sharpe Ratio	Skewness	VaR (5%)	Max Drawdown	Beta	CAPM Alpha	3-Factor Alpha	Trades per year
OLS	0.070	0.088	0.061	0.699	0.019	0.053	-0.164	0.478	0.052	0.041	3,242
LASSO	0.085	0.105	0.076	0.721	0.196	0.061	-0.172	0.485	0.067	0.054	1,531
RF	0.109	0.075	0.100	1.333	0.115	0.033	-0.117	0.270	0.095	0.087	3,350
RNN	0.066	0.048	0.058	1.206	1.194	0.011	-0.070	0.116	0.055	0.051	7,803
DNN	0.084	0.052	0.075	1.454	0.336	0.019	-0.040	0.197	0.071	0.065	7,833
Panel C: Epsilon=0.1											
	Raw Return	Volatility	Excess Return	Sharpe Ratio	Skewness	VaR (5%)	Max Drawdown	Beta	CAPM Alpha	3-Factor Alpha	Trades per year
OLS	0.093	0.119	0.084	0.707	-0.022	0.074	-0.210	0.612	0.074	0.060	921
LASSO	0.110	0.143	0.101	0.703	0.076	0.087	-0.235	0.659	0.092	0.074	293
RF	0.153	0.100	0.143	1.433	-0.003	0.044	-0.155	0.340	0.139	0.128	789
RNN	0.086	0.063	0.077	1.230	1.247	0.015	-0.090	0.153	0.075	0.069	4,437
DNN	0.114	0.071	0.105	1.487	0.274	0.027	-0.060	0.266	0.100	0.092	2,935
Panel D: Epsilon=0.2											
	Raw Return	Volatility	Excess Return	Sharpe Ratio	Skewness	VaR (5%)	Max Drawdown	Beta	CAPM Alpha	3-Factor Alpha	Trades per year
OLS	0.118	0.201	0.109	0.542	-0.119	0.136	-0.349	0.968	0.102	0.079	265
LASSO	0.132	0.239	0.123	0.515	-0.007	0.159	-0.430	1.020	0.123	0.094	62
RF	0.242	0.175	0.232	1.325	-0.245	0.090	-0.278	0.646	0.228	0.209	124
RNN	0.136	0.103	0.127	1.232	1.182	0.027	-0.150	0.244	0.125	0.114	2,002
DNN	0.187	0.127	0.178	1.396	0.134	0.056	-0.117	0.445	0.173	0.159	555

Table 5 - *continued*

Panel E: Epsilon=0.3											
	Raw Return	Volatility	Excess Return	Sharpe Ratio	Skewness	VaR (5%)	Max Drawdown	Beta	CAPM Alpha	3-Factor Alpha	Trades per year
OLS	0.084	0.301	0.075	0.251	-0.131	0.222	-0.662	1.460	0.081	0.048	142
LASSO	0.087	0.351	0.078	0.224	-0.055	0.256	-0.788	1.545	0.098	0.057	28
RF	0.283	0.268	0.272	1.015	-0.327	0.159	-0.458	1.009	0.281	0.252	51
RNN	0.175	0.148	0.166	1.119	1.134	0.045	-0.220	0.362	0.166	0.149	1,005
DNN	0.238	0.201	0.228	1.133	0.143	0.102	-0.222	0.682	0.228	0.206	143
Panel F: Epsilon=0.4											
	Raw Return	Volatility	Excess Return	Sharpe Ratio	Skewness	VaR (5%)	Max Drawdown	Beta	CAPM Alpha	3-Factor Alpha	Trades per year
OLS	-0.009	0.412	-0.017	-0.041	-0.103	0.327	-0.935	2.071	0.014	-0.028	94
LASSO	-0.032	0.476	-0.040	-0.084	-0.022	0.374	-0.974	2.237	0.014	-0.038	16
RF	0.249	0.380	0.239	0.629	-0.248	0.251	-0.637	1.397	0.278	0.239	28
RNN	0.190	0.203	0.181	0.890	0.909	0.082	-0.310	0.571	0.185	0.159	529
DNN	0.256	0.291	0.246	0.846	0.167	0.164	-0.534	0.986	0.261	0.230	48
Panel G: Epsilon=0.5											
	Raw Return	Volatility	Excess Return	Sharpe Ratio	Skewness	VaR (5%)	Max Drawdown	Beta	CAPM Alpha	3-Factor Alpha	Trades per year
OLS	-0.163	0.530	-0.170	-0.322	-0.035	0.446	-0.998	2.826	-0.104	-0.149	70
LASSO	-0.221	0.614	-0.228	-0.371	0.059	0.512	-1.000	3.158	-0.126	-0.184	10
RF	0.136	0.502	0.127	0.252	-0.182	0.362	-0.833	1.932	0.213	0.168	19
RNN	0.180	0.274	0.170	0.621	0.913	0.129	-0.390	0.785	0.185	0.153	286
DNN	0.220	0.396	0.210	0.531	0.240	0.244	-0.816	1.435	0.250	0.213	22

This table reports return characteristics of the models in the rows. All return measures are annualized.Excess returns are calculated as raw returns minus the one-month Treasury bill rate. VaR(5%) is the value-at-risk measured at 95% confidence-level, maximum drawdown is calculated as the percentage loss from peak to trough in the return series, beta is the CAPM beta, CAPM alpha is the intercept from the regression of excess returns on the excess return on the market, value-weight of all CRSP firms incorporated in the US, 3-factor alpha is the intercept from the regression of excess returns on the excess return on the market, a size factor (SMB) and the value factor (HML). Benchmark and factor returns are from Ken French's website.

Table 6: Asset Pricing Regressions

Panel A: Epsilon = 0										
	OLS		LASSO		RNN		DNN		RF	
	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value
Mkt-RF	0.177	0.322	0.208	0.318	-0.078	0.424	0.042	0.720	-0.083	0.593
SMB	**0.778**	**0.001**	**0.921**	**0.001**	**0.379**	**0.004**	**0.463**	**0.003**	**0.557**	**0.008**
HML	**1.040**	**0.010**	**1.237**	**0.009**	**0.399**	**0.067**	**0.695**	**0.008**	0.410	0.235
MOM	-0.024	0.855	0.011	0.944	-0.089	0.220	-0.042	0.626	-0.096	0.406
RMW	0.117	0.713	0.121	0.744	0.019	0.914	0.090	0.662	-0.166	0.548
CMA	-0.263	0.494	-0.368	0.411	-0.086	0.682	-0.335	0.182	0.114	0.733
CFP	0.484	0.263	0.591	0.240	0.234	0.320	0.107	0.701	0.480	0.200
DP	0.026	0.917	-0.083	0.779	0.213	0.125	0.066	0.690	0.225	0.308
EP	0.092	0.859	0.164	0.786	-0.187	0.509	0.191	0.573	-0.433	0.337
ACC	0.581	0.198	0.693	0.187	**0.436**	**0.077**	**0.585**	**0.047**	0.385	0.324
LIQ	0.088	0.436	0.085	0.517	0.033	0.593	0.009	0.904	0.131	0.184
Constant	**0.010**	**0.010**	**0.011**	**0.008**	**0.011**	**0.000**	**0.015**	**0.000**	**0.019**	**0.000**
Panel B: Epsilon = 0.1										
	OLS		LASSO		RNN		DNN		RF	
	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value
Mkt-RF	0.314	0.311	0.313	0.402	-0.161	0.341	0.041	0.830	-0.115	0.667
SMB	**1.276**	**0.002**	**1.518**	**0.003**	**0.608**	**0.008**	**0.695**	**0.007**	**0.941**	**0.009**
HML	**1.758**	**0.012**	**2.200**	**0.009**	0.439	0.243	**1.089**	**0.011**	0.727	0.225
MOM	0.027	0.908	0.007	0.979	-0.203	0.107	-0.075	0.597	-0.129	0.520
RMW	0.177	0.749	0.212	0.749	-0.010	0.973	0.088	0.793	-0.153	0.749
CMA	-0.369	0.579	-0.596	0.458	-0.053	0.883	-0.378	0.354	0.263	0.648
CFP	0.720	0.336	0.837	0.354	0.371	0.361	0.170	0.709	0.667	0.304
DP	0.134	0.760	0.021	0.968	0.367	0.127	0.180	0.504	0.373	0.329
EP	0.309	0.731	0.447	0.681	-0.446	0.364	0.380	0.490	-0.460	0.556
ACC	0.910	0.244	1.148	0.224	**0.787**	**0.065**	**0.792**	**0.098**	0.542	0.423
LIQ	0.097	0.620	0.129	0.585	0.077	0.470	0.025	0.837	0.225	0.185
Constant	**0.017**	**0.010**	**0.021**	**0.008**	**0.020**	**0.000**	**0.025**	**0.000**	**0.034**	**0.000**
Panel C: Epsilon = 0.2										
	OLS		LASSO		RNN		DNN		RF	
	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value
Mkt-RF	0.433	0.416	0.354	0.574	-0.259	0.344	0.041	0.906	-0.099	0.833
SMB	**1.985**	**0.006**	**2.476**	**0.004**	**0.997**	**0.007**	**1.201**	**0.010**	**1.611**	**0.011**
HML	**2.838**	**0.018**	**3.588**	**0.012**	**1.144**	**0.063**	**1.982**	**0.011**	1.475	0.161
MOM	-0.034	0.931	-0.077	0.870	-0.294	0.151	-0.170	0.506	-0.222	0.527
RMW	0.272	0.774	0.339	0.763	0.087	0.858	0.255	0.677	-0.126	0.880
CMA	-0.501	0.661	-1.040	0.443	-0.290	0.622	-0.714	0.335	0.461	0.649
CFP	1.119	0.383	1.402	0.357	0.539	0.415	0.164	0.843	0.992	0.383
DP	0.341	0.652	0.388	0.664	0.634	0.105	0.490	0.317	0.776	0.247
EP	0.587	0.704	0.566	0.758	-0.295	0.711	0.841	0.401	-0.314	0.819
ACC	1.611	0.230	2.014	0.206	**1.422**	**0.041**	**1.594**	**0.068**	0.758	0.522
LIQ	0.166	0.621	0.154	0.699	0.117	0.499	0.005	0.981	0.317	0.287
Constant	**0.022**	**0.047**	**0.026**	**0.045**	**0.031**	**0.000**	**0.041**	**0.000**	**0.053**	**0.000**

Table 6 - *continued*

Panel D: Epsilon = 0.3										
	OLS		LASSO		RNN		DNN		RF	
	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value
Mkt-RF	0.649	0.419	0.518	0.576	-0.439	0.264	0.028	0.959	-0.143	0.843
SMB	**2.709**	**0.012**	**3.385**	**0.007**	**1.317**	**0.013**	**1.791**	**0.014**	**2.247**	**0.021**
HML	**3.797**	**0.035**	**5.087**	**0.015**	**1.832**	**0.038**	**3.084**	**0.012**	2.324	0.152
MOM	-0.090	0.881	-0.124	0.857	-0.425	0.147	-0.269	0.505	-0.361	0.504
RMW	0.125	0.930	0.144	0.930	0.075	0.915	0.125	0.897	-0.421	0.745
CMA	-0.758	0.660	-1.523	0.445	-0.435	0.606	-1.234	0.291	0.638	0.682
CFP	1.515	0.434	1.829	0.414	0.861	0.363	-0.003	0.998	1.312	0.454
DP	0.609	0.594	0.753	0.568	**1.020**	**0.069**	0.795	0.303	1.329	0.199
EP	0.304	0.896	0.498	0.853	-0.409	0.720	1.313	0.406	-0.410	0.846
ACC	2.453	0.226	3.025	0.196	**2.211**	0.027	**2.525**	**0.067**	1.050	0.565
LIQ	0.209	0.680	0.191	0.744	0.243	0.328	-0.068	0.844	0.395	0.389
Constant	0.015	0.347	0.019	0.315	**0.040**	**0.000**	**0.053**	**0.000**	**0.065**	**0.000**
Panel E: Epsilon = 0.4										
	OLS		LASSO		RNN		DNN		RF	
	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value
Mkt-RF	0.674	0.539	0.550	0.659	-0.620	0.243	-0.148	0.849	-0.353	0.731
SMB	**3.504**	**0.018**	**4.468**	**0.008**	**1.510**	**0.034**	**2.708**	**0.010**	**3.011**	**0.029**
HML	**4.873**	**0.048**	**6.655**	**0.018**	**2.595**	**0.030**	**4.360**	**0.014**	3.199	0.165
MOM	-0.340	0.677	-0.376	0.685	**-0.664**	**0.094**	-0.569	0.328	-0.657	0.392
RMW	-0.096	0.961	-0.188	0.933	-0.325	0.731	0.180	0.897	-0.781	0.670
CMA	-0.983	0.676	-1.851	0.490	-0.370	0.745	-1.839	0.275	0.979	0.658
CFP	2.313	0.383	2.582	0.391	1.525	0.234	0.414	0.826	1.781	0.474
DP	1.215	0.436	1.402	0.429	**1.528**	0.044	1.201	0.281	2.232	0.129
EP	-0.360	0.910	0.096	0.979	-0.936	0.544	1.526	0.502	-0.664	0.825
ACC	3.494	0.207	4.351	0.167	**2.818**	**0.036**	**4.061**	**0.041**	1.487	0.566
LIQ	0.419	0.545	0.357	0.651	0.291	0.385	-0.021	0.966	0.480	0.461
Constant	0.000	0.991	-0.001	0.978	**0.046**	**0.000**	**0.061**	**0.000**	**0.067**	**0.002**
Panel F: Epsilon = 0.5										
	OLS		LASSO		RNN		DNN		RF	
	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value	Coeff.	p-value
Mkt-RF	0.827	0.558	0.758	0.636	-0.858	0.235	-0.355	0.738	-0.619	0.649
SMB	**4.198**	**0.027**	**5.467**	**0.011**	**2.057**	**0.033**	**3.627**	**0.011**	**3.880**	**0.034**
HML	**5.668**	**0.074**	**8.233**	**0.023**	**3.284**	**0.043**	**5.511**	**0.021**	4.564	0.134
MOM	-0.596	0.571	-0.661	0.579	**-0.978**	**0.070**	-0.975	0.218	-0.998	0.326
RMW	-0.727	0.772	-0.820	0.774	-0.361	0.778	-0.207	0.913	-1.245	0.608
CMA	-1.567	0.606	-2.563	0.457	-0.523	0.735	-2.663	0.244	0.685	0.815
CFP	2.644	0.438	3.312	0.392	2.178	0.212	0.117	0.964	2.566	0.435
DP	1.635	0.416	1.818	0.425	**1.962**	**0.058**	1.860	0.219	2.998	0.123
EP	-1.037	0.801	-0.357	0.939	-1.318	0.530	1.950	0.528	-0.557	0.888
ACC	4.577	0.200	5.654	0.163	**3.931**	**0.032**	**5.316**	**0.049**	1.775	0.605
LIQ	0.522	0.559	0.483	0.634	0.420	0.357	-0.065	0.923	0.598	0.488
Constant	-0.028	0.339	-0.034	0.294	**0.048**	**0.002**	**0.064**	**0.004**	**0.059**	**0.037**

This table shows the results of asset pricing regressions of excess returns of the various model portfolios in the columns on Fama-French and other research factors described in the rows. Factor mimicking portfolios include the market factor (Mkt-RF), size (SMB), book-to-market (HML), momentum (MOM), operating profitability (RMW), Investment (CMA), cash flow-to-price (CFP), dividend yield (DP), earnings yield (EP), accruals (ACC) and liquidity (LIQ). All factor mimicking portfolio returns, except for LIQ are obtained from Ken French's website. LIQ factor returns are from Robert Stambaugh's website.

Table 7: 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

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 8: 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.

Online Appendix to
Machine Learning-Based Financial Statement Analysis

November 25, 2020

A 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 feed-forward *Deep Neural Network, a GRU 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 compare the non-linear machine learning methods with more traditional linear ones, we also include (linear) *ordinary least squares* (OLS) regression and *Least Absolute Shrinkage and the Selection Operator (LASSO)* approach in our benchmark comparison.

A.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 and is trained in a supervised fashion.

Similar to other types of models, its purpose is to approximate a function $\hat{f}$ that maps inputs x to a prediction $\hat{y}$ in the form of $\hat{y} = \hat{f}(x;\theta)$. Here θ is a parameter of the network. 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:

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

where $\hat{y}$ is the prediction/output, and f is a non-linear activation function (e.g. Sigmoid, tanh, ELU (see equation 2 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.[1]

A layer of a DNN consists of multiple neurons (see eq. 1) 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 (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 1 is defined as:

$$\mathrm{ELU}(x) = \max(0, x) + \min(0, \alpha\,(\exp(x) - 1)), \tag{2}$$

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).

[1] 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.

A.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:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}X_t + b_h) \tag{3}$$

$$y_t = W_{hy}h_t + b_y \tag{4}$$

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).[2]

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 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. Previous work suggests that GRUs can perform quite similarly to LSTMs in applied settings (Chung et al. 2014).

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

[2]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).

$$h_t = (1 - z_t)n_t + z_t h_{(t-1)} \tag{5}$$

$$r_t = \sigma(W_{ir}x_t + b_{ir} + W_{hr}h_{(t-1)} + b_{hr}) \tag{6}$$

$$z_t = \sigma(W_{iz}x_t + b_{iz} + W_{hz}h_{(t-1)} + b_{hz}) \tag{7}$$

$$n_t = \tanh(W_{in}x_t + b_{in} + r_t(W_{hn}h_{(t-1)} + b_{hn})) \tag{8}$$

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 IA-1 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.

A.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.[3]

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.[4]

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).

A.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).

A.4.1 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

[3]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.

[4]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.

independent variables x by a set of coefficients β. It can be written in matrix notation as:

$$y = X\beta + \varepsilon \tag{9}$$

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

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\, g(\beta) \tag{10}$$

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

A.4.2 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

$$\min_{\beta_0,\beta} \left\{ \frac{1}{N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 \right\} \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq t \tag{11}$$

with the tuning parameter $t \geq 0$.

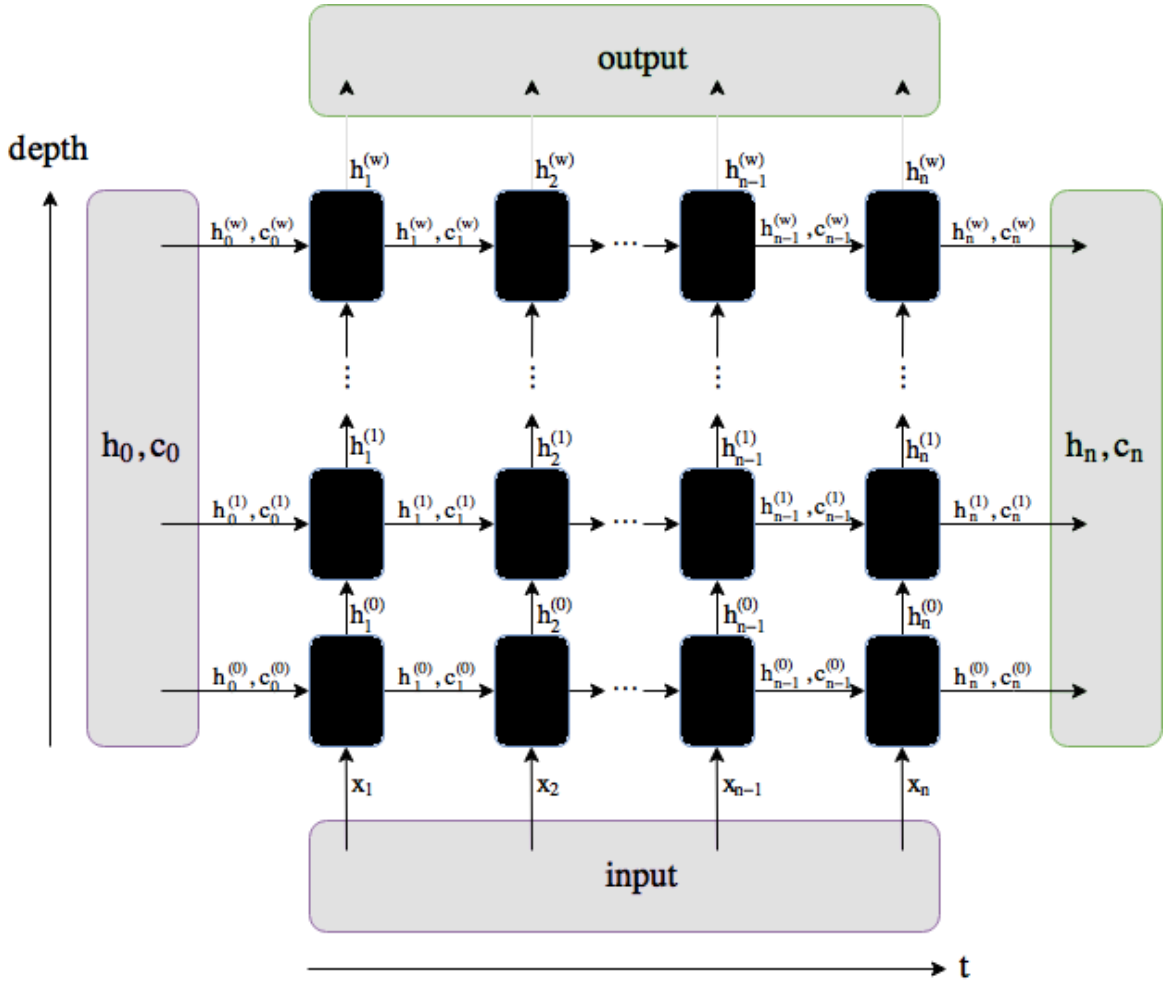
Rewriting this optimisation using the matrix notation in the Lagrangian form gives

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \alpha \|\beta\|_1 \right\} \tag{12}$$

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.

Figure IA-1: 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.

B Other Tables and Graphs

Table IA-1: Overview of alternative prediction test loss metrics

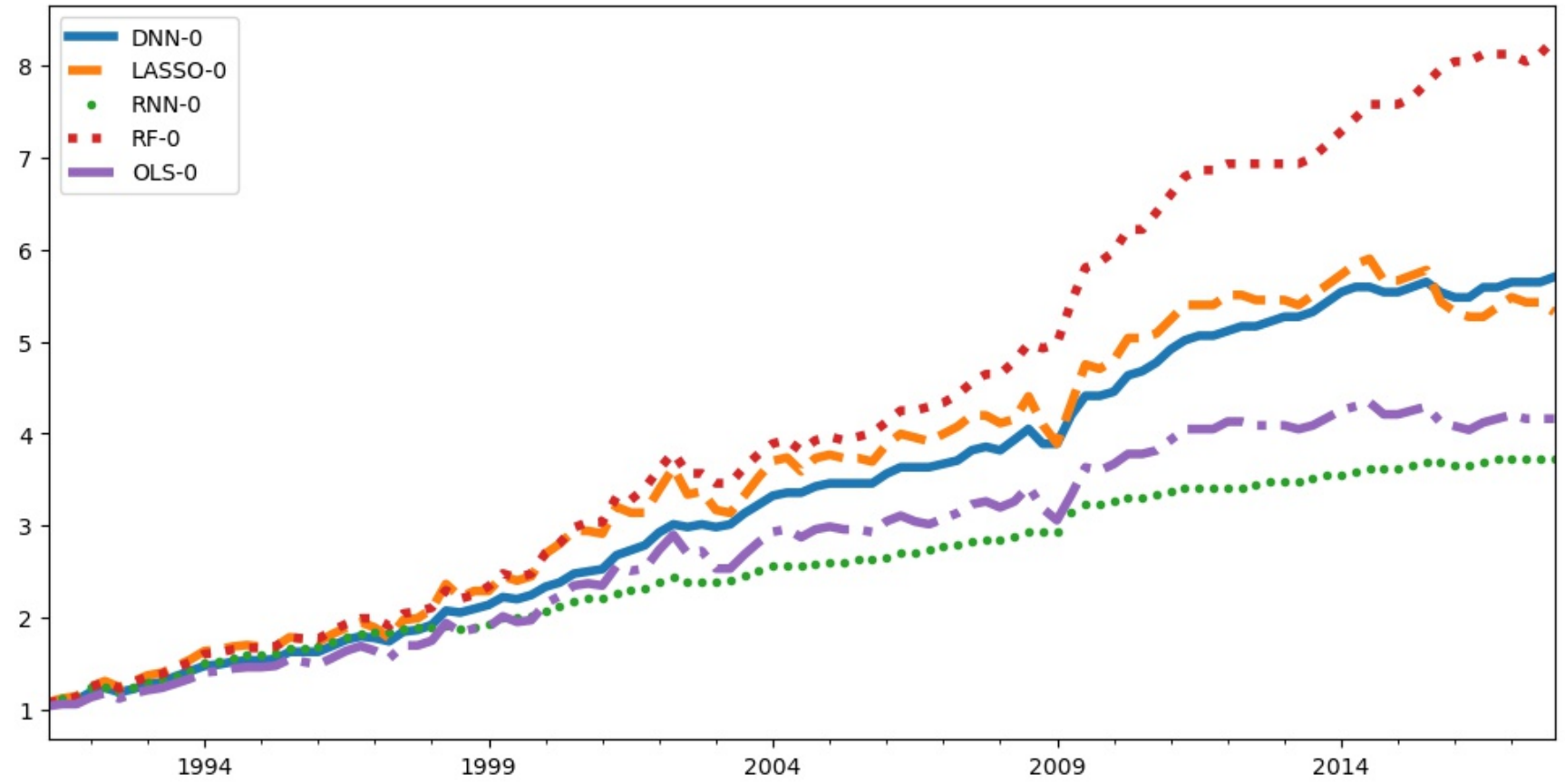
Metric λ^{test}	$\ell(y, y')$	ψ
MSE	$\lvert y - y' \rvert^2$	$\frac{1}{\lvert \mathfrak{I}^{\text{test}} \rvert} \sum_{i \in \mathfrak{I}^{\text{test}}} l_i$
RMSE	$\lvert y - y' \rvert^2$	$\sqrt{\frac{1}{\lvert \mathfrak{I}^{\text{test}} \rvert} \sum_{i \in \mathfrak{I}^{\text{test}}} l_i}$
MAE	$\lvert y - y' \rvert$	$\frac{1}{\lvert \mathfrak{I}^{\text{test}} \rvert} \sum_{i \in \mathfrak{I}^{\text{test}}} l_i$
MedAE	$\lvert y - y' \rvert$	$\text{median}(\left(l_i\right)_{i \in \mathfrak{I}^{\text{test}}})$
PC	Refer to Eq. 4	Refer to Eq. 5

Table IA-2: 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

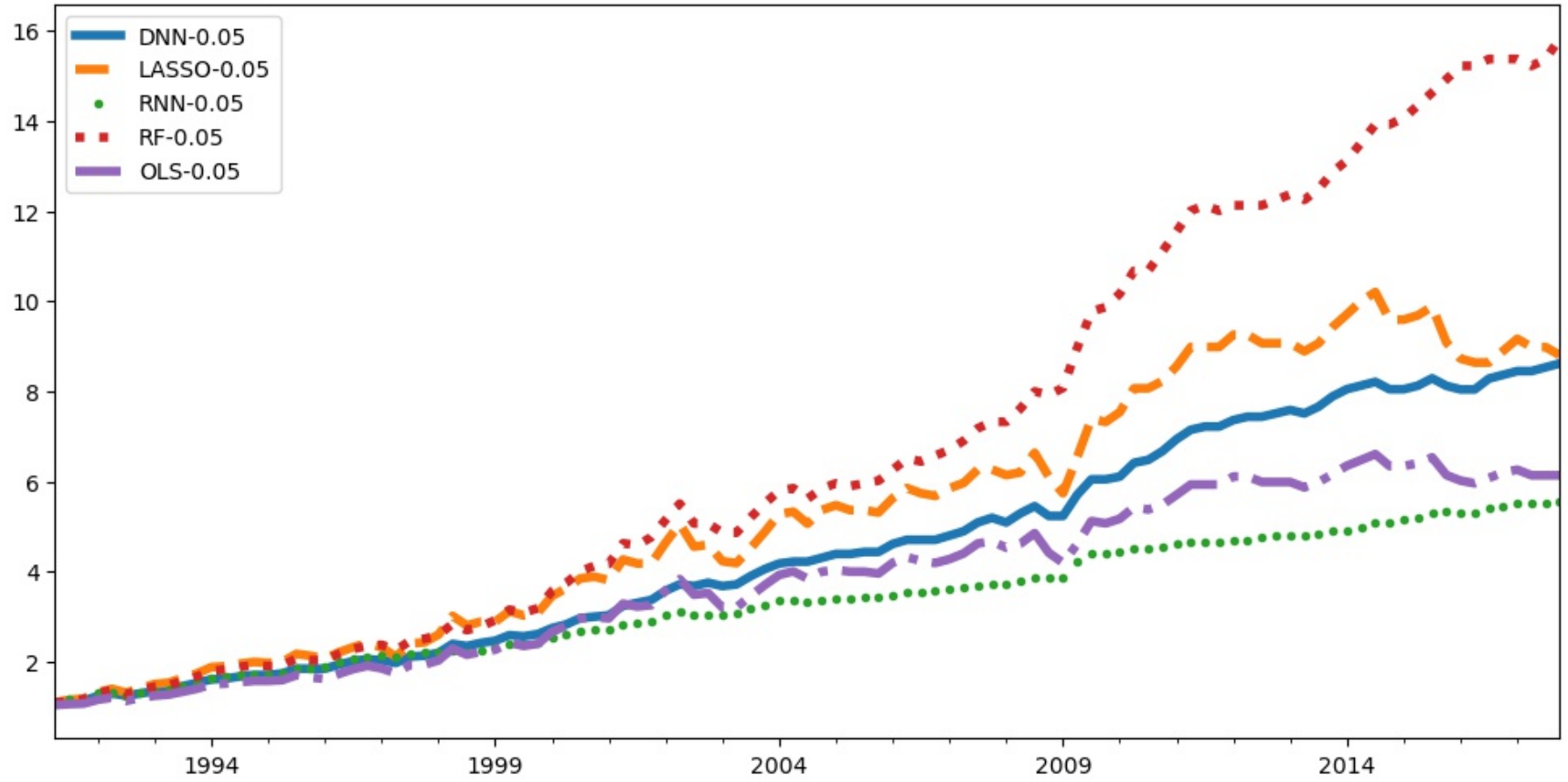
Note: This table presents the number and proportion of ground truth BHAR values inside and outside of the selected epsilon thresholds.

Figure IA-2: Compounded quarterly returns from 1991 to 2017 at epsilon = 0



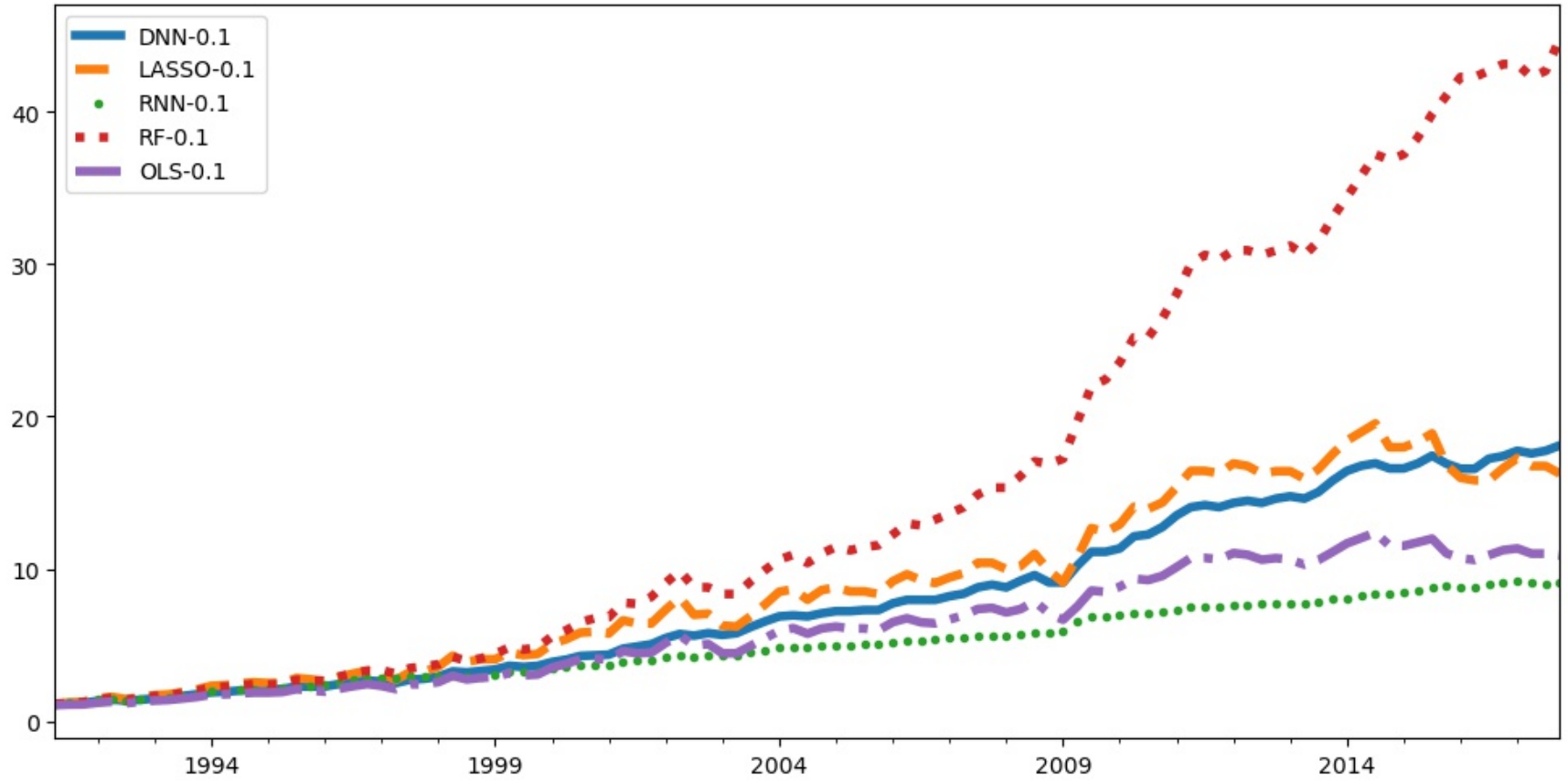
This figure the shows the compounded quarterly returns from 1991 to 2017 for the different machine learning models denoted in the legend with $\epsilon = 0$.

Figure IA-3: Compounded quarterly returns from 1991 to 2017 at epsilon = 0.05



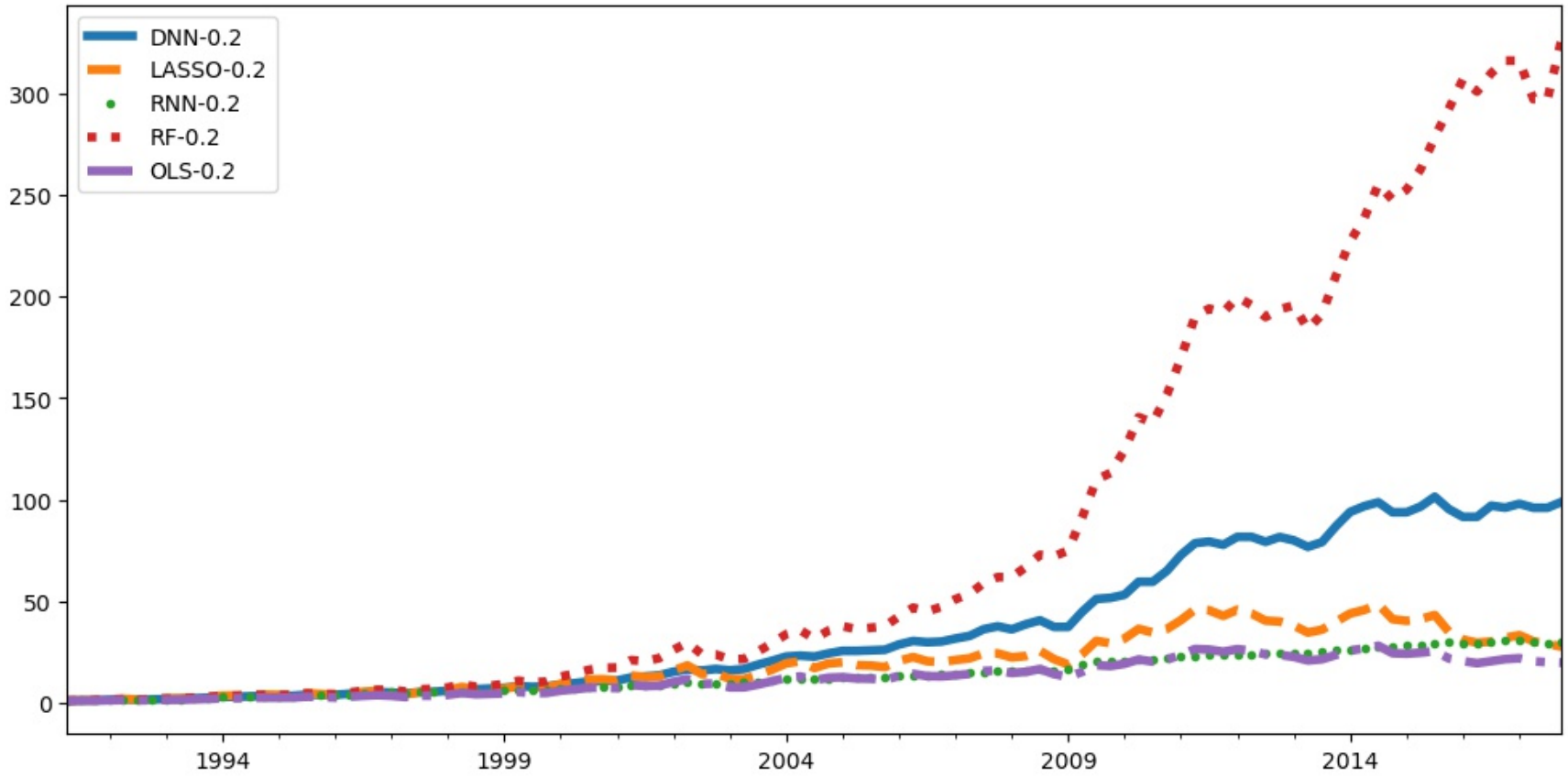
Note: This figure the shows the compounded quarterly returns from 1991 to 2017 for the different machine learning models denoted in the legend with $\epsilon = 0.05$.

Figure IA-4: Compounded quarterly returns from 1991 to 2017 at epsilon = 0.1



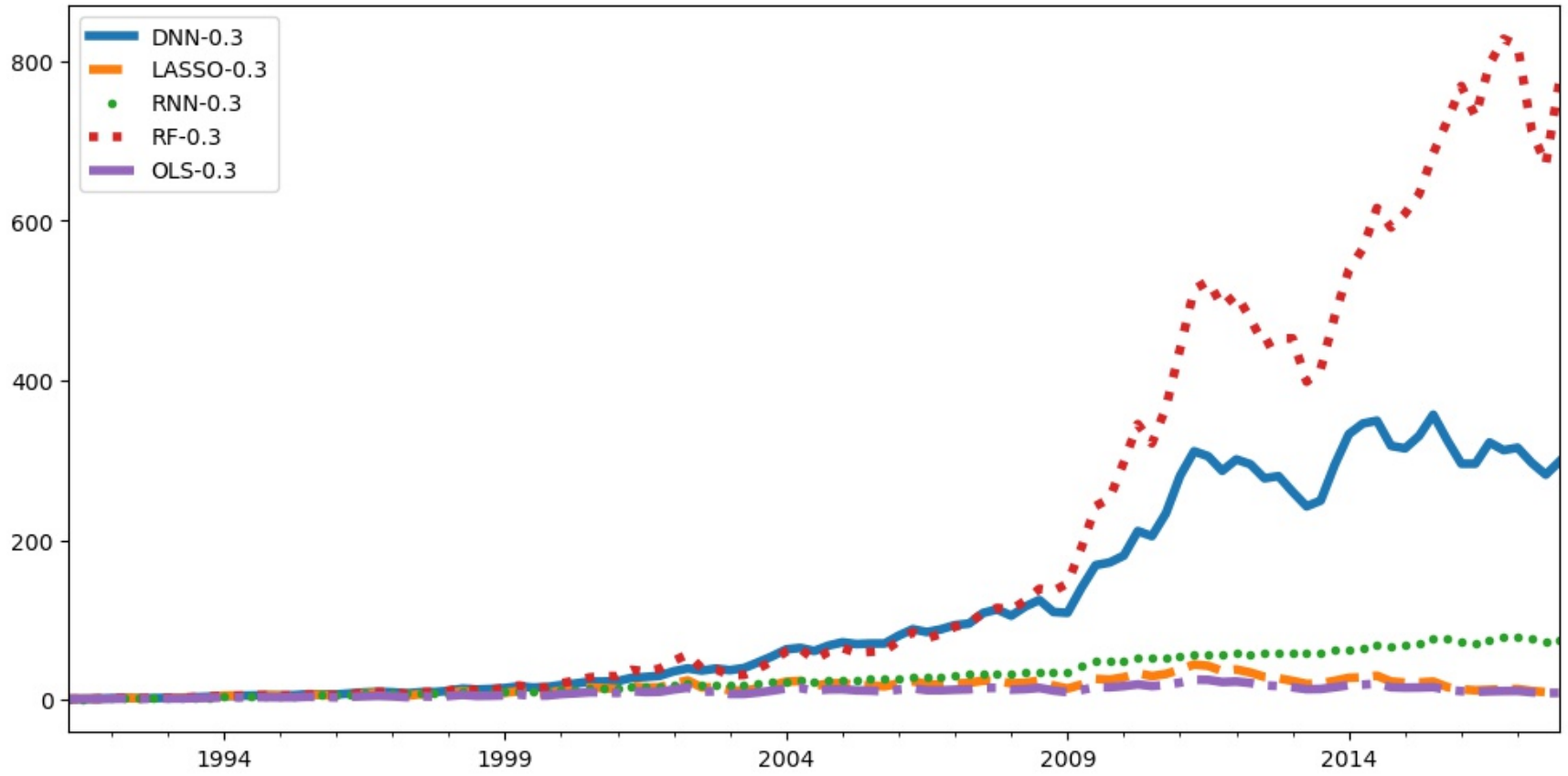
Note: This figure the shows the compounded quarterly returns from 1991 to 2017 for the different machine learning models denoted in the legend with $\epsilon = 0.1$.

Figure IA-5: Compounded quarterly returns from 1991 to 2017 at epsilon = 0.2



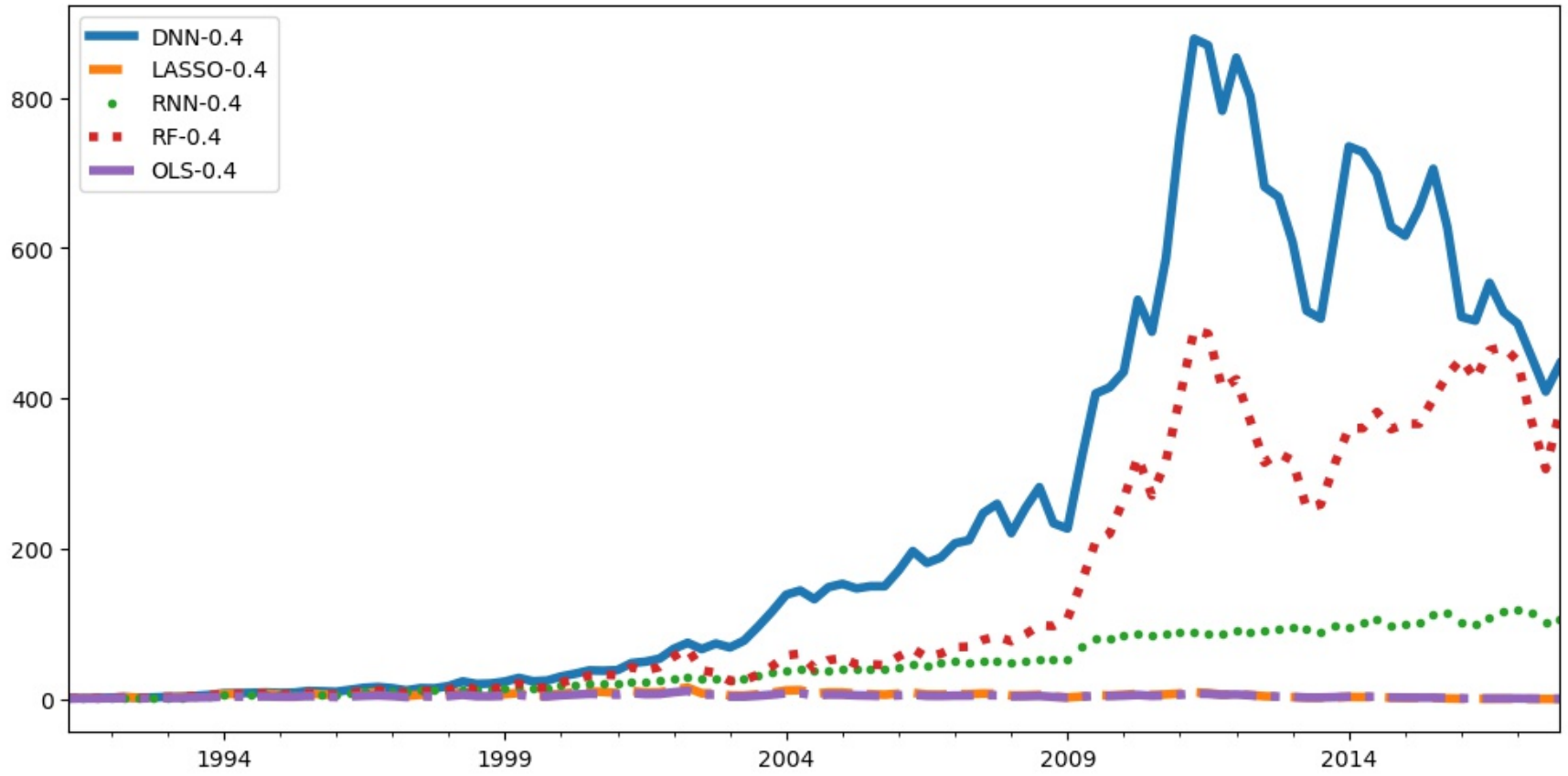
Note: This figure the shows the compounded quarterly returns from 1991 to 2017 for the different machine learning models denoted in the legend with $\epsilon = 0.2$.

Figure IA-6: Compounded quarterly returns from 1991 to 2017 at epsilon = 0.3



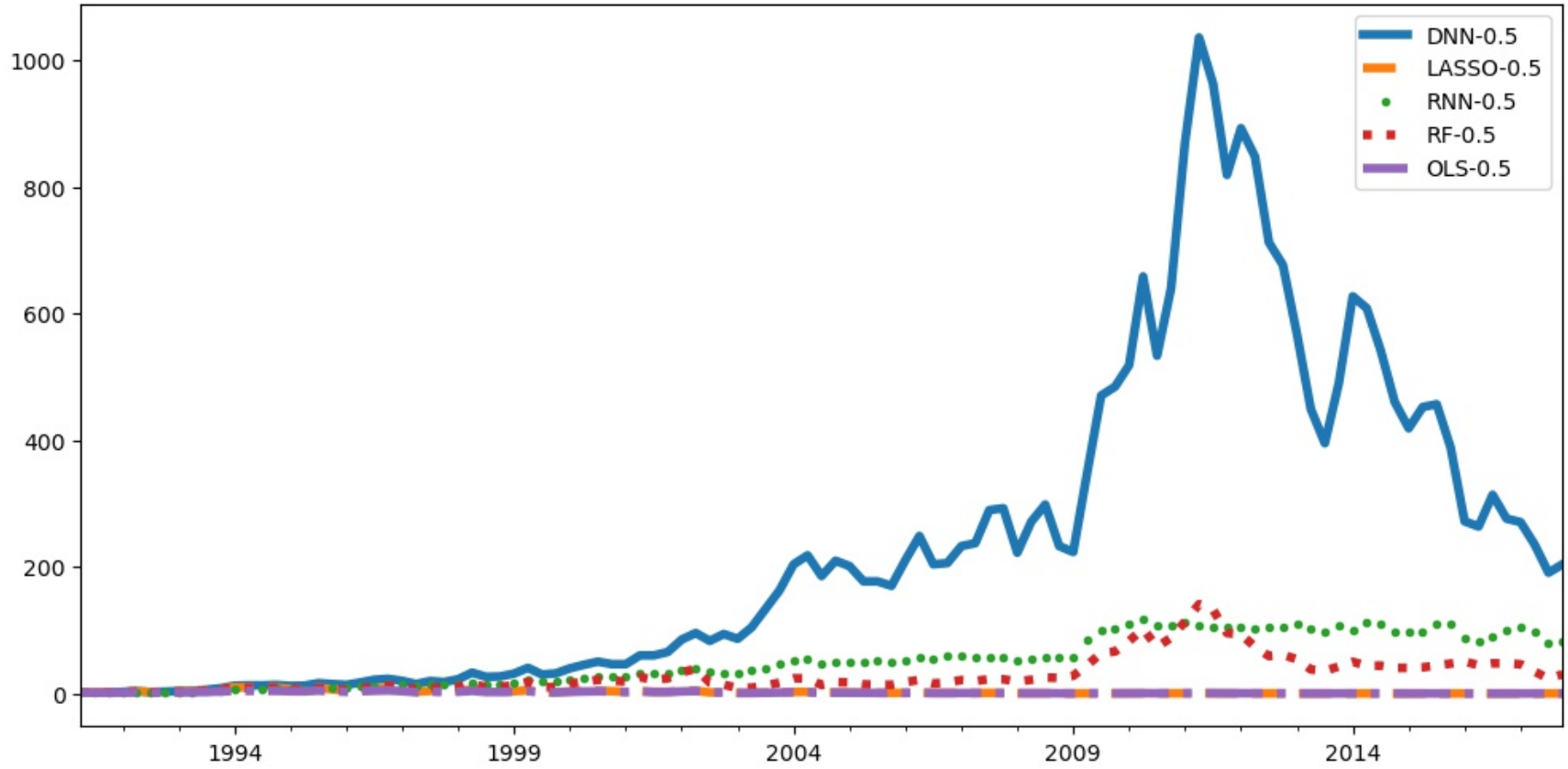
Note: This figure the shows the compounded quarterly returns from 1991 to 2017 for the different machine learning models denoted in the legend with $\epsilon = 0.3$.

Figure IA-7: Compounded quarterly returns from 1991 to 2017 at epsilon = 0.4



Note: This figure the shows the compounded quarterly returns from 1991 to 2017 for the different machine learning models denoted in the legend with $\epsilon = 0.4$.

Figure IA-8: Compounded quarterly returns from 1991 to 2017 at epsilon = 0.5



Note: This figure the shows the compounded quarterly returns from 1991 to 2017 for the different machine learning models denoted in the legend with $\epsilon = 0.5$.