LUISS 'Guido Carli'
MSc in Data Science and Management - Machine Learning Course
May 3, 2024

AI Systematic Trading Challenge

Comparative Analysis of RNN and ANFIS for Stock Price Prediction

Project Technical Report

Coci, Marco, Team Leader
ID: 786471, email: m.coci@studenti.luiss.it

Dong, Thi Kieu Trang
ID: 772701, email: thi.dong@studenti.luiss.it

Navarra, Filippo
ID: 782571, email: filippo.navarra@studenti.luiss.it

Collaborative Business Case Proposed by Euklid

1 Introduction

In the rapidly evolving field of financial markets, leveraging advanced machine learning techniques to predict market trends has become critical to gain a competitive advantage. This project, undertaken as part of the AI Systematic Trading Challenge, aims to develop a robust AI-based trading model that can effectively address the complexities of global financial markets.

The company provided us with six key data sets for our analysis, focusing on three main stocks: Amazon, IBM and Microsoft; and three significant market indexes: Nasdaq, CAC 40 and SP500. This diverse selection allows us to explore both individual stock performance and broader market trends, providing a comprehensive view of the dynamics at play.

Our approach involves two sophisticated predictive models: Recurrent Neural Networks (RNN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS). RNNs were chosen for their expertise in handling sequential time series data, making them ideal for analyzing the temporal patterns inherent in stock prices. Meanwhile, the ANFIS model was selected for its ability to integrate fuzzy logic with neural networks, offering a powerful tool for managing the uncertainties and nonlinear relationships typical of financial datasets (Barak, Dahooie, and Tichý 2015; Esfahanipour and Mardani 2011).

A key innovation in our methodology is the development and implementation of a custom loss function. Traditional regression metrics such as Root Mean Square Error (RMSE) are often focused on the magnitude rather than the direction of price movements, which is a crucial aspect of trading strategies (Prabakaran et al. 2021; Felder and Mayer 2022). Our custom loss function is designed to overcome this limitation by emphasizing the accuracy of directional predictions, thus aligning more closely with real world trading needs.

This project not only tests these models according to standard RMSE criteria but also compares their performance using our innovative loss function. Through this comparative analysis, we aim to validate the effectiveness of our customized approach.

2 Methods

2.1 Covariates Construction

In the development of our systematic trading model, we worked on datasets organized weekly, where variables were crucial for conducting temporal and financial analysis that are:

- **Open, High, Low Prices (OHL)**: These variables represent the opening, highest, lowest, and closing prices within each week.
- **Volume**: This metric denotes the total number of shares or contracts traded during the week for a particular stock or index, reflecting market activity and liquidity.
- **Adjusted Close Price**: Adjusted for corporate actions like dividends, stock splits, and rights offerings, this measure offers a more accurate reflection of the stock's or index's value over time.
- **Date/Time Stamp**: Each record is associated with a specific week, facilitating the temporal analysis of financial metrics.

To augment our model's predictive capabilities, we also incorporated some common technical indicators:

- **SMA (Simple Moving Average)**: A Simple Moving Average (SMA) is an arithmetic moving average calculated by adding recent closing prices and then dividing that by the number of time periods in the calculation average.
- **EMA (Exponential Moving Average)**: An Exponential Moving Average (EMA) is a type of moving average that places a greater weight and significance on the most recent data points. It's more responsive to new information compared to the simple moving average.
- **Stochastic Oscillator (STOCH)**: The Stochastic Oscillator is a momentum indicator comparing a particular closing price of a security to a range of its prices over a certain period of time. The sensitivity of the oscillator to market movements is reducible by adjusting that time period or by taking a moving average of the result.
- **RSI (Relative Strength Index)**: The Relative Strength Index (RSI) is a momentum oscillator that measures the speed and change of price movements. RSI oscillates between zero and 100.
- **MACD (Moving Average Convergence Divergence)**: The Moving Average Convergence Divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price.

2.2 Target Construction

To effectively model and forecast financial outcomes, we focused on forecasting stock returns and used it as a target variable for our models. Logarithmic returns were used for their statistical properties, such as quasi-normal distribution and constant volatility, which simplify calculations and improve model stationarity (Martucci 2024). The formula used to calculate logarithmic returns is:

$$\text{Log Return} = \log\left(\frac{\text{Price}_t}{\text{Price}_{t-1}}\right) \tag{1}$$

Where P_t represent the current price and P_{t-1} the previous one.

2.3 Process Description

Our methodological approach is summarized in a sequential process flow:

- Data Cleaning: We began by addressing missing values and anomalies in our data to ensure accuracy and reliability in our predictions, we had 2 missing values for OHL variables just for index datasets.

- Technical Indicator Selection: Choosing the right indicators based on their predictive power and relevance to our analysis goals.

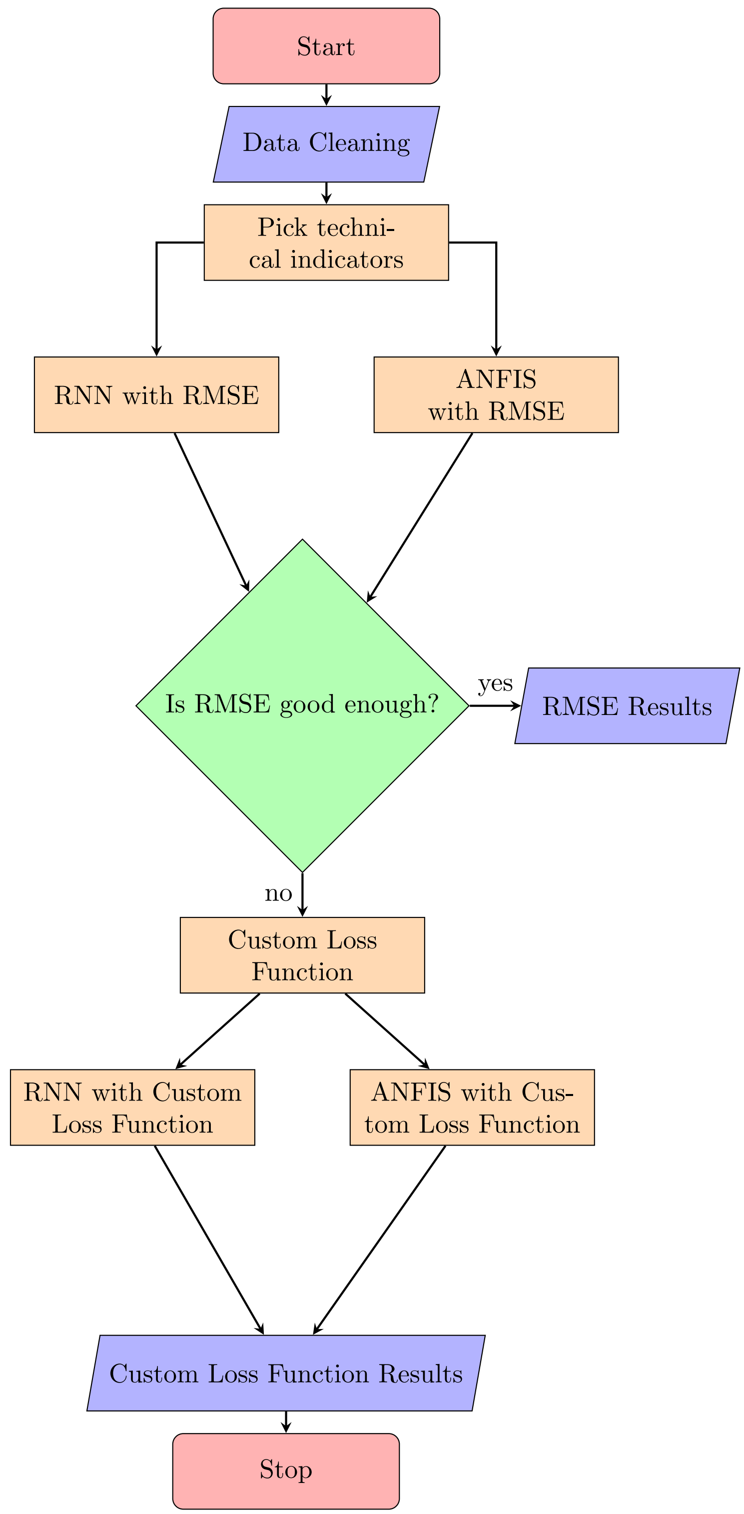
- Model Application:

  - RMSE as loss function: Initially, we applied RNN and ANFIS using the Root Mean Square Error (RMSE) loss function to predict price movements.

  - Evaluation of RMSE: Assessing whether RMSE alone could sufficiently capture the accuracy needed for our predictive models.

  - Custom Loss Function Implementation: Upon finding RMSE inadequate for capturing directional accuracy, we introduced a custom loss function aimed at enhancing predictions by penalizing incorrect directional forecasts more severely.

  - RNN and ANFIS with Custom Loss Function: Both models were then tested under the new loss function to compare performance improvements.



2.4 A new loss function

We realized that the RMSE results were not satisfactory, especially in our case of financial forecasts where the direction of changes is as important as their magnitude. Thus, we decided to create our custom loss function to try to improve the results of our model. One of the goals of our project is to demonstrate how this new loss function and

RMSE behave differently and to show that in some cases, our loss function performs better than RMSE.
Our custom loss function imposes a heavier penalty for incorrect predictions about the direction of price changes, not just the size of the error. This means that if the model predicts an increase when the price actually decreases, or vice versa, the error is considered more significant. The loss for each prediction takes into account both the size of the error and whether the prediction correctly captured the direction of change.

2.5 Models

As previously mentioned, we decided to apply two models, *Recurrent Neural Network* (RNN) and *Adaptive Neuro-Fuzzy Inference Systems* (ANFIS).

2.5.1 RNNs

RNNs are particularly suited for sequential data, processing time series data and remembering past events, which is a crucial element in the dynamic and interconnected world of financial markets. This makes them highly valuable for modeling sequences that demonstrate time-based patterns, such as financial market trends, where past prices can influence future ones.

The essential components of **RNN architecture** include:

- **Input Layer**: Responsible for receiving sequential data, where each element may represent a word or character in tasks like natural language processing.
- **Hidden Layer**: Maintaining an internal state that evolves as the network processes each element in the sequence, capturing information from previous time steps.
- **Recurrent Gate**: A crucial feature involving weights and connections looping back to the hidden layer from the previous time step, allowing the RNN to update its hidden state and remember past information.
- **Output Layer**: Produces predictions based on the information in the hidden state.

Specifically, we trained our model with the Adam optimizer with other hyper-parameters which are listed in Table 1.

2.5.2 ANFIS

For what it concerns ANFIS, it is recognized as a highly effective model for financial predictions, particularly because it combines the interpretability of fuzzy systems with the learning capability of neural networks. This allows it to handle uncertainty and model complex, nonlinear relationships that are typical in financial data. Fuzzy logic, unlike traditional binary systems, allows for a smoother transition between output states.

The essential components of **ANFIS architecture** include:

- **Input Layer**: The layer that handles the input data that the model receives. The input data is shaped and prepared for processing in subsequent layers.
- **Fuzzy Membership Layer**: This layer essentially fuzzifies the input data by calculating how much each input belongs to different fuzzy sets defined by the Gaussian parameters.
- **Fuzzy Rule Layer (T-Norm Layer)**: This layer models the conjunctions in the fuzzy rules, aggregating the degrees to which inputs satisfy fuzzy conditions.
- **Output Layer**: This layer performs the defuzzification process, turning the fuzzy quantities into a single crisp output, which is typical in a regression scenario.

We trained our models with Adam optimizer with L2 regularization (weight_decay=1e-5). Learning rate parameter has different values for each dataset and models. Learning rate scheduler parameters, Step size and Decay rate were set with following values; 10 and 0.95. Learning rate values and other hyper-parameters are listed in Table 2.

3 Experimental Designs

3.1 Main purpose

The main objective of our experiments was to evaluate the predictive capabilities of recurrent neural networks (RNNs) and adaptive neuro-fuzzy inference systems (ANFIS) in predicting stock price movements. Specifically, we aimed to evaluate how well these models could predict the direction of price changes by integrating a custom loss function designed to prioritize the accuracy of directional predictions.

3.2 Experimental Setup

We conducted our experiments using historical stock price data from six major stocks and indixes. Each dataset was preprocessed to normalize the data and to calculate relevant financial indicators like moving averages and volatility indices. We split the data into training, validation, and testing sets, respectively 70%,20% and 10%, to ensure robust model evaluation.

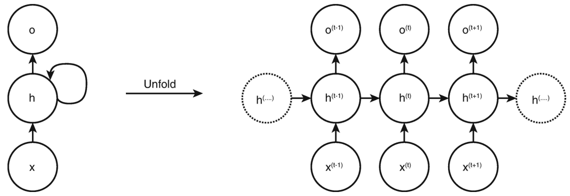


Figure 1: RNNs architecture

3.3 Custom loss function implementation

Our custom loss function was applied to both models to compare its effectiveness against traditional loss functions like RMSE. This function imposes heavier penalties for incorrect directional predictions, aligning more closely with real-world trading objectives where the direction of movement is more critical than the magnitude.

3.4 Evaluation

Results were quantitatively analyzed using accuracy, precision, recall, and F1 scores, and qualitatively discussed to draw conclusions about each model's performance under different experimental conditions. Comparisons were made not only between the models but also between the different loss functions used.

- **Accuracy**: The proportion of total correct predictions (both true positives and true negatives) out of all predictions made.

$$\text{accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

- **Precision**: The ratio of correct positive predictions (true positives) to the total predicted positives (both true positives and false positives).

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}} \tag{3}$$

- **Recall**: The ratio of correct positive predictions to the actual positives (both true positives and false negatives).

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}} \tag{4}$$

- **F1**: The harmonic mean of precision and recall, a measure that balances both metrics.

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

4 Results

Our analysis yielded insightful comparisons between the Recurrent Neural Network (RNN) and the Adaptive Neuro-Fuzzy Inference System (ANFIS) across different datasets and loss functions. The observed differences in model performance are crucial for understanding their applications in financial forecasting and trading strategies.

ANFIS demonstrated consistently higher performance across most metrics when using the Custom loss function, particularly indexes. This could be attributed to ANFIS' ability to model complex, non-linear relationships inherent in financial data more effectively than RNNs.
RNNs, while generally underperforming in comparison to ANFIS, showed more balanced performance across all datasets when evaluated with the custom loss function.
The Recurrent Neural Network (RNN) exhibited superior performance on the Amazon dataset, effectively capturing the underlying price trends. In contrast, the Adaptive Neuro-Fuzzy Inference System (ANFIS) was less effective in

Table 1: RNN Architecture for both loss functions

Hyperparameters	RNN with RMSE	RNN Custom loss function
Number of Units per Layer (Hidden Size)	20	20
Number of Layers	1	1
Learning Rate	0.001	0.001
Number of Epochs	100	100
Sequence Length	10	10
Activation Function	tanh	tanh
Batch Size	32	32
Loss Penalty	–	5000

Table 2: Comparison of ANFIS RMSE and Loss Metrics for Hyperparameters Across Datasets

Hyperparameter	Amazon		CAC		IBM		Microsoft		Nasdaq		SP500	
	RMSE	Loss	RMSE	Loss	RMSE	Loss	RMSE	Loss	RMSE	Loss	RMSE	Loss
N° M.F.[1]	10	10	10	10	10	10	10	10	10	10	10	10
Input Dimensions	15	15	15	15	15	15	15	15	15	15	15	15
M.F. Parameters	μ, σ	μ, σ	μ, σ	μ, σ	μ, σ	μ, σ	μ, σ	μ, σ	μ, σ	μ, σ	μ, σ	μ, σ
Batch Size	32	32	32	32	32	32	32	32	32	32	32	32
Learning Rate	0.007	0.007	0.008	0.008	0.005	0.01	0.005	0.008	0.005	0.005	0.003	0.005
Loss Penalty		5000	–	10000	–	5000	–	10000	–	10000	–	7000

this instance. Notably, internal factors specific to Amazon, such as the stock split in 2022, likely disrupted ANFIS's ability to fully leverage its modeling capabilities, hindering its performance on this particular dataset.
This indicates that RNNs may still be valuable in scenarios where model interpretability and response to temporal dynamics relations are prioritized.

ANFIS adapted well to our custom loss function, particularly in markets characterized less volatile (indexes) or non-linear behaviour.
This loss function helped RNNs improve in specific datasets like Amazon and Microsoft, suggesting that RNNs can be tuned to enhance their sensitivity to directionality in stock price movements.

Traders might leverage ANFIS for short-term trading where capturing quick, significant movements is more profitable. ANFIS not only maintains high precision but also excels in F1 scores, demonstrating its effectiveness in providing balanced precision and recall, ideal for the more consistent patterns observed in the index data.

RNNs show lower performance but have less "overfitting" and work better in datasets with very large fluctuations, but with higher accuracy low we can have an example with Amazon as it is respectively the most volatile as it is the stock of a single company.

The choice between using RNN and ANFIS models should consider the specific characteristics of the financial dataset, including the market's volatility, the trading horizon, and the complexity of the relationships within the data.
The analysis suggests that while RNNs can be optimized for specific datasets where sequences of historical data strongly predict future trends and depend on each other, ANFIS provides a more consistently reliable model for financial indices due to its ability to manage stability and predictability effectively. This comparative vision should guide the selection and application of appropriate models based on the specific characteristics and volatility of the financial data analyzed.

The findings underscore the importance of selecting the right model and loss function based on the specific requirements of the financial market being analyzed. They also highlight the potential of advanced modelling techniques like ANFIS in enhancing the accuracy and profitability of systematic trading strategies.

5 Conclusions

This study demonstrated the effectiveness of advanced machine learning models, specifically Recurrent Neural Networks (RNN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS), in predictive analysis of stock market trends. Through the application of traditional metrics and a new custom loss function, our research highlights the superior ability of ANFIS to capture complex, nonlinear patterns in financial data, thus offering more accurate and robust pre-

Table 3: Performance Evaluation of Models with RMSE

Dataset	RNN				ANFIS			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Amazon	0.436	0.471	0.423	0.446	0.383	0.392	0.354	0.372
IBM	0.480	0.517	0.426	0.467	0.802	0.819	0.808	0.813
Microsoft	0.440	0.462	0.469	0.466	0.751	0.731	0.833	0.779
CAC	0.527	0.595	0.405	0.482	0.883	0.855	0.946	0.898
Nasdaq	0.601	0.585	0.672	0.625	0.857	0.824	0.910	0.865
Sp500	0.480	0.507	0.492	0.5	0.802	0.744	0.958	0.838

Table 4: Performance Evaluation of Models with Custom Loss Function

Dataset	RNN				ANFIS			
	Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Amazon	0.545	0.573	0.593	0.583	0.425	0.422	0.306	0.355
IBM	0.456	0.491	0.411	0.448	0.824	0.795	0.904	0.846
Microsoft	0.543	0.566	0.515	0.539	0.759	0.719	0.888	0.795
CAC	0.496	0.553	0.376	0.448	0.861	0.85	0.906	0.877
Nasdaq	0.560	0.559	0.540	0.55	0.909	0.898	0.925	0.911
SP500	0.496	0.534	0.343	0.418	0.810	0.752	0.958	0.843

dictions than RNNs. This custom loss function, designed to prioritize the direction of price movements and signs has proven to significantly improve model performance.
The performance of the models varied across different datasets, suggesting that there may be underlying factors specific to each stock or market index that could affect the accuracy of the prediction, such as there are factors that are not predictable, i.e. splits, or Covid or other "natural" and "non-natural" problems as in our opinion they are not possible to predict.

Our study did not delve into the individual characteristics of these data sets nor did it explore the impact of external economic variables that could potentially influence the model results. Therefore, future research should focus on integrating broader economic indicators and conducting a more granular analysis of specific dataset characteristics to better understand and improve the predictive accuracy of the models. Additionally, exploring other machine learning techniques, such as deep learning models that can capture more complex patterns and relationships in data that we have not been able to capture with our knowledge, could provide further insights and improvements to the current predictive framework.

Acknowledgement: We would like to thank ChatGPT and our friends Giuseppe and Phan for helping us accomplish this project.

References

Barak, Sasan, Jalil Heidary Dahooie, and Tomáš Tichý (2015). "Wrapper ANFIS-ICA method to do stock market timing and feature selection on the basis of Japanese Candlestick". In: *Expert Systems with Applications* 42.23, pp. 9221–9235. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2015.08.010. URL: https://www.sciencedirect.com/science/article/pii/S0957417415005497.

Esfahanipour, Akbar and Parvin Mardani (2011). "An ANFIS model for stock price prediction: The case of Tehran stock exchange". In: *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pp. 44–49. DOI: 10.1109/INISTA.2011.5946124.

Felder, Christopher and Stefan Mayer (2022). "Customized Stock Return Prediction with Deep Learning". In: *2022 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*. IEEE, pp. 1–8.

Martucci, Giuseppe (2024). "Benchmarking econometrics and deep learning methodologies for mid-frequency forecasting". In: *Available at SSRN 4773344*.

Prabakaran, N et al. (2021). "Forecasting the momentum using customised loss function for financial series". In: *International Journal of Intelligent Computing and Cybernetics* 14.4, pp. 702–713.