Financial Time Series Analysis using Deep Learning Methods

Team 5

Dheeraj M **Harishankar M** **Kartik Agrawal**
Komaragiri Sai Pranav **Hiya Mehta**

Predicting the Market

- Finance industry prioritizes accurate forecasting of financial time series data to maximize profits
- Traditional methods such as ARIMA, ARCH, GARCH are have limited applicability :
  - Assumes stationarity of data.
  - Static models cannot adapt to new data in real-time.
  - Fail to integrate external features like sentiment or
  - macroeconomic indicators.

- Machine Learning (ML) especially Deep learning models outperforms traditional time series methods

Prediction Problem

- Main Objective: Predict the next movement of an asset.
- Two Prediction Types:
  - Price Prediction: Regression problem.
  - Trend Prediction: Classification problem.

Challenges in Real Time Prediction

- Huge amounts of financial data is generated every second
  - Requires ongoing learning without forgetting historical knowledge.
  - Retraining with all data is computationally infeasible.
- The properties of incremental data varies with market conditions
- Thus prediction model's effectiveness diminishes if used without adaptation.

Continual Learning strategies

- Overcomes real-time deployment challenges by enabling:
  - Incremental learning of tasks while retaining past knowledge.
  - Adaptation to dynamic data distributions.
- Addresses catastrophic forgetting (loss of prior knowledge).
- Balances stability and plasticity to ensure adaptability and retention.

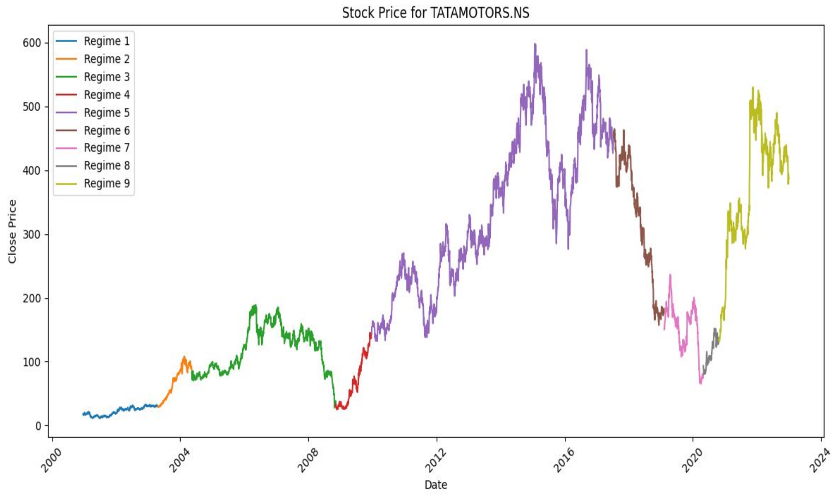


Fig: Market Regime

Problem Framework

- **Aim:** Prediction of future stock prices (or trends) based on historical stock data
- The training input is the closing price of asset for the past d time steps and the label is the closing price of the stock in the next timestep .
- For analyzing the performance of the model, we use regression specific metrics such as R2 score, RMSE.
- Specifically for the continual learning strategy we experiment with a simple investment strategy to analyze the profit percentage after investing using our prediction.
- Investment strategy:
  - We predict the next timestep’s stock price using our model
  - Buy, sell, or retain the stock depending on the models prediction of price is more, less or no change
  - Calculating the profit across the testing dataset and calculating profit percentage with respect to the initial investment.

Literature Review

- Traditional statistical methods, such as autoregressive models [1], hidden Markov models [2] and tree models[3].
- RNNs use feedback mechanisms to simulate long and short term memory, widely used for stock prediction.[4][5][6]
- Hybrid models of CNN-LSTM[7], LSTM-ARIMA[8], LSTM-GRU[9] have outperformed individual models various times in stock price prediction.
- Transformers[10] have been used for stock prediction[11] and to forecast S&P volatility[12] due to their strong modelling abilities to capture long-time dependencies in sequential data.
- Muhammad et al. [13] achieved low error rates and effective price predictions by utilizing time2vec encoding in their transformer model.
- By using market data for feature selection and stock correlation, MASTER [14] achieved improvement in ranking metrics by 13% and portfolio metrics by 47%.

Literature Review (Continued)

Continual Learning methodologies

- A thorough study of current trends in the application of continual learning techniques for time series analysis has been given by Ragot et al. [15].
- In order to make better investment decisions, Philips [16] has generally employed continuous learning.
- A comparative analysis of continuous learning techniques in financial time series was conducted by Ramjattan et al. [17].
- Elastic Weight Consolidation (EWC) was introduced as a regularization technique for continual learning by Kirkpatrick et al. [18].
- Learning Without Forgetting (LWF) was proposed as a regularization-based continual learning method by Li and Hoiem [19].
- GDumb was introduced as a replay-based baseline for continual learning by Prabhu et al. [20].

Implementation

For Comparative Study:

- ○ ***Setting***: Supervised Learning Task with different lookback windows
- ○ ***Non-Recursive Forecasting:*** Predicting next day by taking actual (previous) test data.
- ○ ***Recursive/Iterative Forecasting:*** Predicting a longer period of time by iteratively feeding predictions as input.

- **Dataset:** YFinance Library of Python (Stocks Used: Tata Motors and Bitcoin)
- **Output:** Closing prince of next day of the input batch
- **Models:** LSTM, GRU, Transformer and Hybrid
- **Inputs:**
  - Training: Previous 10 years of closing price of TATA Motors stocks..
  - Lookback windows used: 30, 60, 90
- **Loss:** Mean Squared Error
- **Optimizer:** Adam

For Continual Models:

- **Dataset**: Bitcoin price (every 5 min for past 60 days)
- **Model**: LSTM
- **Lookback**: 60 timesteps
- **Loss:** Mean squared Error
- **Optimizer:** Adam

Continual Learning Methodologies

- **Naive Training**
  - Approach: The model is trained sequentially on new data, overwriting previous knowledge.
  - Limitation: Prone to catastrophic forgetting, as past data is not considered.
- **Joint Training**
  - Approach: Combines all past and new data for retraining.
  - Strengths: Preserves knowledge from all tasks.
  - Limitation: Computationally expensive and requires significant storage.
- **Re-incremental Training**
  - Approach: Train a new model only on the incremental data
  - Strength: Lesser training complexity due to smaller training size
  - Limitations: Training only on the incremental data may fail to capture historic trends.
- **Elastic Weight Consolidation (EWC):**
  - Type: Regularization-Based Algorithm
  - Purpose: Prevents catastrophic forgetting by preserving critical parameters.
  - Key Concept: Adds a regularization term using the Fisher Information Matrix.
  - Strengths: Balances old and new data effectively.
- **Learning Without Forgetting (LWF):**
  - Type: Regularization-Based Algorithm
  - Purpose: Retains knowledge of past tasks using soft targets and knowledge distillation.
  - Strengths: No explicit storage of old data, suitable for sequential learning
- **GDumb :**
  - Type: Replay-Based Algorithm
  - Purpose: Simple baseline method that retrains on past data in a limited memory buffer
  - Strengths: Easy to implement.
  - Limitation: High memory and retraining overhead.

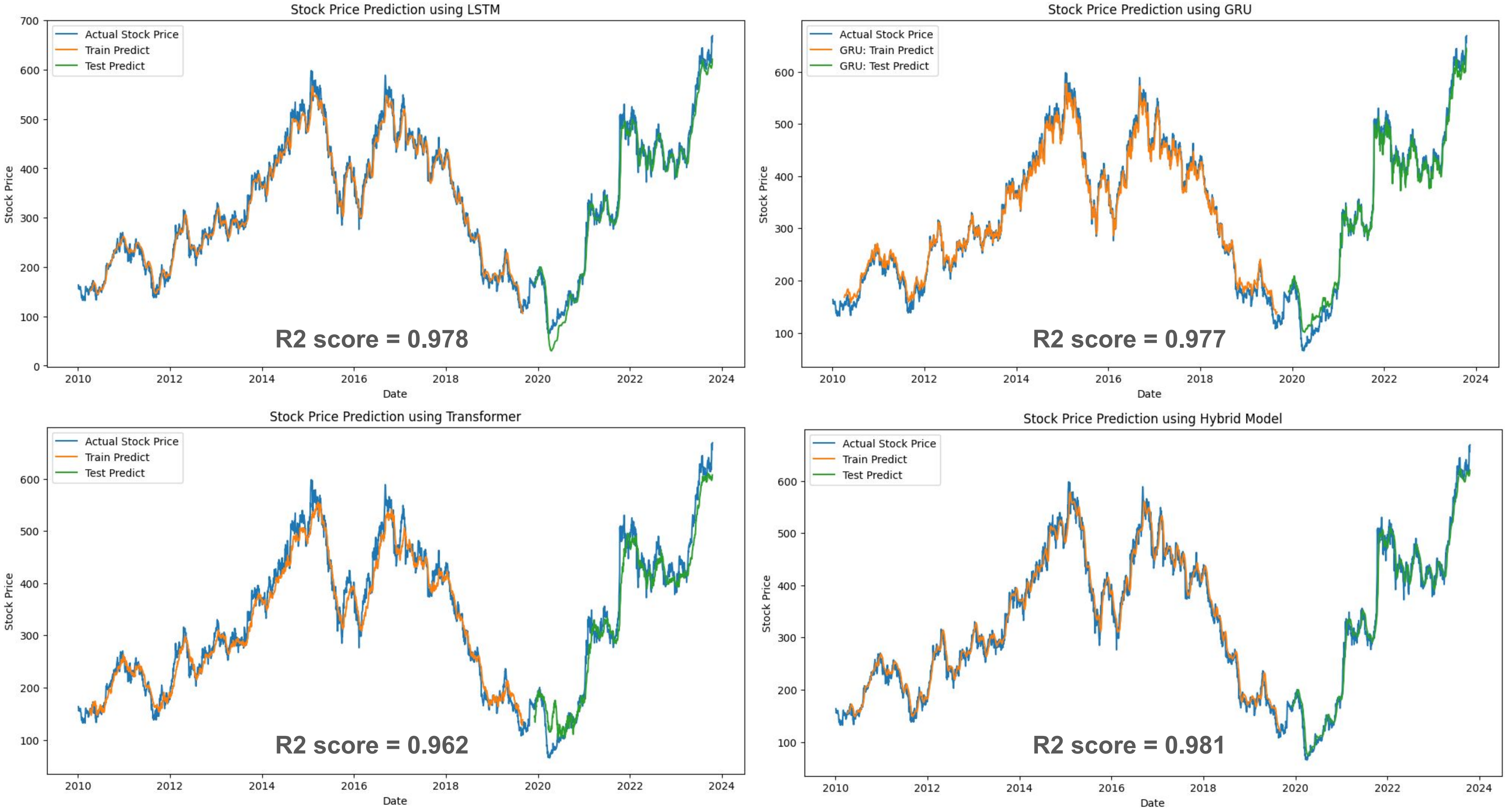
Model Architectures

- **GRU Model:**
  - **3 stacked GRU layers** (16 units each) to capture time dependencies,
  - A dropout layer (20%) to prevent overfitting
  - A Dense layer for final regression output.
  - Total Parameters = **4,193**
- **LSTM Model:**
  - **2 LSTM layers** (10 and 20 units) to extract temporal patterns, interleaved with dropout layers (20% and 30%) for regularization,
  - Dense layer for prediction.
  - Total Parameters = **2,981**
- **Transformer Model:**
  - Encoder block with **4 heads**(head_size = 256), feed-forward layers, and residual connections,
  - Followed by global pooling and Dense layers for regression output.
  - Total Parameters = **7,827**
- **Hybrid Model:**
  - A combination of LSTM and GRU,
  - starting two layers are stacked LSTMs(10 and 20 units) for sequence learning,
  - Followed by a GRU layer(10 units) for further temporal abstraction, regularized with dropout, and ending with a Dense layer for output.
  - Total Parameters = **3,931**

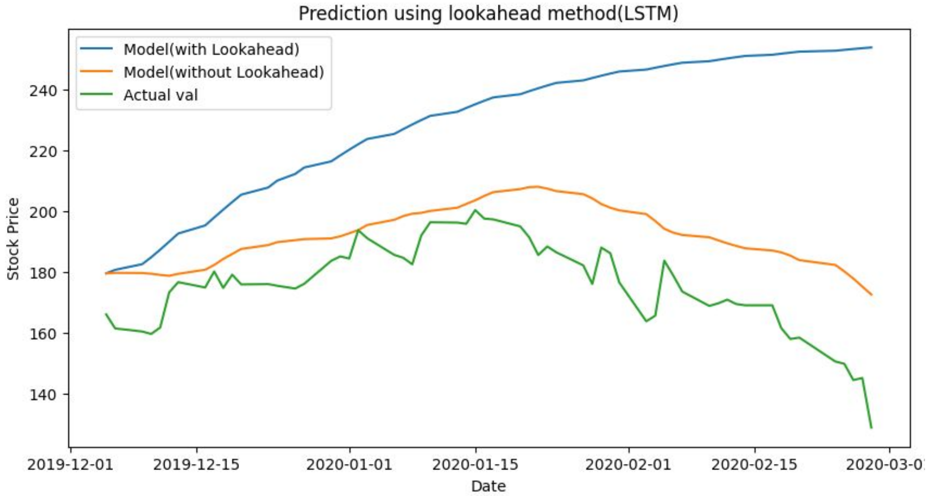
Results: Comparative Study

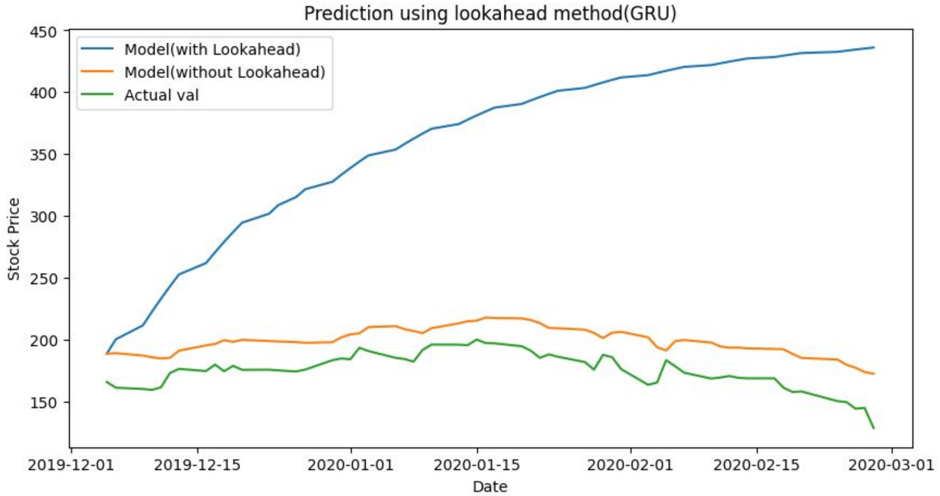
Parameters	LSTM (Params= 4,193)			GRU (Params= 2,981)			Transformer (Params= 7,827)			Hybrid (Params= 3,931)		
Lookback Window (days)	30	60	90	30	60	90	30	60	90	30	60	90
RMSE	18.938	23.986	23.782	20.368	24.079	24.885	31.824	31.054	30.30	24.509	21.765	21.765
R2 Score	0.986	0.978	0.977	0.984	0.977	0.975	0.960	0.962	0.964	0.977	0.981	0.981
Time/epoch (s)	1.65	3.43	4.27	2.41	7.41	9.64	0.96	9.81	19.34	2.18	9.53	10.12
Epochs	50	50	50	50	50	50	50	50	50	50	50	50

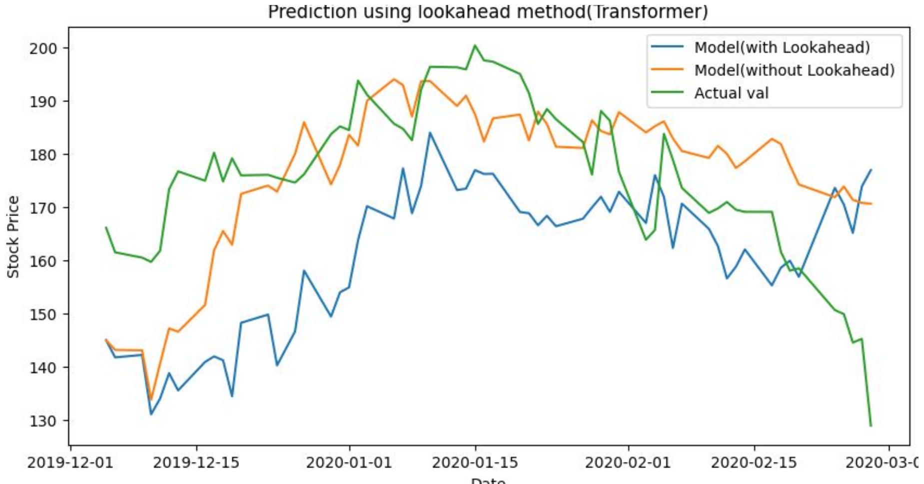
Results: Comparative Study for Non-Reccursive Prediction (Lookback =60)

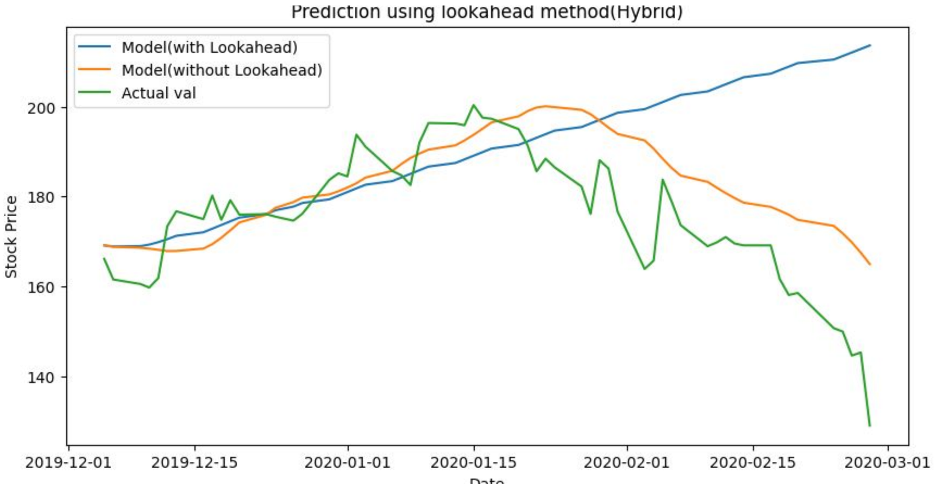


Results: Comparative Study for Reccursive Prediction (Lookback =60)









Results: Continual Learning Strategies

Model Type	R2 Score	RMSE	Profit (\%)
EWC	0.814	386.177	-1.296
LWF	0.493	609.309	-9.115
Gdumb	0.81	353.173	-5.339
Naive Training	-0.102	842.609	0.548
Joint Training	0.902	280.363	3.542
Re-increment Training	-13.991	3574.31	-6.487

Baseline Performance (29/09/24 - 29/10/24)

Model Type	R2 Score	RMSE	Profit (\%)
EWC	0.864	1002.053	-6.206
LWF	-6.875	2291.936	-1.073
Gdumb	-0.991	1160.059	-1.415
Naive Training	0.81	1186.731	-6.737
Joint Training	0.789	1252.121	-4.817
Re-increment Training	-0.196	2980.473	-4.481

Increment 1 (29/10/24 - 08/11/24): Data Performance

Model Type	R2 Score	RMSE	Profit (\%)
EWC	0.964	181.57	3.1
LWF	0.577	350.195	1.458
Gdumb	0.529	222.44	0.213
Naive Training	0.384	754.77	-1.389
Joint Training	-49.609	6840.4	-2.615
Re-increment Training	0.1445	889.36	0.028

Increment 2 (08/11/24 - 18/11/24): Data Performance

Model Type	R2 Score	RMSE	Profit (\%)
EWC	0.991	200.405	6.118
LWF	0.961	274.636	3.122
Gdumb	0.936	353.083	0.908
Naive Training	0.963	400.199	-2.726
Joint Training	0.869	752.47	8.042
Re-increment Training	0.409	1592.585	2.402

Increment 3 (18/11/24 - 28/11/24): Performance

Learnings/Conclusion

- Deep Learning has significantly advanced financial time series forecasting, with extensive research focused on improving prediction models by considering other real-time factors like sentiment analysis.
- Stock price prediction is inherently complex as its a function of various factors like global markets, politics, and social media.
- In this work, we focused on predicting stock prices only using historical stock closing price data to achieve a high prediction accuracy
- Literature survey revealed that most of the implementations were based on LSTMs, GRUs & Transformers.
- Having studied these models in-depth during class lectures, it was much easier for us to follow and understand their implementation.

Learnings/Conclusion

- We also learnt about the implementations of LSTMs, GRUs, Transformers and Hybrid models in PyTorch.
- Experimented with lookahead based predictions
- Explored the usage of continual learning and its benefits in financial time series forecasting.
- We implemented several continual learning strategies and noticed the improvement in the prediction accuracy.

Future Work:

- Usage of sentiment analysis(NLP) and intra-stock correlations (Using GNNs) in the models to achieve realistic predictions.
- Investment Strategy based on predicted trends by bagging of trend predictions of multiple models.

Individual Contributions

- Harishankar M: Continual Learning
- Kartik Agrawal: LSTMs, Transformers
- Hiya Mehta: Transformers, Hybrid Model
- Sai Pranav K: GRUs, GNNs(experiment)
- Dheeraj M: Continual Learning

Citations

[1] Chun Shan Wong and Wai Keung Li. On a mixture autoregressive model. Journal of the Royal Statistical Society Series B: Statistical Methodology, 62(1):95–115, 2000.

[2] Md Rafiul Hassan and Baikunth Nath. Stock market forecasting using hidden markov model: a new approach. In 5th international conference on intelligent systems design and applications (ISDA'05), pages 192–196. IEEE, 2005.

[3] Ernest Kwame Ampomah, Zhiguang Qin, and Gabriel Nyame. Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement Information, 11(6):332, 2020.

[4] Nesma M. Rezk, Madhura Purnaprajna, Tomas Nordstr¨om, and Zain Ul-Abdin. Recurrent neural networks: An embedded computing perspective. IEEE Access, 8:57967–57996,2020.

[5] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. Financial time series forecasting with deep learning : A systematic literature review: 2005–2019. Applied Soft Computing, 90:106181, 2020

[6] Gaurang Sonkavde, Deepak Sudhakar Dharrao, Anupkumar M. Bongale, Sarika T. Deokate, Deepak Doreswamy, and Subraya Krishna Bhat. Forecasting stock market prices using machine learning and deep learning models: A systematic review, performance analysis and discussion of implications. International Journal of Financial Studies, 11(3), 2023.

Citations

[7] Xinrong Zhou. Stock price prediction using combined lstm-cnn model. In 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), pages 67–71. IEEE, 2021.

[8] Chenxi He. A hybrid model based on multi-lstm and arima for time series forcasting. In 2023 8th International Conference on Intelligent Computing and Signal Processing (ICSP), pages 612–616. IEEE, 2023.

[9] A. Bhavani, A. Venkata Ramana, and A. S. N. Chakravarthy. Comparative analysis between lstm and gru in stock price prediction. In 2022 International Conference on Edge Computing and Applications (ICECAA), pages 532–537, 2022.

[10] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey, 2023.

[11] Linyi Yang, Tin Lok James Ng, Barry Smyth, and Riuhai Dong. Html: Hierarchical transformer-based multi-task learning for volatility prediction. In Proceedings of The Web Conference 2020, page 441–451, New York, NY, USA, 2020. Association for Computing Machinery.

[12] Eduardo Ramos-P´erez, Pablo J. Alonso-Gonz´alez, and Jos´e Javier N´u˜nez-Vel´azquez. Multi-transformer: A new neural network-based architecture for forecasting sp volatility. Mathematics, 9(15), 2021.

Citations

[13] Tashreef Muhammad, Anika Bintee Aftab, Muhammad Ibrahim, Md. Mainul Ahsan, Maishameem Meherin Muhu, Shahidul Islam Khan, and Mohammad Shafiul Alam. Transformer-based deep learning model for stock price prediction: A case study on bangladesh stock market. International Journal of Computational Intelligence and Applications, 22(03):2350013, 2023.

[14] Tong Li, Zhaoyang Liu, Yanyan Shen, Xue Wang, Haokun Chen, and Sen Huang. Master: Market-guided stock transformer for stock price forecasting. Proceedings of the AAAI Conference on Artificial Intelligence, 38(1):162–170, 2024.

[15] Quentin Besnard and Nicolas Ragot. Continual learning for time series forecasting: A first survey. Engineering Proceedings, 68(1), 2024.

[16] Daniel Philps. A temporal continual learning framework for investment decisions. PhD thesis, City, University of London, 2020.

[17] Reshawn Ramjattan, Daniele Atzeni, and Daniele Mazzei. Comparative evaluation of continual learning methods in financial and industrial time-series data. In 2024 International Joint Conference on Neural Networks (IJCNN), pages 1–7. IEEE, 2024.

Model Architectures

GRU Model:

Layer(type)	Output Shape	Param #
GRU	(None, 60, 16)	912
GRU_1	(None, 60, 16)	1,632
GRU_2	(None, 16)	1,632
Dropout	(None, 16)	0
Dense	(None, 1)	17

Total Parameters: 4,193 (16.38 KB)
Trainable Parameters: 4,193 (16.38 KB)
Non-Trainable Parameters: 0 (0.00 B)

LSTM Model:

Layer(Type)	Output Shape	Param #
LSTM(lstm)	(None, 60, 10)	480
Dropout (dropout_5)	(None, 60, 10)	0
LSTM (lstm_1)	(None, 20)	2,480
Dropout (dropout_6)	(None, 20)	0
Dense (dense_5)	(None, 1)	21

Total Parameters: 2,981 (11.64 KB)
Trainable Parameters: 2,981 (11.64 KB)
Non-Trainable Parameters: 0 (0.00 B)