Original Research Article

Feedforward Neural Networks: Cross-Validation of a Break-Out Range Strategy

Ulrich R. Deinwallner, PhD[*]

PhD Management and Finance, Walden University, USA

Abstract

One main idea guided this research article: can feedforward neural networks (FNN) be used by investors to cross-validate their investment decisions and stock market strategy entry-signals. Therefore, the research question of this study was: how profitable is a break-out strategy if the strategy is cross-validated by a FNN for U.S. stock markets? The study followed a quantitative, quasi-experimental design, regarding in-sample and out-of-sample tests. For the method of analysis, five hidden layers of a FNN were computed, a sigmoidal function, and break-out strategy conditions as well as entry-signals. As a result, the break-out strategy was profitable to trade with a money management strategy; however, the transaction costs had an effect if for the exit strategy the assets were sold at the end of the day. The FNN could only provide cross-validating results if a dichotomies entry-signal variable was added to the model. The study is relevant for portfolio managers and investors, who are interested in a second assessment of their data or market entry decisions through a cross-validation performed by a FNN.

Keywords: Portfolio Management, Feedforward Neural Networks, Break-Out Strategy, Sigmoidal Function.

INTRODUCTION

Technical Analysis Break-Out Range Strategy

Brock, Lakonishok and LeBaron [1] investigated one of a simpler trading strategy and applied in their research trading range break-out rules to the stock markets, while testing the break-out strategy through the method of bootstrapping. As a result, the buy-signals lead to high return results in regard of the break-out strategy compared to the generated sell-signals. An explanation for this observation was that buy-signals of a break-out strategy exhibited lower volatility compared to the higher volatility of sell-signals. Gençay [2] found for the effect, of different volatility profiles for long or short trades, an asymmetric behavior present in the obtained returns and saw evidence for nonlinearities as mechanisms for the generated data. This nonlinearity behavior makes the break-out rule interesting and relevant to investigate through a neural network (NN). On the one hand the break-out strategy is simple to apply, and on the other hand the strategy is significant in regard of its excess returns.

For the concept behind the break-out range strategy, the following can be mention. The trading range, of a break-out range strategy, is determined between the gap of the local top (or the resistance level) and the local bottom (or the support level) of a security price. According to this concept, investors have the tendency to sell at a resistance level, since a resistance level is a psychological mark and selling pressure can occur (vice versa). If the price closes above (or below) one of these levels, then buy (or sell) -signals can be recognized because further buying (or selling) actions can be initiated by other investors. The logic behind the break-out range trading strategy is, buy low and sell high, however, buy high and sell low if the security price moves above (or below) the resistance (or support level; [3]).

Feedforward Neural Networks and Nonlinear Predictions

In recent times, the nonlinear prediction of buy and sell -signals for stock market strategies have become more and more popular and an issue in the finance scholarly literature. Gençay [2] investigated historical foreign exchange returns for technical trading rules and discussed the predictability through feedforward neural networks (FNN), while applying a nearest neighbor regression for the computation. Interestingly, Gençay found certain predictability proficiency for technical trading rules and that a single

hidden layer for the feedforward network was in this case sufficient to generate significant results. However, the optimal number of hidden layers in a FNN and the optimal size of the training data depended on the cross-validation method through which the mean square errors were minimized. This means that the optimal number of hidden units and the number of training data can vary depending on the number of observations of the prediction data sample [2]. Gupta and Wang [4] used FNN's to forecast and trade index futures of the S&P500. Khoa, Sakakibara, and Nishikawa [5] used a recursive network to adjust for time series and profits as weight factors, while Sezer and Ozbayoglu [6], Tsantekidis *et al.* [7] had success with convolutional NN (CNN) to drive from these networks significant results by considering limit-order-book data or the application of stock market indicators. In conclusion, the different kinds of NN could provide meaningful prediction on a nonlinearity basis for investment decisions. This means that all different forms of NNs such as CNN's, recurrent NN's (RNN), long short term memory (LSTM), deep belief networks (DBN), or restricted Boltzmann machines (RDM) could provide prediction results with a precision degree of about 60%-70% for stock markets (see Sezer and Ozbayoglu [6]). These precision dimensions are quite exceptional since hit rates that lie above >51% can already generate in the long-run positive returns for an investment strategy with money management rules applied. In perspective, average values for trading strategies are in praxis rather at values assumed of about 53%-57%.

Several studies have reported significant findings for the NN in regard of stock market predictions. For this, researchers have used and compared all kinds of indicators and influencing variables for the computation of NNs that could impact security price by predicting entry- and exit- signals. Since these NNs are fairly complex in their construction and can require some programming skills, the question is, if simple NNs can provide meaningful and profitable results for a cross-validation of investment strategies? Therefore, it seemed relevant to understand how crucial the complexity of NNs really is in order to make predictions and in order to identify entry-signals for a trading strategy through a FNN. The novelty of the study is that current stock market data is processed to closer asses a FNN and the break-out strategy. The study is significant because it can contribute to the literature of technical analysis and it can contribute to the literature of NNs. Older findings for example made by Brock, Lakonishok and LeBaron [1] or Gençay [2] could be verified under present market conditions by this study. The study can be relevant for portfolio managers who are interested to quantify and cross-validate their investment decisions through a FNN.

Purpose and Research Question

The purpose of this quantitative, quasi-experimental study is to investigate the impact and profitability of a break-out strategy on the US stock market, while verifying and cross-validating the predictions for the entry-signals through a FNN. Therefore the study is guided by one research question (R), R1: How profitable is a break-out strategy if the strategy is cross-validated by a FNN for U.S. stock markets?

To answer the research question one Hypothesis (*H*) is relevant to investigate:

$H_0 1$: if a FNN is computed and perditions are made for a cross-validation of market entry-signals and the profitability of a break-out strategy, then the break-out strategy does not exhibit profitable significant returns.

$H_1 1$: if a FNN is computed and perditions are made for a cross-validation of market entry-signals and the profitability of a break-out strategy, then the break-out strategy does exhibit profitable significant returns.

Summary

For this study, Dow Jones Industrial Average (DJIA) index data was considered for the time period of 1996-2019. First the training data was investigated; through an in-sample test (see Table 1) in regard of its profitability distribution and profitability metrics. Then, a diagram (see Figure 1) was presented that displayed the operationalization of the FNN, presenting the considered variables, hidden layers and outcome for the FNN. Finally, an out-of-sample test (see Table 2) was computed with the test data for the FNN to make predictions of the market entry signals and to cross-validate the results of the break-out strategy. In the Appendix (see Table A1) the optimized weights, biases and minimized sum of squared errors (SSE) are presented. The results revealed that a money management, with -1% stops for daily losses, was relevant for the break-out strategy to obtain significant positive returns. The transaction costs had an effect on the profitability of the break-out strategy, if for the exit-strategy the assets were sold at the end of the day. This paper is different to Brock, Lakonishok and LeBaron [1] paper and Gençay [2], because it applies the break-out strategy in a different, simple form and applies a FNN with the pre-setting of five hidden layers. Essentially, the FNN can be an interesting supplement to support quantitative investment decisions; however, more complex NNs are required for more sophisticated investment decisions. This paper is relevant for portfolio managers or investors who are interested in technical analysis and the support of FNNs to quantify and cross-validate their investment decisions.

Data and Limitations

For the analysis, I use open, high, low, and dividend adjusted daily closing prices from the DJIA index for the time period of January 01, 1996 until December 31, 2019 (24 years). I collect the DJIA data from the internet source Yahoo! Finance (see Ren & Ren, 2018). The computation of the break-out strategy and the FNN model will require the software Microsoft Excel.

I make the following limitations for the results for this study and for the data of the DJIA index. Size effects might not be present in the selected data since the DJIA index only represents large cap stocks. Other NN models have been mentioned in the scholarly literature besides the FNN that might provide higher precision rates for a prediction of entry-signals of an investment strategy; however, the scope of this study is only based on the investigation of a simple form of a NN construction. The findings can be relevant for investors and portfolio managers who are interested to support their investment decisions by a nonlinear quantitative approach.

METHODOLOGY

In regard of the method to compute a FNN, a computation can be conducted by R Studio, with the package "neuralnet", while the authors Riedmiller and Braun [8], Anastasiadis, Magoulas, and Vrahatis [9], and Intrator and Intrator [10] have reported more to the application of the R software.

In this study, the FNN is computed through Microsoft Excel and follows the following equations:

The hidden layer (Hl) is computed by:

$$Hl_n = \frac{1}{1 + e^{weight_1 * data_1 + ... + weight_n * data_n + bias_{Hl_n}}} \qquad [1]$$

Where $n = 1, ..., k$, the data are for example stock returns or stock index prices, while the initial weights are for the start random numbers that later need to be optimized to make a prediction. The equation is a sigmoidal function and the data exhibit in a chart a S-form that can only have values from 0 until 1.

The estimated score (ES) is computed through:

$$ES = Hl_1 * weight_1 + ... + Hl_n * weight_n + bias_{ES} \qquad [2]$$

Where the bias is as well at the beginning of the computation a random number that later need to be optimized to make a prediction.

The computation of the predicted probability (PP) is computed through:

$$PP = \frac{1}{1 + e^{ES}} \qquad [3]$$

And the squared error (SE), which needs to be minimized trough the Excel solver to obtain the optimal weights and biases, is computed by:

$$SE = (AC - PP)^2 \qquad [4]$$

Where the Actual Signal (AS) is a binary variable of 0 or 1, depending if the AS for the investigated data is false or true and is determined for example by hand from the research for the training data. This means, if a buy signal occurs from the training data the AS value would be 1 else 0. This binary value will later be predicted through the FNN. The Predicted Signal (PS) will be programmed by the command: if the ES is larger than 50%, then print the value 1, else 0. In order to optimize the FNN the following values also need to be computed: (a) the SSE, which is the sum of the SE; (b) A penalty value can be computed by:

$$Penalty = \varphi * squared\ sum(weights_{1,...,n}, biases_{1,...,n}, bias_{ES})/(n*2) \qquad [5]$$

Where lambda φ is an error-term a can be adjusted to fit the results, while n is the number of weights and biases that are squared and added to a sum. The values in the parentheses of the Equation 5 need to be optimized. The value that needs to be minimize with the Excel solver are the SE or the Minimize value computed through Minimize = SSE + Penalty (see Table A1).

The transaction costs are computed as in Brock, Lakonishok, and LeBaron [1] described, with $30 for up to 1,000 shares independent of the shares prices. An assumption is made that each share is worth $100 and 10,000 shares are considered per trade, which equals to a transaction of $1 Mill. This means that the total transaction costs are computed as $600 per trade. The $600 transaction costs per trade include the opening and closing of the daily position. For example, the total transaction costs for 250 days are equal to $150 000 of $1 Mill. Investment amount; or 0.06% for one trade.

Research Design

The following steps will be tested in regard of the research design: (a) First, I will conduct an in-sample-test with the training data for a break-out strategy with the DJIA index return data to investigate the distribution and profitability of the break-out strategy. (b) Then, I present a diagram for the operationalization of the FNN (see Figure 1) and display the considered variables, hidden layers, and outcome. (c) Essentially, I conduct an out-of-sample test with the test data for the break-out strategy and with the DJIA index return data to investigate the profitability and the relevant metrics if the break-out strategy entry signals are predicted through a FNN for a cross-validation of these entry-signals. The weights and the biases for the FNN will be obtained from an optimization of the training data and will be applied as pre-setting to the test data.

Because private investors tend to lose capital in the financial markets, this study is for educational reasons, does not represent investment advice, and investors should conduct their own research before investing their capital [11-13]. Threats to external validity are that analyzing just the DJIA index might not be representative for an inference conclusion for the U.S. stock markets in general. Further, the findings assessed from a stock market index might not apply for the market behavior of different stocks and securities, for example for small cap stocks. In this study, the break-out strategy is applied in a very simple form and can lead with longer ranges to different or more profitable results through which external validity issue can occur for making inferences in regard of break-out range strategy statements.

Threats to internal validity are that the computations could be not precise because of rounding or different fitting methods for a NN (i.e. over-fitting and under-fitting). For the computation of a FNN different computation methods for example through (R Studio packages) can lead to deviating results. Further the hidden layers had a pre-setting of five hidden layers and were not evaluated in different variations, which could have an impact for an optimal result. Essentially, the transaction costs, performed in this study, are only one form of approaching costs issues and do not represent all different kinds of broker agreements that mean while exist in regard of costs and can therefore lead to deviation when interpreting the net profit results profitability.

EMPIRICAL FINDINGS

For the analysis of H1, first the profitability of a break-out strategy needs to be investigated through an in-sample test in order to see the profitability distribution. In Figure 1, I can find that for the training data, the break-out strategy, as it is constructed in this study, is only profitable if a money management strategy is applied, with stops at -1% daily losses. Relevant larger profits occurred during 1996-1998 and 2016 -2018 of total returns on average per year with 10% and 7%. The buy and hold (B&H) strategy was only outperformed in two time periods, 2000-2002 and during the financial crises around 2008. For the trading strategy short sales were not allowed in this test-setting. The reason is that short sales did not seem to be so profitable compared to long positions for the break-out strategy, and a reason was seen in a higher volatility that can diminish the positive returns for short sales [1].

The break-out strategy was computed by the following algorithm:

(1)   Is the break-out "condition" met?

$$IF((\text{Open Price}_n < \text{Close Price}_n), 1, 0) \qquad [6]$$

    Where 1 represents that the "condition" is met.

(2)   Does a buy-signal occur?

$$IF(AND(\text{Condition} = 1, \text{Close Price}_n >= \text{High Price}_{n-1}), 1, 0) \qquad [7]$$

    Where 1 represents a buy-signal.

Each trade was held one day and was sold at the end of the day. The transaction costs revealed that the break-out strategy is in this form and under these constrains not consistently profitable to trade with total net losses of around -6% per year on average. The Hit-rate informs that in more than 42% until 57% of the cases the returns were positive. The break out-range strategy can be more profitable, if a leverage is applied with a factor x2. This would mean for example an average yearly total return of 3%, and impacted by the transaction costs with an average total net return of -2%, would result with leverage of x2 at about 1% total net return per year. It is to mention that leverage can lead in consequence to larger deviation in a portfolio and could even lead, depending on the equity and leverage size, to a knockout of a portfolio in case of longer loss series.

Table-1

In-Sample test - Training Data; Averages per Year						
Tests	1996-1998	2000-2002	2004-2006	2008-2010	2012-2014	2016-2018
Total return	0.10	0.05	-0.01	0.03	0.04	0.07
Net return	0.05	0.01	-0.06	-0.02	-0.01	0.02
σ	0.004	0.006	0.003	0.005	0.003	0.003
μ/σ	23.85	8.27	-4.24	6.07	13.74	20.68
Hit rate	0.54	0.42	0.48	0.50	0.52	0.57
B&H return	0.16	-0.10	0.05	-0.05	0.11	0.09
σ DJIA	0.006	0.008	0.004	0.010	0.004	0.005
μ/σ DJIA	25.88	-12.45	14.08	-5.00	27.20	19.42
Conditions	140	124	134	133	138	137
Buy Signals	87	65	70	75	83	83
Positive returns	47	27	34	38	43	47

Note. In the table, the performance results of a break-out strategy is investigated in an in-sample test. The time period was three years each for the training data during 1996 until 2019, while the remaining year(s) were selected for the test data. The break-out strategy values are one year averages in percent for a better comparison with the out-of-sample test, while for the results a money management strategy of stops at -1% daily losses was applied. Sigma is the standard deviation, and Mu is the total return. The Hit rate is the amount of positive returns divided by the amount of buy-signals. The condition is the algorithm number 6 and the buy-signal is algorithm 7.

Next, a FNN was computed according to the conceptualization and operationalization of Figure 1. For the data Open, High, Low, Close DJIA prices were considered and were divided by 10,000 to obtain smaller values. Also, a binary variable of $N(0,1)$ was used displaying market entry singles 1 else none 0 for the break-out strategy. It is to mention that a simple FNN does not provide meaning full results for a prediction if only stock market data are selected for the prediction of break-out strategy entry-signals. Therefore, a variable with break-out strategy entry-signal was added to the FNN for a prediction of the outcome. The outcome was again entry-signals of the break-out strategy; however, these signals were used to cross-validation the initial break-out strategy investment decisions in the subsequent out-of-sample test.

In regard of the FNN construction, a cross-evaluation of the hidden layers, as recommended by Gençay [2], was not conducted to optimize the hidden layers of the FNN, since this objective was not a part of the test-design. Instead, five hidden layers were selected as a pre-setting to conduct the computations of the FNN for the out-of-sample test.

The following can be mentioned to the training and test data. The training data exhibited three years of DJIA index data to compute the weights and the biases metrics, while the test data exhibited one year of DJIA index return data [1]. In the Appendix Table A1, the data for the solved and optimized weights and biases; and the minimized SSE is provided.
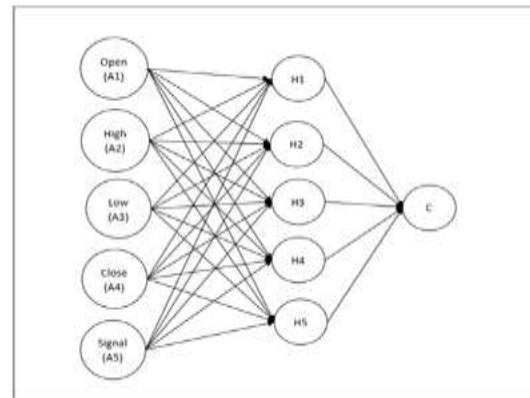
Fig-1: Operationalization of the FNN. Displaying the FNN variables, hidden layers, and predicted outcome

For the out-of-sample test; first, the training data was considered, to minimize the error-terms in order to obtain the weights and biases of the FNN (see Table A1). Then, the test data was used to determine the predictions of the FNN with the training data pre-settings (i.e., weights and biases).

From Table 2, I can find that the highest returns occurred for the cross-validated break-out strategy in 2019 with 10% total return per year, while only in 2019 the B&H strategy was outperformed. The transaction costs were still an issue for the break-out strategy with two losses of -4% until -7% total net returns per year during 2007 and 2015. About 75 buy-signals occurred during a year in average, which equals to about 30% trading activities on a daily basis per 250 trading days a year. I cannot reject the Null Hypothesis 1, because the application of the transaction costs to the returns of the break-out strategy realistically do not show significant profitable results with -0.33% total net return on average per year under these constrains.

Table-2

Out-of-Sample Test per Year						
Tests	1999	2003	2007	2011	2015	2019
Total return	0.05	0.05	0.01	0.07	-0.03	0.10
Net return	0,01	0.01	-0.04	0.02	-0.07	0.05
σ	0.0089	0.0080	0.0056	0.0075	0.0066	0.0055
μ/σ	6.09	6.65	1.77	9.26	-4.22	17.98
Hit rate	0.36	0.51	0.55	0.5	0.41	0.58
B&H return	0.23	0.23	0.06	0.05	-0.02	0.19
σDJIA	0.010	0.010	0.009	0.013	0.010	0.008
μ/σ DJIA	23.00	23.00	6.67	3.85	-2.00	23,75
Conditions	124	137	145	140	121	141
Buy Signals	66	75	75	80	73	80
Positive returns	24	38	41	40	30	46

Note. In the table, the performance results of a break-out strategy were investigated in an out-of-sample test. The time period is from 1996 until 2019, and in this table the results for the test data (one year each) is displayed. The break-out strategy values are in percent. The results consider a money management strategy, were stops at -1% daily losses were applied. Sigma is the standard deviation, and Mu is the total return. The Hit rate is the amount of positive returns divided by the amount of buy-signals. The condition is the algorithm number 6, while the buy-signal is the algorithm number 7.

CONCLUSION

In conclusion, the research question could be answered of R1: how profitable is a break-out strategy if the strategy is cross-validated by a FNN for U.S. stock markets? The break-out strategy is profitable with 2.67% total return per year on average, if money management with a stop or -1% daily losses is applied and cross-validated by a FNN (see Table 1); however, the transaction costs impacted the performance of the break-out strategy significantly and made the total net returns of the strategy with the selected exit rules inconsistent with -0.33% net total return per year on average. A simple FNN only provided meaningful results for a cross-validation, if a dichotomous variable with a strategies entry-signal was added in order to obtain prediction data. The study could corroborate Brock, Lakonishok, & LeBaron [1], findings that FNN can be used for an investment strategy entry-signal prediction. The study can contribute to the literature of NNs. During the investigation of the literature for this study, CNNs seemed to provide sophisticated and most promising results in order to make stock marked predictions based on the transformation of charts of indictor data into images and could provide further research potential for other researchers. However, the study is relevant for portfolio managers and investors who are interested in a second assessment of their data or market entry decisions through the cross-validation of a FNN.

REFERENCES

1. Brock, W.A., Lakonishok, J., & LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance* 47(5), 1731–1764. https://doi.org/10.1111/j.1540-6261.1992.tb04681.x
2. Gençay, R. (1997b). Linear, nonlinear and essential foreign exchange prediction. *Journal of International Economics 47*, 91-107. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.451.7823&rep=rep1&type=pdf
3. Fell, J. P. C. (1994). Technical trading rules and exchange rate volatility, 1-68. https://www.researchgate.net/profile/John_Fell5/publication/283726941_Technical_Trading_Rules_and_Exchange_Rate_Volatility/links/5751ba2308ae6807fafb4da7/Technical-Trading-Rules-and-Exchange-Rate-Volatility.pdf
4. Gupta, S., & Wang, L. P. (2010). Stock forecasting with feedforward neural networks and gradual data sub-sampling. *Australian Journal of Intelligent Information Processing Systems*, *11*(4), 14-17. http://ajiips.com.au/papers/V11.4/AJIIPS_vol11n4.pdf#page=17
5. Khoa, N. L. D., Sakakibara, K., & Nishikawa, I. (2006). Stock price forecasting using back propagation neural networks with time and profit based adjusted weight factors. *SICE-ICASE International Joint Conference,* 5484-5488. https://doi.org/10.1109/SICE.2006.315683
6. Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, *70*, 525-538. https://doi.org/10.1016/j.asoc.2018.04.024
7. Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2017). Forecasting stock prices from the limit order book using convolutional neural networks. *IEEE 19th Conference on Business Informatics (CBI), 1*, 7-12. https://doi.org/10.1109/CBI.2017.23
8. Riedmiller M., & Braun H. (1993). A direct adaptive method for faster backpropagation learning: The RPROP algorithm. IEEE International Conference on Neural Networks (ICNN), 586-591. https://doi.org/10.1109/ICNN.1993.298623
9. Anastasiadis, A. D., Magoulas, G. D., & Vrahatis, M. N. (2005). New globally convergent training

scheme based on the resilient propagation algorithm. *Neurocomputing*, *64*, 253-270. https://doi.org/10.1016/j.neucom.2004.11.016

10. Intrator, O., & Intrator, N. (1993). Using neural nets for interpretation of nonlinear models. *Proceedings of the Statistical Computing Section*, 244-249. San Francisco: American Statistical Society (eds).

11. Abbey, B. S., & Doukas, J., A. (2012). Is technical analysis profitable for individual currency traders? *Journal of Portfolio Management, 39*(1), 142-150. https://doi.org/10.3905/jpm.2012.39.1.142

12. Gencay, R. (1997a). Optimization of technical trading strategies and the profitability in security markets. *Economics Letters*, *59*(2), 249-254. https://doi.org/10.1016/S0165-1765(98)00051-2

13. Ren, L., & Ren, P. (2018). Applying a combined max-min simple moving average trading strategy to market indexes. *Economics, Management, & Financial Markets, 13*(2), 11-23. https://www.ceeol.com/search/article detail?id=676861

APPENDIX

Table-A1

Optimized and Minimized Weights, Biases, SSE, Minimal Value						
Metrics	1996-1998	2000-2002	2004-2006	2008-2010	2012-2014	2016-2018
A1 to 1 weight	2.21	1.91	1.91	0.95	1.11	0.69
A2 to 1 weight	2.21	1.90	1.90	0.95	1.11	0.69
A3 to 1 weight	2.18	1.88	1.88	0.94	1.10	0.69
A4 to 1 weight	2.18	1.87	1.87	0.94	1.10	0.69
A5 to 1 weight	24.06	14.29	14.29	4.82	4.66	1.43
A1 to 2 weight	0.44	2.09	2.09	0.81	0.69	0.73
A2 to 2 weight	0.44	2.08	2.08	0.81	0.69	0.73
A3 to 2 weight	0.44	2.05	2.06	0.80	0.69	0.73
A4 to 2 weight	0.44	2.04	2.05	0.80	0.69	0.73
A5 to 2 weight	7.66	-19.17	-16.40	-1.23	1.83	3.35
A1 to 3 weight	1.27	-0.61	-0.61	0.29	0.50	0.11
A2 to 3 weight	1.26	-0.61	-0.61	0.29	0.50	0.11
A3 to 3 weight	1.25	-0.60	-0.59	0.29	0.50	0.11
A4 to 3 weight	1.24	-0.60	-0.60	0.29	0.49	0.11
A5 to 3 weight	11.25	-10.40	-10.35	-1.46	0.54	-1.29
A1 to 4 weight	0.07	1.51	1.51	0.63	0.49	0.53
A2 to 4 weight	0.08	1.53	1.53	0.63	0.49	0.54
A3 to 4 weight	0.09	1.51	1.51	0.63	0.49	0.54
A4 to 4 weight	0.09	1.53	1.53	0.64	0.49	0.54
A5 to 4 weight	19.56	14.21	14.21	6.22	6.63	7.31
A1 to 5 weight	0.07	1.51	1.51	0.63	0.49	0.53
A2 to 5 weight	0.08	1.53	1.53	0.63	0.49	0.54
A3 to 5 weight	0.09	1.51	1.51	0.63	0.49	0.54
A4 to 5 weight	0.09	1.53	1.53	0.64	0.49	0.54
A5 to 5 weight	19.56	14.21	14.21	6.22	6.63	7.31
H1 to C weight	166.13	136.41	132.67	41.62	35.32	23.43
H2 to C weight	55.33	30.37	31.06	12.57	12.90	11.77
H3 to C weight	253.33	182.13	203.53	60.07	53.08	31.26
H4 to C weight	98.00	46.47	46.04	20.39	21.25	15.31
H5 to C weight	98.00	46.47	46.04	20.39	21.25	15.31
Bias into H1	24.06	14.29	14.29	4.82	4.66	1.43
Bias into H2	-0.16	15.99	17.92	3.48	1.79	1.60
Bias into H3	0.24	10.06	10.11	1.25	0.37	-0.89
Bias into H4	0.5	0.5	0.5	0.5	0.5	0.5
Bias into H5	0.5	0.5	0.5	0.5	0.5	0.5
Bias into C	-588.09	-334.00	-356.69	-127.14	-127.75	-61.98
SSE	1.335E-24	0.000128	0	1.07E-06	2.2588E-07	3.5E-08
Lamda	0	0	0	0	0	0
Penalty	0	0	0	0	0	0
Minimize	1.335E-24	0.000128	0	1.07E-06	2.2588E-07	3.5E-08
Note. In this table, the weights, biases, sum of square errors (SSE), the lambda (penalty value) are displayed. The equations for the metrics are presented in the method section and consider the Equations 1-5. The values of the metrics were obtained through minimizing the squared errors and by optimizing the weights and biases for the FNN.						