Boosting over Deep Learning for Earnings

Stephen J. Choi,[1] Xinyue Cui,[2] Jingran Zhao[3]
LORA Technologies,[1,2] Hong Kong Polytechnic University[3]
stepchoi@loratechai.com,[1] xinyue.cui@loratechai.com,[2] jingran.zhao@polyu.edu.hk[3]

Abstract

Deep learning techniques have become the leading choice for applications in many fields, and finance is no exception. However, the success of such applications in other sciences does not necessarily extend to those in finance. In this paper, we examine the application of machine learning and deep learning techniques to the modeling of corporate earnings. Corporate earnings data are noisy, have limited sample sizes, and require a large amount of disparate inputs for modeling. We illustrate the success of gradient-boosting models in forecasting earnings and examine how standard deep learning techniques fall short in comparison. We also show how encoding is crucial for deep learning to be effective. Overall, our work highlights how deep learning might not be the optimal approach for addressing forecasting corporate earnings.

Introduction

Historic breakthroughs in deep learning have occurred over the past decade. Some scholars suggest that gains from deep learning have translated to the quantitative areas of finance, such as asset pricing and portfolio theory (Ozbayoglu, Gudelek, and Sezer 2020). However, such gains may prove illusory in actual application. Industry practitioners estimate that AI models during actual market implementation have a failure rate of about 90% (Kumar 2019).

In this paper, we compare the performance of deep learning and machine learning models on forecasting corporate earnings. We differentiate deep learning as models involving multiple neural network layers, models of layer *depth*. As the focal datapoint of many investors, forecasting stock earnings replicates a key task of financial analysts. Our analysis limits the input data to financial statements and financial/economic markets - we do not utilize alternative data. The efficacy of alternative data is highly related to the accessibility of the data (Kolanovic and Smith, 2019) so the input data is restricted only to those that is available to the general investing public. Finally, the objective is to benchmark AI models across a broad universe of stocks, thus we will focus on aggregate performance on a broad universe rather than anomaly detection or earnings surprises.

One of the biggest hurdles for AI models in finance is the low signal-to-noise ratio that characterizes finance (Lopez de Prado 2018). Forecasting earnings with AI models is further complicated by the lack of samples. The quarterly statements of 1,000 companies over a 5-year period represent only 20,000 samples. This tally does not account for irregular accounting periods or non-reported financials: in practice, the actual applicable sample size can be much lower. Combining these issues with high noise, we designate the situation as a noisy high-dimension low-sample size (HDLSS) problem, and it is the archetypal problem for finance and economics with low-frequency data. In this paper, we demonstrate that noisy HDLSS problem in company financials prevents deep learning from finding hierarchical structure in the data. We analyze the deficiencies of standard deep learning models and compare the efficiency of machine learning (e.g., gradient boosted tree) counterparts. Our contributions include (1) an application of gradient boosting tree models that outperforms market practitioners and models of established finance and accounting literature, and (2) evidence that the potential of deep learning models fails in comparison for modeling company financials and requires robust encoding to be effective. The remainder of the paper is organized as follows. We start by outlining our data and model structure. Then, we examine the results of gradient boosting tree models that can beat professional and academic benchmarks. This examination is followed by an analysis of the deep learning model results and their shortcomings. Finally, we discuss the implications of our research.

Methodology

Data

Our data universe consists of constituent companies of major global stock market indices, including S&P500,

Nikkei 225, Hang Seng Composite Index, and CSI 300, which account for over 950 total listed companies. Our data period is from 1998Q1 to 2019Q1 with over 25,000 quarterly statements. Our data for dependent (input) variables include three major groups. The first is fundamental data, sourced from WorldScope Fundamentals database, which is composed of reported financial information such as earnings, revenue, assets, and liabilities. From these data, we select 22 value-relevant ratios from accounting and finance literature, including Ou and Penman (1989), Lev and Thiagarajan (1993), and Abarbanell and Bushee (1997). The second group, sourced from Institutional Brokers Estimate System (I/B/E/S), is composed of forecast estimates of fundamental data made by research analysts at financial firms. The key ratios from this group involve 12 months forward earnings, revenues, and capex. The averages of the I/B/E/S forecasts, or consensus estimates, are the established industry standard and the designated performance benchmark for our analysis. For both the fundamental and I/B/E/S data, non-GAAP financial measures were used when available. The third group involves the price history of the companies' publicly listed stock as well as 16 macroeconomic ratios to reflect the different regional markets' economic performance. These data are sourced from Refinitiv (formerly Thomson Reuters) are listed in Table 1. The model's dependent variables are composed of 48 ratios and datapoints from these three data groups that are readily available and generally accepted by market practitioners as standard information for company evaluation (Valentine 2011).

Variables	Description
roic	(TTM EBITDA - TTM Capex) / (Net Debts + Market Cap)
roe	TTM Net Profit / Common Equities
ni_to_cfo	TTM Net Profit / TTM Cash Flow
inv_turnover	TTM COGS / Total Inventory
interest_to_earnings	TTM Interest Expenses / TTM EBIT
fwd_roic	(TMF EBITDA - TMF Capex) / (Net Debts + Market Cap)
fwd_ey	TMF EPS / Period Close Price
fa_turnover	TTM Net Sales / Net PPE
ebitda_to_ev	TTM EBITDA / Enterprise Value
earnings_yield	TTM EPS / Period Close Price
div_payout	TTM Dividends Per Share / TTM EPS
cash_ratio	Cash & Short-term Investment / Total Current Liabilities
capex_to_dda	TTM Capex / TTM Depreciation, Depletion & Amortization
ca_turnover	TTM Net Sales / Total Current Assets
gross_margin	TTM Gross Margin
pretax_margin	TTM Pretax Margin
debt_to_asset	Total Debts / Total Assets
cap_adequacy_ratio	Capital Adequacy Ratio
ni_ts	CAGR (*1/3/5*-year) of Net Incomes
sales_ts	CAGR (*1/3/5*-year) of Net Sales
pretax_margin_ts	CAGR (*1/3/5*-year) of Pretax Margin
cfps_ts	CAGR (*1/3/5*-year) of Cash Flow PS
stock_return	YoY and QoQ of Stock Returns
crudoil	YoY Crude Oil Prices
usinter3	YoY 3-month LIBOR
usfrbpim	YoY US Philadelphia Fed Index
usrettotb	YoY US Retail Sales & Food Service
ushouse.o	YoY US New Private Housing Starts
usgdp...d	YoY US GDP
dollar_index	YoY Dollar Index
uscnper.d	YoY US Personal Cons Expenditures
gdp	YoY GDP
unemployment	YoY Unemployment Rate
ipi	YoY Industrial Production Index
cpi	YoY Consumer Price Index
reer	YoY Real Effective Interest Rate from World Bank
index	YoY Equity Market Indices Returns
interest_rate_3m	3-month Interest Rate
interest_rate_10y	10-year Interest Rate

Table 1: Model Input Variables

For a few core datapoints, we use one-year, three-year and five-year growth rates (CAGR) in place of time series where necessary (e.g., machine learning models). For these datapoints we use non-overlapping periods—for example, average annual growth rate from five years back to three years back, from three years back to one year ago, and the past year. For stock returns, we use QoQ and 4Q back to 1Q back (=YoY) periods,

Model Structure

The inherent noise in company earnings would suggest the classification objective for the model. However, categorical errors preclude ranking distance between incorrect forecasts—the distinction between large and small forecast errors is lost. We utilize a "hybrid" ranking loss function by partitioning the target earnings into 10 equal-size deciles and assigning the decile medians to all of the target earnings in the particular decile for regression. The target earnings are defined as the relative earnings change: the yearly change in I/B/E/S trailing 12-month earnings (TTM Net Income) over the total company equity valuation (market capitalization),

$$E_t = f\left(\frac{TTM\ Net\ Income_{q+4} - TTM\ Net\ Income_q}{Market\ Capitalization_q}\right)$$

We employ five-fold cross-validation splits across *companies* to minimize data snooping and data leakage (i.e., five independent models on non-overlapping 80/20 train/validation splits of the data). For each model, the stocks for training (*and validation*) will be distinct from the stocks for testing; the train/test splits are not performed across time. The test sets are one-quarter rolling windows that form 21 sets from 2013Q1 (i.e., quarter ending 31 March 2013) to 2018Q1. The mean absolute error (MAE) and R^2 statistics are listed for the median forecast among the five cross-validation splits. MAE or optimizing for L1 loss functions are preferred due to the excess kurtosis in corporate earnings (Longstaff and Piazzesi 2004). For each section, as we analyze the performance of machine learning and deep learning models, there are at least five configurations of tests. Configuration I is limited to a randomly selected 50% sample of our universe, limiting our sample size to 475 companies. Configuration II and further configurations consist of the entire universe, expanding the dataset to the full 950 companies. Configuration III divides the dataset into industries (based on FTSE Russell ICB) and builds a separate independent model for each industry group. This corresponds to market practice, with each research analyst usually being assigned to an industry or sector. Configuration IV involves incorporating the consensus estimates to model inputs in addition to the original inputs of the previous configurations. Configurations V and VI are ad hoc configurations as and when required by the section. All of our model implementations are available at:
https://github.com/stepchoi/KDF21.git

Machine Learning: Gradient Boosting Trees

We focus on decision tree models, machine learning algorithms that have been well established among the top algorithms for data mining (Wu et al, 2008). The binary splits in tree-based models can provide more interpretability and efficiency than other machine learning or deep learning models. Boosted decision trees have been shown as the best overall learning algorithm for various types of data (Caruna and Niculescu-Mizil, 2006), and we apply LightGBM model (Ke et al, 2017), a leading-edge tree-based model that incorporates random forests (Breiman, 2001) and gradient boosting (Friedman, 2001) with a leaf-wise (best-first) strategy.

The model results are in Table 2. We see an improvement in both MAE and R^2 between the realized and model forecasts of earnings from increasing the sample size from 475 to 950 stocks (configurations I and II). Evidently, all machine learning models will improve with larger sample size for training. MAE and R^2 also improve with the implementation of industry partitions (configuration III); R^2 has improved significantly and MAE is lower than the consensus estimates. Using only widely available data, the LightGBM model outperforms the average forecasts of future earnings from the leading industry practitioners with only the innovation of modeling along industry partitions, something that is standard practice.

Configuration	**MAE**	R^2
I	0.009745	0.1498
II	0.009554	0.1794
III	0.009283	0.2305
IV	0.008612	0.2942
V	0.008855	0.2866
Consensus	0.009674	0.2583

Table 2: Gradient Boosting Tree Model Results

The I/B/E/S data that form the consensus estimates are a priori information as with all the other inputs for the LightGBM model. However, unlike the other inputs, they are not purely raw data. Instead, they are informed predictions by finance professionals, usually with additional information and insight (e.g., management changes, supply chain relationships) beyond what is available in standard financial statements. The inclusion of I/B/E/S data to the input variables should *enhance* performance; the informative value of the consensus estimates should be additive. Configuration IV confirms this, as the inclusion of consensus estimates to the model inputs improves the MAE and R^2 to levels beyond other established methods in conventional finance literature (Ball and Ghysels 2017). Table 3 presents the comparison of MIDAS - Mixed Data Sampling (Ghysels, Santa-Clara, and Valkanov 2002), an econometric regression method with substantial literature in economic and finance (Andreou, Ghysels, and Kourtellos 2013), with the corresponding statistics of configuration IV. Both results are from models that incorporate similar historical data and consensus estimates to forecast future earnings. Configuration IV is able to identify the value of I/B/E/S estimates to achieve a lower median absolute error ratio - the preferred statistic of the MIDAS earnings forecast research. (MABER: the ratio of median absolute error of model results over median absolute error of analysts' predictions.)

Model	MABER
Configuration IV	0.857
Ball & Ghysels (2018)	0.898

*MABER median absolute ratio.

Table 3: MIDAS Results Comparison

This level of success is not specific to LightGBM but also applies to other gradient boosting algorithms. Configuration V is the equivalent application of XGBoost (Chen and Guestrin 2016), another recently developed model that incorporates gradient boosting. The XGBoost model fares slightly worse than LightGBM but still maintains accuracy levels significantly above the consensus benchmark. The limited sample size seems to suffice for gradient boosting trees to address the noisy HDLSS problem.

Deep Learning: Fully Connected Networks and Recurrent Neural Networks

We initiate the deep learning application with a fully connected neural network (FCN), the standard deep learning architecture. FCN is composed of N-layers, with each having nodes of bias/threshold weight parameters and an activation function. Each node in one layer is connected to the nodes in the subsequent layer. The nonlinear activation functions provide the capacity to estimate non-linear behavior, and full connectedness provides the potential for complex structure. With multiple layers, Lapedes and Farber (1987) and Cybenko (1988) asserted that the neural network could approximate any function: The FCN is a more powerful model that should be able to span LightGBM or XGBoost model capabilities. However, as we find, this is more of a “curse” than a blessing.

Our model is composed of three to six layers with four to 16 nodes per layer. Hyperparameter optimization is performed through Hyperopt (Bergstra et al. 2013), a sequential model-based hyperparameter optimization interface based on Tree-of-Parzen-Estimators. (The hyperparameters are optimized on the *validation* sets.) The results for configuration I imply near randomness, with significant improvement in results in configuration IIa by doubling the training sample size (Table 4). To avoid potential overfitting, we forcefully decrease the model size by reducing the layers and nodes to lower ranges during Hyperopt in configuration IIb. Reducing the model parameters with fewer layers and fewer nodes (from 7500+ to below 1200) in configuration IIb improves both MAE and R^2 significantly; narrow (minimal nodes) and shallow (minimal layers) improves FCN performance, illustrating the curse of dimensionality. Further partitioning of the data by industry in configuration III reverts back to near randomness—the underperformance from smaller datasets is almost trivially self-evident. All three configurations struggle against the basic gradient boosting results. The potential for non-linear and complex structures in deep learning seems to be blunted by noisy HDLSS, and even reducing the model complexity is not adequate.

The addition of consensus estimates does not result in significant improvement (configuration IV); while both MAE and R^2 have improved, they are still far worse than every configuration of the previous section. Even relatively simple FCN models cannot adequately capture the informative value of consensus estimates. The only comparable results are in configuration V. Here we lean on the LightGBM models of the previous section to provide the top 15 important features, such as quarterly earnings and revenue. Only then does the FCN model overcome the curse of high dimensionality and produce meaningful results. Nonetheless, it still fails in comparison with gradient boosting models.

Table 4: Deep Learning Model Results.

Configuration	MAE	R^2	Model Parameters
I	0.010870	0.0799	1,831
II a	0.010602	0.0928	7,517
II b	0.010355	0.1175	1,167
III	0.010648	0.0545	1,194
IV	0.009960	0.1800	1,220
V	0.009264	0.2464	1,873
VI	0.009302	0.2734	1,643

We propose two explanations for the underwhelming results. The first is more evident: FCN models do not explicitly model time series. Multiple growth rates were used as replacements for panel data. The second explanation involves the heterogenous data types of model inputs. Recent research (Poggio, Banburski, and Liao 2020) shows that applying convolutional neural networks (CNN) on data with hierarchical locality leads to the exponential costs of dimensionality becoming more linear. Our data unfortunately lack the structure or homogeneity of visual pixels or even stock price data. By combining macroeconomic data, stock price history, and company financials, the total input dataset retains varying distributions and attributes with divergent frequencies (quarterly or monthly or daily records). The exponential dependence on the number of parameters for accuracy prevents neural networks from finding the hierarchical structure generated by gradient boosting models. This outcome is most evident in the lack of significant improvement in accuracy of configuration V with the addition I/B/E/S consensus estimates. Deep learning models cannot even identify “good” information input.

We address both deficiencies in configuration VI by incorporating recurrent neural networks (RNN) structures. We also repeat the use of the top 15 input variable (features) provided by LightGBM models and revert to standard year-over-year rates for any previous compound annual growth rates. These features are ranked by the average LightGBM ranking across the different industry partition models. The top 15 were chosen because the other features were consistently ranked lower across all the models. The features are listed in Table 5. The times series representation further reduces the input dimensionality to 10 features (including I/B/E/S consensus estimates), and we deploy gated recurrent units (GRU) (Cho et al. 2014) layers for each feature. Twenty consecutive quarters form a sequence, and each sequence is the input to predict the next quarter for each feature. Ten GRU models are simultaneously trained and the hidden state outputs are combined and passed into FCN layers for prediction. The model architecture is illustrated in Figure 1. This multi-GRU model (with LightGBM assist), not only models time series, but reduces the dimensionality of the inputs and facilitates the discovery of hierarchical structure among hidden states.

Name	Feature Importance Rank
Stock_return_1qa	6.7
Sales_ts	8.3
Stock_return_3qb	8.4
Pretax_margin_ts01	10
Sales_ts13	11.9
Capex_to_dda	13.5
Fwd_roic	13.8
Eps_ts01	14.1
Ebitda_to_ev	14.1
Cfps_ts01	14.6
Eps_ts35	14.8
Ni_to_cfo	15
Ibes_qcut_as_x	15.4
Pretax_margin_ts13	15.8
Pretax_margin_ts35	16.1

Table 5: Feature Importance Rankings from LightGBM.

For these models of non-linear topology, multiple losses are usually required. For instance, the losses of each feature GRU is usually combined with the losses of the final FCN layer (the actual earnings forecast). But accuracy *improved* when we focused only on the final FCN losses. The GRU layers are being utilized for encoding the data, not prediction, to allow the final FCN layers to *learn* the hierarchal structure of the data. Our tests included other encoders such as CNN or transformer layers, but the GRU "encoders" yielded the best results. The final multi-GRU deep learning model yields higher accuracy (in both MAE and R^2) than the consensus benchmarks, albeit with help from the competition. This implies that lower dimensions and robust encoding are both required before application of deep learning models on noisy HDLSS financial data. Unfortunately, the resulting accuracy measures still fare worse and the multi-GRU model requires exponential more than computing resources. LightGBM models take on average 4 minutes to converge on an 8 core Intel i7 CPU while the GRU model requires 4 minutes *per epoch* to go through on an RTX 2080 GPU. Even assuming comparable results, gradient boosting trees would be highly preferred in actual application.

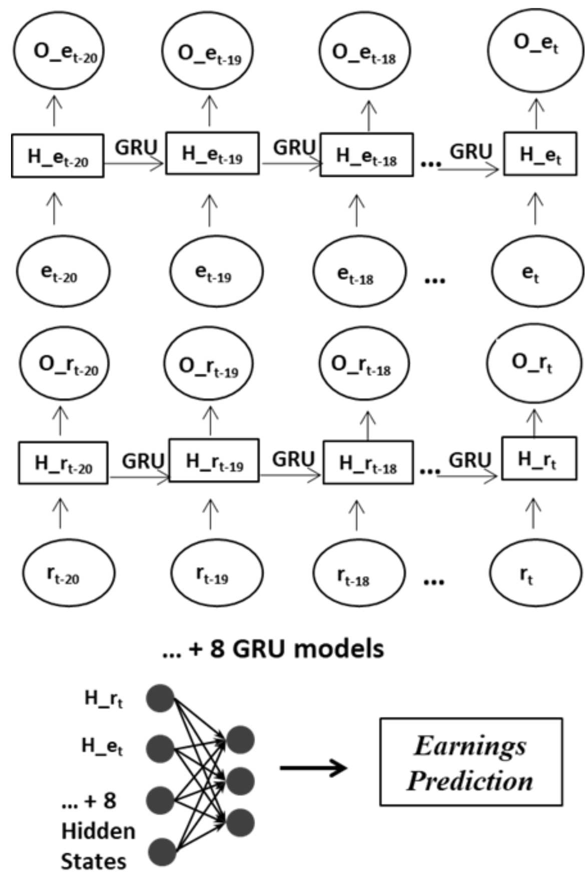

Figure 1: Configuration V, GRU model.

Conclusion

In this paper, we present evidence against the application of deep learning models for earnings forecasting. For such noisy HDLSS problems, we offer gradient boosting models as a preferable solution. However, this work does not represent an exhaustive study on deep learning models in aggregate. New developments in attention models (Vaswani et al. 2017) and temporal convolutional networks (TCN) (Lea et al. 2017) have shown significant improvements in time series modeling. Nonetheless, for problems such as company financials in which the data frequency is low and the sample size is limited, these complex models face problems with the requisite *large* number of model parameters. For homogeneous data, such as stock price data, in which the hierarchical structure is more obtainable (Mantegna 1999) and higher data frequencies produce greatly larger sample sizes (tick or daily), attention and TCN models can achieve success (Qiu, Wang, and Zhou 2020). But for other financial and economic data, for which the frequency is measured in months and quarters rather than minutes and days, deep learning might not be optimal.

In ongoing and future work, we plan to extend our analysis to other financial and economic data and broaden the scope to a larger family of deep learning and machine learning models.

References

Abarbanell, J. S.; and Bushee, B. J. 1997. Fundamental Analysis, Future Earnings, and Stock Prices. Journal of Accounting Research 35(1): 1–24. doi.org/10.2307/2491464.

Andreou, E.; Ghysels, E.; and Kourtellos, A. 2013. Should Macroeconomic Forecasters Use Daily Financial Data and How? Journal of Business & Economic Statistics 31(2): 240–251. doi.org/10.1080/07350015.2013.767199.

Ball, R. T.; and Ghysels, E. 2017. Automated Earnings Forecasts: Beat Analysts or Combine and Conquer? Management Science 64(10): 4936–4952. doi.org/10.1287/mnsc.2017.2864.

Bergstra, J.; Komer, B.; Eliasmith, C.; Yamins, D.; and Cox D. D. 2013. Hyperopt: A Python Library for Model Selection and Hyperparameter Optimization. Computational Science & Discovery 8(1): 014008. doi.org/10.1088/1749-4699/8/1/014008.

Breiman, L. 2001. Random Forests. Machine Learning 45: 5–32. doi.org/10.1023/A:1010933404324.

Caruana, R.; and Niculescu-Mizil, A. 2006. An Empirical Comparison of Supervised Learning Algorithms. In Proceedings of the 23rd International Conference on Machine Learning. doi.org/10.1145/1143844.1143865.

Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. doi.org/10.1145/2939672.2939785.

Cho, K.; van Merrienboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. In Proceedings of Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8) 103–111. doi.org/10.3115/v1/W14-4012.

Cybenko, G. 1988. Continuous Valued Neural Networks with Two Hidden Layers are Sufficient. Technical Report. University of Illinois at Urbana-Champaign.

Friedman, J. H. 2001. Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics 29(5): 1189–1232. doi:10.1214/aos/101320345.

Ghysels, E.; Santa-Clara, P; and Valkanov, R. 2002. The MIDAS Touch: Mixed Data Sampling Regression Models. Working paper, UNC and UCLA.

Kolanovic, M.; and Smith, R. 2019. Big Data and AI Strategies: 2019 Alternative Data Handbook. Global Quantitative and Derivatives Strategy Report October 2019, New York, NY: JP Morgan.

Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; and Liu, T. Y. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 3146–3154.

Kumar, N. 2019. Why Machine Learning Hasn't Made Investors Smarter. Bloomberg, July 2019.

Lapedes, A.; and Farber, R. 1987. How Neural Nets Work. In Proceedings of the 1987 International Conference on Neural Information Processing Systems 442–456. doi.org/10.1142/9789814434102_0012.

Lea, C.; Flynn, M.; Vidal, R.; Reiter, A.; and Hager, G. 2017. Temporal Convolutional Networks for Action Segmentation and Detection. IEEE Conference on Computer Vision and Pattern Recognition, CVPR. 2017. 113

Lev, B.; and Thiagarajan, S. R. 1993. Fundamental Information Analysis. Journal of Accounting Research 31(2): 190–215. doi.org/10.2307/2491270.

Longstaff, F. A.; and Piazzesi, M. 2004. Corporate Earnings and the Equity Premium. Journal of Financial Economics 74(3): 401–421.

Lopez de Prado, M. L. 2018. The 10 Reasons Most Machine Learning Funds Fail. Journal of Portfolio Management 44(6): 120–133. doi.org/10.2139/ssrn.3104816.

Mantegna, R. N. 1999. Hierarchical Structure in Financial Markets. The European Physical Journal B 11: 193–197. https://doi.org/10.1007/s100510050929.

Ou, J. A.; and Penman, S. H. 1989. Financial Statement Analysis and the Prediction of Stock Returns. Journal of Accounting and Economics 11: 295–329. doi.org/10.1016/0165-4101(89)90017-7.

Ozbayoglu, A. M.; Gudelek, M. U.; and Sezer, O. B. 2020. Deep Learning for Financial Applications: A Survey. Applied Soft Computing 93: 106384. doi.org/10.1016/j.asoc.2020.106384.

Poggio, T.; Banburski, A.; and Liao, Q. 2020. Theoretical Issues in Deep Networks. In Proceedings of the National Academy of Sciences. doi.org/10.1073/pnas.1907369117.

Qiu, J.; Wang, B.; and Zhou, C. 2020. Forecasting Stock Prices with Long-Short Term Memory Neural Network Based on Attention Mechanism. PLoS One 15(1): e0227222.

Valentine, J. J. 2011. Best Practices for Equity Research Analysts: Essentials for Buy-Side and Sell-Side Analysts. New York: McGraw-Hill Education

Vaswani, D.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17).

Wu, X.; Kumar, V.; Quinlan, J. R.; Ghosh, J.; Yang, Q.; Motoda, H.; McLachlan, G. J.; Ng, A.; Liu, B.; Philip, S. Y.; and Zhou, Z. H. 2008. Top 10 Algorithms in Data Mining. Knowledge and Information Systems 14(1): 1–37. doi.org/10.1007/s10115-007-0114-2.