Dynamic and Context-Dependent Stock Price Prediction Using Attention Modules and News Sentiment

Nicole Königstein (nicole.koenigstein@impactvise.com)

Research Article

Keywords: financial time series, deep learning, recurrent neural network, attention

Posted Date: January 6th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-2430807/v1

License: © This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Digital Finance on August 1st, 2023. See the published version at https://doi.org/10.1007/s42521-023-00089-7.
Dynamic and Context-Dependent Stock Price Prediction Using Attention Modules and News Sentiment

Nicole Königstein$^{1,1*}$

$^{1*}$impactvise AG, Zählerweg 5, Zug, 6300, Switzerland.

Corresponding author(s). E-mail(s):
nicole.koenigstein@impactvise.com;

Abstract

The growth of machine-readable data in finance, such as alternative data, requires new modeling techniques that can handle non-stationary and non-parametric data. Due to the underlying causal dependence and the size and complexity of the data, we propose a new modeling approach for financial time series data, the $\alpha_t$-RIM (recurrent independent mechanism). This architecture makes use of key-value attention to integrate top-down and bottom-up information in a context-dependent and dynamic way. To model the data in such a dynamic manner, the $\alpha_t$-RIM utilizes an exponentially smoothed recurrent neural network, which can model non-stationary time series data, combined with a modular and independent recurrent structure. We apply our approach to the closing prices of three selected stocks of the S&P 500 universe as well as their news sentiment score. The results suggest that the $\alpha_t$-RIM is capable of reflecting the causal structure between stock prices and news sentiment, as well as the seasonality and trends. Consequently, this modeling approach markedly improves the generalization performance, that is, the prediction of unseen data, and outperforms state-of-the-art networks such as long short-term memory models.

Keywords: financial time series, deep learning, recurrent neural network, attention

$^1$The author would like to thank YUKKA Lab, Berlin, for providing the raw data for this research. Furthermore, the author would like to thank Matthew Dixon and Saeed Amen, who provided significant support to the research with their insights and expertise. Finally, the author would like to thank Jörg Osterrieder for his comments and suggestions on this paper.
1 Introduction

Non-stationary time series data is common in finance. In the case of price series, each value depends on a long history of prior levels. However, most machine learning models that predict financial time series expect stationary inputs. Consequently, these models rely on standard stationarity transformations, such as integer differentiation, to produce returns, which have a memory cut-off hence the series loses important signals (Prado, 2018), (Sugiyama & Kawanabe, 2012). In addition, in the field of causal inference, the concept of independent or autonomous processes has proven to be important because these processes make the model capable of cause and effect inference (Pearl, 2009), (Bengio, 2017). Thus, a complex model may be thought of as a collection of separate processes or “causal” modules. As a result, individual modules may be robust or invariant even when other modules change, as in a distribution shift (Peters, Janzing, & Schölkopf, 2017), (Schoelkopf et al., 2012). Furthermore, most machine learning modules are monolithic and based on bottom-up signals, i.e. directly observed content, in contrast to a top-down signal, which is based on past experience and short-term memory. Moreover, human cognition has a modular structure with sparse interactions, as described by Carruthers in his book The Architecture of the Mind (Carruthers, 2006). Carruthers argues that one of the distinctive characteristics of the human mind is that it is composed of numerous cognitive systems, each of which communicates with a small number of other systems or experts, each of which has little influence over the processes occurring within them. Thus, the human mind is flexible and capable of practical reasoning and thus gains the capacity for scientific thinking. Consequently, if we think of the brain as capable of solving problems by using different systems (or modules), we hypothesize that it could be beneficial to leverage this kind of structure by learning separate processes that can be reused, constructed, and flexibly re-purposed. Humans also seldom utilize all available inputs to complete tasks. For these reasons, using sparse interactions and focusing attention in machine learning models may reduce learning difficulties by minimizing interference. In other words, models that learn in this manner may more accurately reflect the causal structure of the data and hence generalize more effectively (Simon, 1991), (Peters et al., 2017). Accordingly, within this context, we analyze how such a modeling approach can be used to incorporate stock prices and news sentiment to predict stock price movements.

2 Related Work

Prior to the use of deep learning within the field of financial times series prediction, methods such as ARIMA and its modifications were mainly used. For instance, Minyoung Kim replaced the conventional maximum likelihood estimation with an asymmetric loss function owing to the asymmetric distribution of financial time series returns (M. Kim, 2015), and Adebiyi et al. evaluated the ARIMA and artificial neural network prediction performance on NYSE stocks (e. g., stock price of Dell Incorporation) (Adebiyi, Adewumi,
The empirical findings of these studies demonstrated the superiority of neural networks over ARIMA models. Recent advances in deep learning have resulted in enhanced model performance, resulting from the increased computing capacity and the ability to understand non-linear connections embedded in a variety of financial variables.

Sreelekshmy Selvin et al. examined three distinct deep learning architectures for stock prediction, including recurrent neural network (RNN), long short-term memory (LSTM), and convolutional neural network (CNN)-sliding window models for the prediction of Nation Stock Exchange of India listed stocks (Selvin, Vinayakumar, Gopalakrishnan, Menon, & Soman, 2017). Within their study, they concluded that the CNN outperforms the RNN and LSTM because it is capable of identifying trend changes in the stocks. Further, Panday et al. evaluated a hybrid algorithm, which uses an LSTM to predict stock values and a sentiment analysis to verify the predictions (Harsh Panday, 2020). Their findings indicated that there is, in fact, a relationship between public sentiment and stock prices. Moreover, Matthew Dixon has introduced a new class of smoothed RNNs, the $\alpha$-RNN and the $\alpha_t$-RNN (Dixon, 2020), and shown that they are superior and more robust than simple RNNs and ARIMA models for industrial forecasting. Recently, the concept of attention has also acquired considerable importance in the area of artificial intelligence as an essential component of neural networks for a large number of Computer Vision applications (Cho, Courville, & Bengio, 2015) (Wang, Huang, Zhu, & Zhao, 2016), speech recognition, natural language processing (Galassi, Lippi, & Torroni, 2020) and, more recently, for financial time series prediction (S. Kim & Kang, 2019) (Zhang et al., 2019). In those cases, the experimental findings demonstrated that the networks with attention outperformed models such as ARIMA and LSTM without attention in financial time series prediction.

3 Proposed Model

In this section, we outline our proposed approach, which relies on two deep learning architectures, namely the recurrent independent mechanism (Goyal et al., 2020) and a general class of exponentially smoothed RNNs, the $\alpha$-RNN and the $\alpha_t$-RNN (Dixon, 2020). More specifically, we use the traditional RIM architecture, where the modules are based upon an $\alpha_t$-RNN cell as a recurrent model, as opposed to an LSTM cell.

3.1 Recurrent Independent Mechanism

The RIM is a recurrent architecture with multiple modules of recurrent cells, which operate with independent transition dynamics (Henaff, Weston, Szlam, Bordes, & LeCun, 2016), (Kipf, Fetaya, Wang, Welling, & Zemel, 2018). They communicate sparsely via attention and compete for the most relevant input for each time step. In RIMs, the updates for the hidden state follow a three-step process:
1. A subset of modules is selectively activated based on the determination of the relevance of the modules input.
2. The activated modules independently process the information made available to them.
3. The RIMs communicate with one another sparingly via key-value attention, then the active modules gather contextual information from all the other modules and consolidate this information in their hidden state.

The entire model is subdivided into $k$ small subsystems, so-called RIMs, in which each represents a recurrent model to capture the dynamics in observable sequences. As a consequence, each RIM has its own unique functions that are automatically trained from data, resulting in the vector-valued state $h_{t,k}$ at timestep $t$. Each RIM also contains $\theta_k$ parameters that are shared between time steps.

However, only when the input is relevant are the RIM modules active and updated. For each time step, the attention selects and then activates a subset of the RIMs. Soft attention takes the product of a query represented as a matrix of dimension $N_r \times d$, with $d$ being the dimension of each key, and a collection of $N_0$ objects, each associated with a key as a row in matrix $K^T(N_0 \times d)$, and after normalizing (using Softmax) produces the following outputs:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d}} \right) V,$$

where $Q$, $K$ and $V$ are the query, key and value matrices, respectively. The Softmax algorithm is then performed on each row of the argument matrix, resulting in a set of convex weights. As a consequence, a convex combination of the values in the rows of $V$ is obtained. Note that when the attention is focused on one element of a specific row, which means that the softmax is saturated, it selects one of the objects and copies its value to row $j$ of the result. Further, the $d$ dimensions of the keys may be divided into heads, each of which has its own attention matrix and writes independently calculated values.

Without attention, neurons in neural networks operate on fixed variables and are thus fed by the previous layer. The key-value attention mechanism enables a dynamic selection of which a variable instance will be used as the input for each of the dynamics of the RIM’s arguments. These inputs may originate from an external source or be the output of another RIM. Thus, the model learns to dynamically choose those RIMs that are relevant to the present input. The RIMs provide the queries for this specific application of key-value attention, while the current input provides the keys and values. The following summarizes the input attention paid to a particular RIM:

At time $t$, the input $x_t$ is seen as a collection of rows of a matrix, followed by the concatenation of a row of zeros to obtain the following equation:
Following this, linear transformations are applied to generate keys \((K = XW^e, \text{one for each input element and the null element})\), values \((V = XW^v, \text{one for each element})\), and queries \((Q = h_tW^q_k, \text{one for each RIM attention head})\). \(W^v\) is a simple matrix translating an input element to the weighted attention’s associated value vector, and \(W^e\) is similarly a weight matrix mapping the input to the keys. \(W^q_k\) is a weight matrix for each RIM that relates the RIM’s hidden state to its queries. \(\oplus\) refers to the concatenation operator at the row level. Thus, the attention can be written as follows:

\[
A_k^{(in)} = \text{softmax} \left( \frac{h_tW^q_k (XW^e)^T}{\sqrt{d_s}} \right) XW^v, \text{where } \varrho_k^{(in)} = (W^q_k, W^e, W^v)
\]

The RIMs use multiple heads for the input and communication attention, analogously to “Attention is all you need” (Vaswani et al., 2017). In general, the RIMs operate independently by default and the attention mechanism allows the model to share information among the modules. Furthermore, the activated RIMs are allowed to read from all the other RIMs’ inputs. The reason for this is that non-activated RIMs do not need to change their value because they are not related to the current input. Nonetheless, they may retain important contextual information, therefore, there is communication between the RIMs through the usage of residual connections, as described in the paper “Relational recurrent neural networks” (Santoro et al., 2018). Figure 1 illustrates the described RIM dynamics. The illustrations were adapted from the original paper and partially modified to fit the use case of this work.

**Fig. 1** Illustration of recurrent independent mechanisms.
However, the RIMs were originally used with gated recurrent units (Chung, Gulcehre, Cho, & Bengio, 2014) and LSTM (Hochreiter & Schmidhuber, 1997), which we decided against, because the $\alpha$-RNN and $\alpha_t$-RNN are being a much simpler architecture, yet perfectly suitable for forecasting stationary and non-stationary time series, respectively.

4 $\alpha$-RNNs and $\alpha_t$-RNNs

The $\alpha$-RNNs are a generic family of exponentially smoothed RNNs that excel in modeling non-stationary time series data as seen in financial applications. They characterize the time series’ non-linear partial autocorrelation structure and directly capture dynamic influences such as seasonality and trends. The $\alpha$-RNN is almost identical to a standard RNN except for the addition of a scalar smoothing parameter, which provides the recurrent network with extended memory, that is, autoregressive memory beyond the sequence length.

To extend RNNs into dynamic time series models, the combination of the hidden state $\hat{h}_t$ and the exponentially smoothed output $\tilde{h}_t$, which is time-dependent and convex, is used. This combination means that the model is capable of modeling non-stationary time series data, as in the following equation:

$$\tilde{h}_{t+1} = \alpha_t \hat{h}_t + (1 - \alpha_t) \tilde{h}_t$$  \hspace{1cm} (4)

Thus, smoothing may be thought of as a type of dynamic forecast error correction. Alternatively, smoothing may be viewed as a weighted summation of the lagged data with either equal or lower weights, $\alpha_{t-s} \prod_{r=1}^{t-s} (1 - \alpha_{t-r+1})$ at the $s \geq 1$ lagged hidden state, $\hat{h}_{t-s}$:

$$\tilde{h}_{t+1} = \alpha_t \hat{h}_t + \sum_{s=1}^{t-1} \alpha_{t-s} \prod_{r=1}^{s} (1 - \alpha_{t-r+1}) \hat{h}_{t-s} + g(\alpha),$$  \hspace{1cm} (5)

where $g(\alpha) := \prod_{r=0}^{t-1} (1 - \alpha_{t-r}) \tilde{y}_1$. Note that beyond the $r^{th}$ lag, for any $\alpha_{t-r+1} = 1$, the model will forget the hidden states.

While the $\alpha_t$-RNN is free to specify how $\alpha$ is updated (including altering the update frequency) in response to the hidden state and input, using a recurrent layer is a convenient choice.

5 $\alpha_t$-RIM

The activated RIMs in the $\alpha_t$-RIM uses the $\alpha_t$-RNN as their per-RIM independent transition dynamics. This choice was made because for industrial forecasting, LSTMs and GRUs are likely over-engineered, and light-weight exponentially smoothed architectures capture the key properties while being superior and more robust than simple RNNs.
6 Experiments and Results

We evaluated our modeling approach using financial time series data with and without news sentiment, hereafter denoted as univariate and bivariate, respectively. To further study our model’s performance, we compared it with two RNN models, namely a simple RNN and an LSTM. TensorFlow v2.4.1 (Abadi et al., 2016) was used to implement all the models, while the $\alpha_t$-RIM was a custom implementation and the SimpleRNN and the LSTM from the TensorFlow-Keras-API were used. Further, time series cross-validation was performed using separate training, validation, and test sets for all the models and stocks. Each set represents a contiguous sample period, with the test set containing the most recent observations in order to maintain the data’s temporal structure and prevent look-ahead bias. The hidden layer was activated using Tanh functions. The Glorot and Bengio uniform (Glorot & Bengio, 2010) approach was used to initialize the non-recurrent weight matrices, and an orthogonal matrix was used to initialize the recurrence weights for stability, which ensures that the absolute value of the eigenvalues is initially limited by unity (Henaff, Szlam, & LeCun, 2016). Further, Adam (Kingma & Ba, 2017) is used as optimizer. The hyperparameters for the RNN and LSTM were determined using time series cross-validation with five folds. The hidden units were evaluated starting from 5 up to and including 250 in steps of 5. L1 regularization was 0.0001, 0.001, 0.01, 0.10 and the dropout rate was 0.1, 0.2, 0.3, 0.4, 0.5, 0.6. The setup for the $\alpha_t$-RIM was identical, with the exception that the hyperparameters were determined with a three fold randomized grid search due to the 12 hyperparameters of the model. See Appendix Subsection 6.1 for more information on the model’s hyperparameters. Within that setup, we used three metrics: mean squared error (MSE), mean absolute error (MAE), and mean percentage error (MAPE). The first two were used to evaluate the models during training, validation, and testing to examine their generalization performance on unseen data. The MAPE was used after training on the re-scaled testing dataset, where re-scaled means that the data transformations
were reverted in order to retrieve the original closing prices, and separately for each prediction time step ahead, as our prediction horizon was five days ahead.

6.1 Data

Two data sources were used for the analyses: end of day pricing data from quandl.com and sentiment data from 2015 onwards, provided by YUKKA Lab, Berlin; the data is available via the research repository (Königstein, 2021). The two datasets were then joined and contained 6.5 years of complete data from within the S&P 500 Index. To account for variability in the experiments, trading volume, the total article count, and the number of positive and negative articles of all stocks were analyzed, resulting in the following chosen stocks:

1. **Amazon**: With high dollar volume and high article count.
2. **Brown-Forman**: With low dollar volume and low article count.
3. **Thermo Fisher**: With medium dollar volume and medium-high article count.

To account for the noise in the sentiment data, we smoothed the sentiment score with a convolutional kernel filter (Pourahmadi, 2016), (Drury, 2017) during pre-processing. The datasets were then divided into training, validation, and testing sets. The training set was standardized using the mean and standard deviation of the training set and not the whole time series. Additionally, to avoid introducing systematic bias into the validation and test sets, identical normalization was used for the validation and test sets. In other words, the mean and standard deviation of the training set were used to normalize the validation and test sets. Further, log transformations were performed to decrease the data variability and bring the data closer to the normal distribution. Moreover, due to the strong downward impact of the COVID-19 pandemic on stock prices, the choice was made to remove the observation with the steep downward trend in the chosen stocks, as we expected the models to perform poorly during this phase. This resulted in the following divisions:

1. **Training set**: 2015-01-05 – 2019-12-31
2. **Validation set**: 2020-05-04 – 2020-12-28
3. **Testing set**: 2020-12-29 – 2021-06-08

Moreover, we analyzed three different look-back time windows, 5, 10, and 21 days, to study the short- and long-term effects of the generalization related to the networks and time windows.

6.2 Results

After comparing the results for all the networks, stocks, and time windows, we observed that the results depicted the same pattern. The performance of the training data sets differed little, that is, the MSE and MAE were almost identical during the training phase of the three different networks, namely the simple RNN, LSTM, and $\alpha_t$-RIM, but the values of the metrics increased
steadily from the validation to the testing sets, for the simple RNN and LSTM.

Accounting for the aforementioned results and the results from the comparison with each prediction step ahead led to the following two observations. Our model outperforms the simple RNN and the LSTM for each prediction time step and during the validation and testing phases of the networks. The Figures 3-8 depict the results from the 10 day look-back and 5 day ahead prediction from the model evaluation and clearly demonstrate how the RNN and LSTM fail to generalize for the validation and testing sets. Furthermore, the $\alpha_t$-RIM is capable of using the sentiment score as an additional feature and improves its overall performance with the use of news sentiment. Regarding the input lags needed for the 5 days ahead prediction, the 10 lags resulted in the best overall performance for all tested networks, with two exceptions: the $\alpha_t$-RIM performance was slightly better with 21 lags input for Brown Forman stock, and the LSTM performed also better with 21 input lags, but for Thermo Fischer stock. A comparison of the results for Brown Forman stock with 10 input lags is presented in Tables 1 and 2 with the best values in bold.

**Fig. 3** Univariate evaluation metrics of 10 lags input for Amazon stock.

**Fig. 4** Bivariate evaluation metrics of 10 lags input for Amazon stock.

**Fig. 5** Univariate evaluation metrics of 10 lags input for Brown Forman stock.

**Fig. 6** Bivariate evaluation metrics of 10 lags input for Brown Forman stock.
This work demonstrates the effectiveness of combining an RIM with an exponentially smoothed RNN for modeling non-stationary time series data. As our findings indicate, the $\alpha_t$-RIM outperforms the simple RNN and the LSTM, particularly within the validation and testing phases. This observation leads to the inference that the simple RNN and LSTM are not able to model changes in the unseen data. This demonstrates that the $\alpha_t$-RIM can generalize effectively on unseen data. Additionally, the bivariate time series prediction results suggest that the $\alpha_t$-RIM is capable of extracting patterns from news sentiment and using them as an additional input within its communication pattern to further stabilize the prediction accuracy. This result substantiates our earlier statement that it is necessary to utilize a new modeling approach that takes advantage of key-value attention, an exponentially smoothed RNN, and subsystems to model financial time series data in a dynamic and context-dependent way. Furthermore, in addition to the more accurate predictions that it provided, the advantage of the $\alpha_t$-RIM is that the attention weights and activation pattern of the modules can be visualized.
to further study the behavior of the model. In closing, for further research, we suggest investigating the possibility of including DSelect-k (Hazimeh et al., 2021) for the $\alpha_t$-RIM, which would the model allow to learn to select a different number of modules at each time-step.

8 Statements and Declarations

No funding was received for conducting this study. The author has no competing interests to declare that are relevant to the content of this article.

Appendix A

A.1 The $\alpha_t$-RIM Hyper-Parameter

Due to the models’ constraints of the hyper-parameters (e.g., the number of RIMs have to be smaller or equal to $k$ modules), the normal cross-validation could not be performed. Therefore, a special function was implemented to generate a list of dictionaries to be fed into the grid search as a parameter grid. The list encompasses the following parameters:

- **Units**: 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 25, 30, 35, 40, 45, 50
- **Number of RIMs**: 4, 6, 8, 10, 12, 14
- **K modules**: 4, 6, 8, 10, 12, 14
- **Input key size**: 4, 6, 8, 10, 12
- **Input value size**: 4, 6, 8, 10, 12
- **Input query size**: 4, 6, 8, 10, 12
- **Input keep probability**: 0.6, 0.7, 0.8, 0.9
- **Number of communication heads**: 2, 4, 6, 8
- **Communication key size**: 4, 6, 8, 10, 12
- **Communication value size**: 4, 6, 8, 10, 12
- **Communication query size**: 4, 6, 8, 10, 12
- **Communication keep probability**: 0.6, 0.7, 0.8, 0.9
## A.2 Complete Training Results

### A.2.1 Evaluation Metrics

**AMAZON**

| Table A1 | Univariate 5 input lags |
|------------|-------------------------|
| **Model** | **MSE** | **MAE** |
| RNN train | 0.0043 | 0.0476 |
| RNN val | 0.0164 | 0.1123 |
| RNN test | 0.0158 | 0.1158 |
| LSTM train | 0.0049 | 0.0508 |
| LSTM val | 0.0778 | 0.2683 |
| LSTM test | 0.0916 | 0.2981 |
| $\alpha_t$-RIM train | 0.0045 | 0.0484 |
| $\alpha_t$-RIM val | 0.0048 | 0.0536 |
| $\alpha_t$-RIM test | 0.0028 | 0.0416 |

Univariate evaluation metrics of 5 lags input for the Amazon stock.

| Table A2 | Bivariate 5 input lags |
|------------|-------------------------|
| **Model** | **MSE** | **MAE** |
| RNN train | 0.0065 | 0.0610 |
| RNN val | 0.0404 | 0.1894 |
| RNN test | 0.0433 | 0.2013 |
| LSTM train | 0.0049 | 0.0516 |
| LSTM val | 0.1044 | 0.3133 |
| LSTM test | 0.1259 | 0.3504 |
| $\alpha_t$-RIM train | 0.0055 | 0.0535 |
| $\alpha_t$-RIM val | 0.0050 | 0.0547 |
| $\alpha_t$-RIM test | 0.0027 | 0.0406 |

Bivariate evaluation metrics of 5 lags input for the Amazon stock.

| Table A3 | Univariate 10 input lags |
|------------|-------------------------|
| **Model** | **MSE** | **MAE** |
| RNN train | 0.0065 | 0.0620 |
| RNN val | 0.0314 | 0.1643 |
| RNN test | 0.0317 | 0.1704 |
| LSTM train | 0.0055 | 0.0538 |
| LSTM val | 0.0668 | 0.2463 |
| LSTM test | 0.0778 | 0.2703 |
| $\alpha_t$-RIM train | 0.0042 | 0.0451 |
| $\alpha_t$-RIM val | 0.0059 | 0.0601 |
| $\alpha_t$-RIM test | 0.0059 | 0.0495 |

Univariate evaluation metrics of 10 lags input for the Amazon stock.

| Table A4 | Bivariate 10 input lags |
|------------|-------------------------|
| **Model** | **MSE** | **MAE** |
| RNN train | 0.0048 | 0.0504 |
| RNN val | 0.0353 | 0.1738 |
| RNN test | 0.0394 | 0.1903 |
| LSTM train | 0.005 | 0.0522 |
| LSTM val | 0.0791 | 0.2703 |
| LSTM test | 0.0939 | 0.3 |
| $\alpha_t$-RIM train | 0.0043 | 0.0468 |
| $\alpha_t$-RIM val | 0.0053 | 0.0565 |
| $\alpha_t$-RIM test | 0.0030 | 0.0424 |

Bivariate evaluation metrics of 10 lags input for the Amazon stock.

| Table A5 | Univariate 21 input lags |
|------------|-------------------------|
| **Model** | **MSE** | **MAE** |
| RNN train | 0.0048 | 0.0503 |
| RNN val | 0.0148 | 0.1044 |
| RNN test | 0.0129 | 0.1066 |
| LSTM train | 0.0062 | 0.0590 |
| LSTM val | 0.0969 | 0.3011 |
| LSTM test | 0.1038 | 0.3141 |
| $\alpha_t$-RIM train | 0.0040 | 0.0444 |
| $\alpha_t$-RIM val | 0.0086 | 0.0749 |
| $\alpha_t$-RIM test | 0.0058 | 0.0627 |

Univariate evaluation metrics of 21 lags input for the Amazon stock.

| Table A6 | Bivariate 21 input lags |
|------------|-------------------------|
| **Model** | **MSE** | **MAE** |
| RNN train | 0.0044 | 0.0474 |
| RNN val | 0.0125 | 0.0948 |
| RNN test | 0.0107 | 0.0917 |
| LSTM train | 0.0059 | 0.0590 |
| LSTM val | 0.1893 | 0.4250 |
| LSTM test | 0.2156 | 0.4562 |
| $\alpha_t$-RIM train | 0.0038 | 0.0441 |
| $\alpha_t$-RIM val | 0.0061 | 0.0612 |
| $\alpha_t$-RIM test | 0.0028 | 0.0415 |

Bivariate evaluation metrics of 21 lags input for the Amazon stock.
### Table A7: Univariate 5 input lags

| Model         | MSE   | MAE  |
|---------------|-------|------|
| RNN train     | 0.0156| 0.091|
| RNN val       | 0.0345| 0.1467|
| RNN test      | 0.0289| 0.1394|
| LSTM train    | 0.0166| 0.0950|
| LSTM val      | 0.0550| 0.1897|
| LSTM test     | 0.0518| 0.1975|
| $\alpha_t$-RIM train | 0.014 | 0.0865 |
| $\alpha_t$-RIM val | 0.0225 | 0.1146 |
| $\alpha_t$-RIM test | 0.0164 | 0.0967 |

Univariate evaluation metrics of 5 lags input for the Brown Forman stock.

### Table A8: Bivariate 5 input lags

| Model         | MSE   | MAE  |
|---------------|-------|------|
| RNN train     | 0.0153| 0.0914|
| RNN val       | 0.0392| 0.1571|
| RNN test      | 0.0435| 0.1804|
| LSTM train    | 0.0177| 0.1011|
| LSTM val      | 0.1001| 0.2575|
| LSTM test     | 0.1720| 0.3885|
| $\alpha_t$-RIM train | 0.0145 | 0.0901 |
| $\alpha_t$-RIM val | 0.0207 | 0.1096 |
| $\alpha_t$-RIM test | 0.0153 | 0.1002 |

Bivariate evaluation metrics of 5 lags input for the Brown Forman stock.

### Table A9: Univariate 10 input lags

| Model         | MSE   | MAE  |
|---------------|-------|------|
| RNN train     | 0.0146| 0.0887|
| RNN val       | 0.0289| 0.1334|
| RNN test      | 0.0219| 0.1234|
| LSTM train    | 0.0214| 0.1114|
| LSTM val      | 0.1048| 0.2682|
| LSTM test     | 0.1090| 0.2929|
| $\alpha_t$-RIM train | 0.0145 | 0.0877 |
| $\alpha_t$-RIM val | 0.0254 | 0.1237 |
| $\alpha_t$-RIM test | 0.0171 | 0.0983 |

Univariate evaluation metrics of 10 lags input for the Brown Forman stock.

### Table A10: Bivariate 10 input lags

| Model         | MSE   | MAE  |
|---------------|-------|------|
| RNN train     | 0.015 | 0.0912|
| RNN val       | 0.0570| 0.1957|
| RNN test      | 0.0685| 0.2333|
| LSTM train    | 0.0179| 0.1026|
| LSTM val      | 0.1411| 0.3087|
| LSTM test     | 0.2609| 0.4876|
| $\alpha_t$-RIM train | 0.0152 | 0.0925 |
| $\alpha_t$-RIM val | 0.0248 | 0.1228 |
| $\alpha_t$-RIM test | 0.0151 | 0.0962 |

Bivariate evaluation metrics of 10 lags input for the Brown Forman stock.

### Table A11: Univariate 21 input lags

| Model         | MSE   | MAE  |
|---------------|-------|------|
| RNN train     | 0.0166| 0.0946|
| RNN val       | 0.0561| 0.1924|
| RNN test      | 0.0556| 0.2054|
| LSTM train    | 0.0247| 0.1210|
| LSTM val      | 0.1739| 0.3700|
| LSTM test     | 0.1872| 0.4011|
| $\alpha_t$-RIM train | 0.0136 | 0.0843 |
| $\alpha_t$-RIM val | 0.0208 | 0.1091 |
| $\alpha_t$-RIM test | 0.0130 | 0.0884 |

Univariate evaluation metrics of 21 lags input for the Brown Forman stock.

### Table A12: Bivariate 21 input lags

| Model         | MSE   | MAE  |
|---------------|-------|------|
| RNN train     | 0.0161| 0.0942|
| RNN val       | 0.0707| 0.2219|
| RNN test      | 0.0846| 0.2624|
| LSTM train    | 0.0254| 0.1253|
| LSTM val      | 0.2590| 0.4507|
| LSTM test     | 0.3775| 0.5908|
| $\alpha_t$-RIM train | 0.0136 | 0.0854 |
| $\alpha_t$-RIM val | 0.0237 | 0.1194 |
| $\alpha_t$-RIM test | 0.0187 | 0.1159 |

Bivariate evaluation metrics of 21 lags input for the Brown Forman stock.
## THERMO FISCHER

### Table A13 Univariate 5 input lags

| Model        | MSE   | MAE   |
|--------------|-------|-------|
| RNN train    | 0.0076| 0.0669|
| RNN val      | 0.066 | 0.2272|
| RNN test     | 0.1031| 0.3013|
| LSTM train   | 0.0089| 0.0737|
| LSTM val     | 0.1200| 0.3146|
| LSTM test    | 0.2002| 0.4345|
| $\alpha_t$-RIM train | 0.0086| 0.0727|
| $\alpha_t$-RIM val  | 0.0106| 0.0797|
| $\alpha_t$-RIM test | 0.0138| 0.0895|

Univariate evaluation metrics of 5 lags input for the Thermo Fischer stock.

### Table A14 Bivariate 5 input lags

| Model        | MSE   | MAE   |
|--------------|-------|-------|
| RNN train    | 0.0073| 0.0640|
| RNN val      | 0.2113| 0.4069|
| RNN test     | 0.2637| 0.4837|
| LSTM train   | 0.0091| 0.0746|
| LSTM val     | 0.4518| 0.6262|
| LSTM test    | 0.6633| 0.8007|
| $\alpha_t$-RIM train | 0.0079| 0.0685|
| $\alpha_t$-RIM val  | 0.0125| 0.0889|
| $\alpha_t$-RIM test | 0.0108| 0.0803|

Bivariate evaluation metrics of 5 lags input for the Thermo Fischer stock.

### Table A15 Univariate 10 input lags

| Model        | MSE   | MAE   |
|--------------|-------|-------|
| RNN train    | 0.0073| 0.0638|
| RNN val      | 0.0255| 0.1321|
| RNN test     | 0.0307| 0.1526|
| LSTM train   | 0.0098| 0.0775|
| LSTM val     | 0.1107| 0.3025|
| LSTM test    | 0.1602| 0.3932|
| $\alpha_t$-RIM train | 0.0073| 0.0648|
| $\alpha_t$-RIM val  | 0.0608| 0.2194|
| $\alpha_t$-RIM test | 0.0894| 0.2830|

Univariate evaluation metrics of 10 lags input for the Thermo Fischer stock.

### Table A16 Bivariate 10 input lags

| Model        | MSE   | MAE   |
|--------------|-------|-------|
| RNN train    | 0.0075| 0.0656|
| RNN val      | 0.2912| 0.4925|
| RNN test     | 0.3599| 0.5755|
| LSTM train   | 0.0126| 0.0866|
| LSTM val     | 0.3855| 0.5720|
| LSTM test    | 0.5568| 0.7312|
| $\alpha_t$-RIM train | 0.0077| 0.0656|
| $\alpha_t$-RIM val  | 0.0097| 0.0757|
| $\alpha_t$-RIM test | 0.0090| 0.0764|

Bivariate evaluation metrics of 10 lags input for the Thermo Fischer stock.

### Table A17 Univariate 21 input lags

| Model        | MSE   | MAE   |
|--------------|-------|-------|
| RNN train    | 0.0076| 0.0653|
| RNN val      | 0.0461| 0.1860|
| RNN test     | 0.0563| 0.2120|
| LSTM train   | 0.0118| 0.0818|
| LSTM val     | 0.0901| 0.2643|
| LSTM test    | 0.1290| 0.3421|
| $\alpha_t$-RIM train | 0.0079| 0.0689|
| $\alpha_t$-RIM val  | 0.0195| 0.1179|
| $\alpha_t$-RIM test | 0.0131| 0.0944|

Univariate evaluation metrics of 21 lags input for the Thermo Fischer stock.

### Table A18 Bivariate 21 input lags

| Model        | MSE   | MAE   |
|--------------|-------|-------|
| RNN train    | 0.0074| 0.0647|
| RNN val      | 0.192 | 0.3984|
| RNN test     | 0.2637| 0.4837|
| LSTM train   | 0.1816| 0.4037|
| LSTM val     | 0.4833| 0.6598|
| LSTM test    | 0.5983| 0.7646|
| $\alpha_t$-RIM train | 0.0065| 0.0608|
| $\alpha_t$-RIM val  | 0.0146| 0.0884|
| $\alpha_t$-RIM test | 0.0176| 0.1028|

Bivariate evaluation metrics of 21 lags input for the Thermo Fischer stock.
### A.2.2 Re-Scaled Metrics

**AMAZON**

#### Table A19 Univariate 5 input lags

| Lag | RNN MAPE | LSTM MAPE | α_t-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 6.578    | 14.9147   | **1.7589**   |
| 2   | 5.0371   | 14.3504   | 1.9456       |
| 3   | 6.7379   | 14.7175   | 2.1742       |
| 4   | 5.9735   | 14.0910   | 2.4410       |
| 5   | 5.2029   | 14.646    | 2.6204       |

Re-scaled univariate metrics of 5 input lags for the Amazon stock.

#### Table A20 Bivariate 5 input lags

| Lag | RNN MAPE | LSTM MAPE | α_t-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 9.8698   | 15.6884   | **1.6553**   |
| 2   | 8.5604   | 16.8496   | 1.9656       |
| 3   | 11.682   | 18.0337   | 2.1480       |
| 4   | 10.6241  | 16.8135   | 2.3762       |
| 5   | 9.6089   | 17.0085   | 2.5653       |

Re-scaled bivariate metrics of 5 input lags for the Amazon stock.

#### Table A21 Univariate 10 input lags

| Lag | RNN MAPE | LSTM MAPE | α_t-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 9.1246   | 11.4027   | **1.6793**   |
| 2   | 8.4374   | 13.8804   | 2.0933       |
| 3   | 9.0449   | 13.4336   | 2.5583       |
| 4   | 8.3960   | 15.2850   | 3.1922       |
| 5   | 7.8868   | 12.3275   | 3.3351       |

Re-scaled univariate metrics of 10 input lags for the Amazon stock.

#### Table A22 Bivariate 10 input lags

| Lag | RNN MAPE | LSTM MAPE | α_t-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 8.0755   | 12.3300   | **1.5730**   |
| 2   | 8.9933   | 16.0122   | 1.9777       |
| 3   | 11.4880  | 15.8318   | 2.2137       |
| 4   | 8.4545   | 15.0134   | 2.6374       |
| 5   | 10.6264  | 13.8912   | 2.7156       |

Re-scaled bivariate metrics of 10 input lags for the Amazon stock.

#### Table A23 Univariate 21 input lags

| Lag | RNN MAPE | LSTM MAPE | α_t-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 6.6462   | 12.1883   | **2.0397**   |
| 2   | 3.9523   | 16.8744   | 2.8626       |
| 3   | 5.4141   | 17.4566   | 2.9411       |
| 4   | 4.7591   | 15.8323   | 3.8771       |
| 5   | 4.9461   | 13.8310   | 4.4702       |

Re-scaled univariate metrics of 21 input lags for the Amazon stock.

#### Table A24 Bivariate 21 input lags

| Lag | RNN MAPE | LSTM MAPE | α_t-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 4.4279   | 16.6017   | **1.4672**   |
| 2   | 3.7260   | 22.7144   | 1.9301       |
| 3   | 6.3038   | 24.3148   | 2.1216       |
| 4   | 4.2521   | 21.8113   | 2.4899       |
| 5   | 4.7856   | 21.2809   | 2.8684       |

Re-scaled bivariate metrics of 21 input lags for the Amazon stock.
## BROWN FORMAN

### Table A25 Univariate 5 input lags

| Lag | RNN MAPE | LSTM MAPE | α₁-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 4.5914   | 3.5285    | 1.4819       |
| 2   | 1.8316   | 3.6831    | 1.7094       |
| 3   | 2.0918   | 3.9442    | 2.0129       |
| 4   | 2.1048   | 3.7976    | 2.2163       |
| 5   | 3.2806   | 4.5023    | 2.4563       |

### Table A26 Re-scaled univariate metrics of 5 input lags for the Brown Forman stock.

### Table A27 Bivariate 5 input lags

| Lag | RNN MAPE | LSTM MAPE | α₁-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 2.7613   | 6.5932    | 1.3910       |
| 2   | 3.0322   | 7.6978    | 1.7784       |
| 3   | 4.0114   | 8.1611    | 1.9711       |
| 4   | 3.6145   | 7.6531    | 2.2752       |
| 5   | 4.3920   | 7.4530    | 2.6458       |

### Table A28 Univariate 10 input lags

| Lag | RNN MAPE | LSTM MAPE | α₁-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 2.3279   | 5.2203    | 1.7081       |
| 2   | 1.9341   | 5.3022    | 1.7780       |
| 3   | 2.6795   | 5.7144    | 1.9815       |
| 4   | 2.3332   | 6.3815    | 2.1937       |
| 5   | 3.0264   | 5.9082    | 2.3718       |

### Table A29 Bivariate 10 input lags

| Lag | RNN MAPE | LSTM MAPE | α₁-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 4.4243   | 8.7381    | 1.3157       |
| 2   | 3.7217   | 9.1994    | 1.6702       |
| 3   | 5.9833   | 9.5561    | 1.9839       |
| 4   | 3.9705   | 9.5775    | 2.2366       |
| 5   | 4.8024   | 9.6431    | 2.4917       |

### Table A30 Univariate 21 input lags

| Lag | RNN MAPE | LSTM MAPE | α₁-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 4.4237   | 7.7809    | 1.2187       |
| 2   | 2.9773   | 7.9345    | 1.5905       |
| 3   | 5.4763   | 7.2335    | 1.6570       |
| 4   | 3.1714   | 7.8949    | 1.9313       |
| 5   | 4.1796   | 7.8409    | 2.5372       |

### Table A31 Bivariate 21 input lags

| Lag | RNN MAPE | LSTM MAPE | α₁-RIM MAPE |
|-----|----------|-----------|--------------|
| 1   | 5.7473   | 11.0067   | 1.5181       |
| 2   | 4.2199   | 10.8874   | 2.0879       |
| 3   | 6.0034   | 11.9115   | 2.2657       |
| 4   | 4.8727   | 11.046    | 2.6279       |
| 5   | 4.8366   | 11.172    | 3.0677       |

Re-scaled bivariate metrics of 5 input lags for the Brown Forman stock.

Re-scaled bivariate metrics of 10 input lags for the Brown Forman stock.

Re-scaled bivariate metrics of 21 input lags for the Brown Forman stock.
# THERMO FISCHER

## Table A32  Univariate 5 input lags

| Lag | RNN MAPE | LSTM MAPE | $\alpha_t$-RIM MAPE |
|-----|----------|-----------|---------------------|
| 1   | 9.6686   | 11.9074   | 2.1803              |
| 2   | 8.3009   | 11.8454   | 2.4475              |
| 3   | 9.9669   | 12.1983   | 2.7177              |
| 4   | 6.0381   | 11.1738   | 2.7458              |
| 5   | 7.5086   | 11.6914   | 3.0844              |

Re-scaled univariate metrics of 5 input lags for the Thermo Fischer stock.

## Table A33  Bivariate 5 input lags

| Lag | RNN MAPE | LSTM MAPE | $\alpha_t$-RIM MAPE |
|-----|----------|-----------|---------------------|
| 1   | 15.4747  | 20.6451   | 1.8585              |
| 2   | 11.2671  | 21.233    | 2.2550              |
| 3   | 15.5406  | 21.135    | 2.5692              |
| 4   | 10.8006  | 20.0298   | 2.4522              |
| 5   | 11.6321  | 19.922    | 2.5253              |

Re-scaled bivariate metrics of 5 input lags for the Thermo Fischer stock.

## Table A34  Univariate 10 input lags

| Lag | RNN MAPE | LSTM MAPE | $\alpha_t$-RIM MAPE |
|-----|----------|-----------|---------------------|
| 1   | 4.9015   | 11.0398   | 2.7522              |
| 2   | 3.4432   | 12.4441   | 2.9103              |
| 3   | 5.2922   | 10.1909   | 2.4824              |
| 4   | 3.5335   | 10.8379   | 2.5419              |
| 5   | 4.2818   | 8.9843    | 2.5887              |

Re-scaled univariate metrics of 10 input lags for the Thermo Fischer stock.

## Table A35  Bivariate 10 input lags

| Lag | RNN MAPE | LSTM MAPE | $\alpha_t$-RIM MAPE |
|-----|----------|-----------|---------------------|
| 1   | 17.8271  | 18.6483   | 1.7189              |
| 2   | 13.5076  | 19.7892   | 1.9708              |
| 3   | 18.0445  | 20.2516   | 2.2388              |
| 4   | 11.2579  | 18.7996   | 2.4439              |
| 5   | 15.5029  | 17.412    | 2.7314              |

Re-scaled bivariate metrics of 10 input lags for the Thermo Fischer stock.

## Table A36  Univariate 21 input lags

| Lag | RNN MAPE | LSTM MAPE | $\alpha_t$-RIM MAPE |
|-----|----------|-----------|---------------------|
| 1   | 8.8081   | 9.1374    | 2.8521              |
| 2   | 5.0309   | 9.6582    | 2.7941              |
| 3   | 6.5571   | 9.7462    | **2.5004**          |
| 4   | 4.1092   | 10.6142   | 2.6365              |
| 5   | 5.0009   | 7.7130    | 2.6647              |

Re-scaled univariate metrics of 21 input lags for the Thermo Fischer stock.

## Table A37  Bivariate 21 input lags

| Lag | RNN MAPE | LSTM MAPE | $\alpha_t$-RIM MAPE |
|-----|----------|-----------|---------------------|
| 1   | 11.537   | 20.6396   | 2.4834              |
| 2   | 8.5560   | 19.5691   | 2.3457              |
| 3   | 14.5108  | 19.4300   | **2.1511**          |
| 4   | 10.5839  | 19.5961   | 2.4707              |
| 5   | 9.5467   | 19.7083   | 2.8236              |

Re-scaled bivariate metrics of 21 input lags for the Thermo Fischer stock.


References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... Zheng, X. (2016, November). Tensorflow: A system for large-scale machine learning. 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265–283). Savannah, GA: USENIX Association. Retrieved from https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi

Adebiyi, A., Adewumi, A., Ayo, C. (2014, 03). Comparison of arima and artificial neural networks models for stock price prediction. Journal of Applied Mathematics, 2014, 1-7. 10.1155/2014/614342

Bengio, Y. (2017). The consciousness prior. CoRR, abs/1709.08568. Retrieved from http://arxiv.org/abs/1709.08568 https://arxiv.org/abs/1709.08568

Carruthers, P. (2006). The architecture of the mind: Massive modularity and the flexibility of thought. Oxford: Oxford University Press UK.

Cho, K., Courville, A., Bengio, Y. (2015). Describing multimedia content using attention-based encoder-decoder networks. IEEE Transactions on Multimedia, 17(11), 1875-1886. 10.1109/TMM.2015.2477044

Chung, J., Gulcehre, C., Cho, K., Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling.

Dixon, M. (2020). Industrial forecasting with exponentially smoothed recurrent neural networks. arXiv preprint arXiv:2004.04717v2.

Drury, M. (2017). Polynomial, spline, gaussian and binner smoothing are carried out building a regression on custom basis expansions. https://github.com/madrury/basis-expansions/blob/master/examples/comparison-of-smoothing-methods.ipynb. GitHub.

Galassi, A., Lippi, M., Torroni, P. (2020). Attention in natural language processing. IEEE Transactions on Neural Networks and Learning Systems, 1–18. Retrieved from http://dx.doi.org/10.1109/TNNLS.2020.3019893 10.1109/tnnls.2020.3019893
Glorot, X., & Bengio, Y. (2010, 13–15 May). Understanding the difficulty of training deep feedforward neural networks. Y.W. Teh & M. Titterington (Eds.), Proceedings of the thirteenth international conference on artificial intelligence and statistics (Vol. 9, pp. 249–256). Chia Laguna Resort, Sardinia, Italy: PMLR. Retrieved from http://proceedings.mlr.press/v9/glorot10a.html

Goyal, A., Lamb, A., Hoffmann, J., Sodhani, S., Levine, S., Bengio, Y., Schölkopf, B. (2020). Recurrent independent mechanisms. arXiv preprint arXiv:1909.10893v6.

Harsh Panday, V.S.P., V. Vijayarajan (2020). Stock prediction using sentiment analysis and long short term memory. European Journal of Molecular & Clinical Medicine, 7(2), 5060-5069. Retrieved from https://ejmcm.com/article_3126.html https://ejmcm.com/article__16b444b632c88db6fed0c6558dd6930a3126.pdf

Hazimeh, H., Zhao, Z., Chowdhery, A., Sathiamoorthy, M., Chen, Y., Mazumder, R., ... Chi, E.H. (2021). Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. CoRR, abs/2106.03760. Retrieved from https://arxiv.org/abs/2106.03760 https://arxiv.org/abs/2106.03760

Henaff, M., Szlam, A., LeCun, Y. (2016). Orthogonal rnns and long-memory tasks. CoRR, abs/1602.06662. Retrieved from http://arxiv.org/abs/1602.06662 https://arxiv.org/abs/1602.06662

Henaff, M., Weston, J., Szlam, A., Bordes, A., LeCun, Y. (2016). Tracking the world state with recurrent entity networks. CoRR, abs/1612.03969. Retrieved from http://arxiv.org/abs/1612.03969 https://arxiv.org/abs/1612.03969

Hochreiter, S., & Schmidhuber, J. (1997, 11). Long Short-Term Memory. Neural Computation, 9(8), 1735-1780. Retrieved from https://doi.org/10.1162/neco.1997.9.8.1735 https://arxiv.org/abs/https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf 10.1162/neco.1997.9.8.1735

Kim, M. (2015, 10). Cost-sensitive estimation of arma models for financial asset return data. Mathematical Problems in Engineering, 2015, 1-8. 10.1155/2015/232184

Kim, S., & Kang, M. (2019). Financial series prediction using attention lstm.
Kingma, D.P., & Ba, J. (2017). *Adam: A method for stochastic optimization*.

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., Zemel, R. (2018). *Neural relational inference for interacting systems*.

Königstein, N. (2021). *Dynamic and context-dependent stock price prediction using attention modules and news sentiment*. https://github.com/Nicolepcx/alphat-RIM. GitHub.

Pearl, J. (2009). *Causality: Models, reasoning and inference* (2nd ed.). USA: Cambridge University Press.

Peters, J., Janzing, D., Schölkopf, B. (2017). *Elements of causal inference: foundations and learning algorithms*. Cambridge: The MIT Press.

Pourahmadi, M. (2016). *Time series modelling with unobserved components*, by matteo m. pelagatti. published by crc press, 2015, pages: 257. isbn-13: 978-1-4822-2500-6. *Journal of Time Series Analysis*, 37(4), 575-576. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/jtsa.12181 https://arxiv.org/abs/https://onlinelibrary.wiley.com/doi/pdf/10.1111/jtsa.12181 https://doi.org/10.1111/jtsa.12181

Prado, M.L.d. (2018). *Advances in financial machine learning*. New York: Wiley.

Santoro, A., Faulkner, R., Raposo, D., Rae, J., Chrzanowski, M., Weber, T., … Lillicrap, T. (2018). *Relational recurrent neural networks*.

Schoelkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., Mooij, J. (2012). *On causal and anticausal learning*.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V., Soman, K. (2017). *Stock price prediction using lstm, rnn and cnn-sliding window model*. 2017 *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 1643-1647.

Simon, H.A. (1991). *The architecture of complexity*. In *Facets of systems science* (pp. 457–476). Boston, MA: Springer US. Retrieved from https://doi.org/10.1007/978-1-4899-0718-9_31 10.1007/978-1-4899-0718-9_31

Sugiyama, M., & Kawanabe, M. (2012). *Machine learning in non-stationary environments: introduction to covariate shift adaptation*. Cambridge: MIT.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., ... Polosukhin, I. (2017). Attention is all you need. *CoRR, abs/1706.03762*. Retrieved from http://arxiv.org/abs/1706.03762  https://arxiv.org/abs/1706.03762

Wang, Y., Huang, M., Zhu, X., Zhao, L. (2016, November). Attention-based LSTM for aspect-level sentiment classification. *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 606–615). Austin, Texas: Association for Computational Linguistics. Retrieved from https://aclanthology.org/D16-1058 10.18653/v1/D16-1058

Zhang, X., Liang, X., Li, A., Zhang, S., Xu, R., Wu, B. (2019, 08). At-lstm: An attention-based lstm model for financial time series prediction. *IOP Conference Series: Materials Science and Engineering, 569*, 052037. 10.1088/1757-899X/569/5/052037