



**HAL**  
open science

# **A Comparative Analysis of Liquid Neural Networks and Incremental Learning Approaches for Stock Market Prediction**

Hussein Jammal, Farah Srour, Ali Salman, Michel Owayjan, Omran Ayoub,  
Cristina Rottondi, Roger Achkar

## **► To cite this version:**

Hussein Jammal, Farah Srour, Ali Salman, Michel Owayjan, Omran Ayoub, et al.. A Comparative Analysis of Liquid Neural Networks and Incremental Learning Approaches for Stock Market Prediction. 2025 International Conference on Control, Automation, and Instrumentation (IC2AI), Feb 2025, Beirut, Lebanon. pp.1-6, <10.1109/IC2AI62984.2025.10932165>. <hal-05148958>

**HAL Id: hal-05148958**

**<https://hal.science/hal-05148958v1>**

Submitted on 7 Jul 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# A Comparative Analysis of Liquid Neural Networks and Incremental Learning Approaches for Stock Market Prediction

Hussein Jammal<sup>1,4</sup>, Farah Srour<sup>1</sup>, Ali Salman<sup>1</sup>, Michel Owayjan<sup>1,2</sup>, Omran Ayoub<sup>3</sup>, Cristina Rottondi<sup>4</sup>, Roger Achkar<sup>1</sup>

<sup>1</sup>Faculty of Engineering and Computer Science, American University of Science & Technology, Beriut, Lebanon

<sup>2</sup>Institut de Recherche en Energie Electric de Nantes Atlantique (IREENA), Nantes University, Saint-Nazaire, France

<sup>3</sup>University of Applied Sciences of Southern Switzerland, Switzerland

<sup>4</sup>Politecnico di Torino, Italy

HAI20010@students.aust.edu.lb, FYS20008@students.aust.edu.lb, ahs20045@students.aust.edu.lb, mowayjan@aust.edu.lb,

omran.ayoub@supsi.ch, cristina.rottondi@polito.it, rachkar@aust.edu.lb

**Abstract**—Due to the complexity, challenge, and fast growth of stock markets, they encourage more efficient use of financial resources and the expansion of macroeconomics. Tesla and Apple stock prices fluctuate constantly due to the impact of various industries and market conditions. The extraction of previously unknown patterns and information from time series is central to numerous real-world applications. This paper presents a comparative study of stock market prediction techniques for Tesla and Apple stocks, focusing primarily on the application of Liquid Neural Networks (LNN) and evaluating their performance against Incremental Learning methods. Employing LNN, a novel approach in the field of time-series forecasting, we developed models that leverage the dynamic and flexible structure of LNN to adapt to the complex and non-linear patterns observed in stock prices. The performance of the LNN model was quantitatively assessed using standard metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Directional Accuracy. The LNN model achieved an MSE of 0.000317, RMSE of 0.0178, MAE of 0.014131, MAPE of 1.8%, and a Directional Accuracy of 49.36%. In contrast, the Incremental Learning approach yielded an MSE of 0.54355, RMSE of 0.73725, MAE of 0.73158, MAPE of 89.39%, and Directional Accuracy of 28.37%. These results underscore the superior accuracy and effectiveness of LNNs in modeling stock market behaviors compared to traditional Incremental Learning methods. The findings not only demonstrate the potential of LNNs in financial analytics but also suggest avenues for further research into enhancing predictive accuracy and model robustness.

**Keywords**— *Liquid Neural Networks (LNN), Incremental Learning, Stock Market Prediction, Time-Series Forecasting, Financial Analytics, Deep Learning, Market Behavior Analysis, Predictive Modeling, Tesla Stock, Apple Stock.*

## I. INTRODUCTION

The stock market plays an important driver of economic growth. It has attracted both individual and institutional investors because of the potential for high returns. However, its complex and often unpredictable nature can result in considerable losses. By anticipating and projecting shifts in stock prices, investors may be able to reduce their exposure to risk.

In recent years, research into applying machine learning to predict stock performance has made great strides. Within the machine learning domain, algorithms build mathematical models based on sample data—often called “training data”—to generate forecasts or decisions. This approach can be seen in predicting movements for companies like Tesla and Apple [1]. Algorithms examine large amounts of data, discovering subtle market patterns and predicting future stock values. Researchers employ a variety of criteria, including historical data and temporal trends, to test current ideas and find patterns in the stock market. Studies use machine learning techniques like supervised learning to predict stock prices, as well as methodologies like deep learning and non-linear regression.

Incremental learning is a machine learning methodology where a model continuously learns new information over time, enhancing its performance without retraining from scratch. In stock market prediction, incremental learning techniques adapt to evolving market conditions, making them particularly useful for predicting volatile assets. Specifically, in stock market prediction, incremental learning algorithms can adjust to real-time data streams, capturing sudden market shifts and complex patterns [2]. This adaptability enables more accurate and timely predictions, which is crucial for investors navigating dynamic market environments.

Liquid Neural Networks (LNNs) represent a novel approach to prediction tasks, using their time-continuous recurrent architecture to process sequential data dynamically. In prediction processes, LNNs exhibit remarkable capabilities owing to their unique characteristics. They offer adaptability and fast adjustment to changing data patterns, thus enhancing prediction accuracy and robustness. This dynamic adaptability enables LNNs to effectively capture temporal dependencies in the data, resulting in more precise predictions across various domains such as weather prediction, speech recognition, and autonomous driving. LNNs offer several advantages in prediction processes. The characteristic architecture of LNNs allows for real-time adjustments to changing data patterns, ensuring robust performance even in dynamic environments. Thus, LNNs emerge as a powerful tool for predictive analytics, offering enhanced adaptability, efficiency, and accuracy.

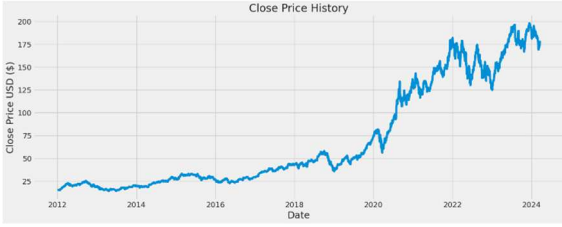


Fig. 1. Stock Market Close Price History

A Liquid Neural Network (LNN) learns from historical market data—such as stock prices, trading volumes, and economic indicators—by capturing patterns over time and continuously updating its internal states to adapt to new information. This adaptability is essential in fast-changing stock markets like Tesla and Apple, which exhibit high volatility. In this study, LNNs are compared with Incremental Learning models using a range of metrics (MSE, RMSE, MAE, MAPE, and Directional Accuracy) to assess how effectively each approach predicts stock prices. By examining these methodologies side-by-side, the paper reveals key insights into their capabilities, limitations, and overall potential for financial market forecasting.

The rest of the paper is organized as follows. Section II discusses related work in the field of stock market prediction and the specific technologies of LNN and Incremental Learning. Section III describes the methodology, including data collection, model architecture, and evaluation. Section IV discusses the experimental results. Section V concludes with a summary of findings and potential areas for future research.

## II. LITERATURE REVIEW

### A. Incremental Learning

Incremental Learning can follow two main approaches. The first is a purely incremental method, where the model gets updated with each newly acquired stock price from the data stream. The second is an Offline-Online method, where the model is retrained after every trading session. For the incremental approach, Incremental Linear Regression has been employed, while the Offline-Online approach typically uses LSTM and CNN variants [3]. In the Offline-Online process, “offline” refers to analyzing a batch of data to refine the model before generating predictions, whereas “online” involves continuously pulling data from the stream and creating predictions. By contrast, the goal of purely incremental learning is to adapt the model to new information while preserving what it has already learned. Unlike incremental models, the Offline-Online approach does not adjust after every single piece of new data but waits until the close of each trading session—recognizing that the market can be influenced by multiple factors within that timeframe. Incremental learning, on the other hand, updates with every incoming data point, eliminating the need to retrain using the entire dataset [4].

Implementing an incremental learning model typically involves several stages: data gathering, data processing, feature selection, and then applying the incremental learning algorithm itself (illustrated in Fig. 2). Recent developments have combined deep learning with Incremental Learning to

more effectively handle real-time changes. For instance, Singh et al. (2023) [5] demonstrated how deep networks can adapt dynamically to tasks, a concept directly applicable to adjusting financial models for market fluctuations. In the financial world, Incremental Learning has been used to continuously refine forecasts as new market data streams in, helping models remain robust over time.

### B. Liquid Neural Network (LNN)

In 2020, a group of MIT scientists produced a study titled “Liquid Time-constant Network” in which they introduced the concept of liquid neural networks. This concept is influenced by brain research. They investigated the underlying computation of a tiny worm, *C.elegans*, whose nervous system has just 302 neurons yet generates adaptive dynamics. Liquid neural networks are exciting because they can adapt to changing situations, rather than simply during training. They are especially effective for processing time series data, making them ideal for applications like autonomous driving and medical diagnostics.

Liquid Neural Networks (LNN) have emerged as a promising new frontier. It is a time-continuous system that processes data sequentially, remembers previous inputs, modifies its behaviors depending on new inputs, and can handle variable-length inputs to improve NNs’ task-understanding skills. LNN design varies from typical neural networks in that it can efficiently process continuous or time series data. If fresh data becomes available, LNNs can adjust the number of neurons and connections per layer. Liquid Neural Networks (Fig. 3) are designed specifically for time series data processing and prediction [6].

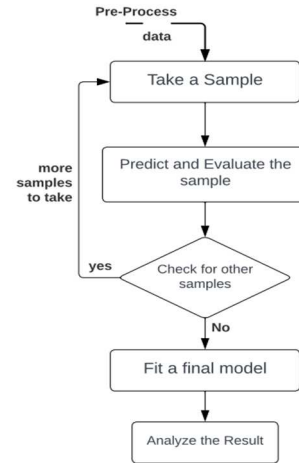


Fig. 2. Incremental Learning Process Flowchart

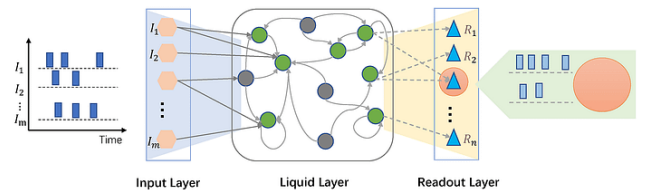


Fig. 3. Liquid Neural Network Architecture

$$\frac{dx(t)}{dt} = - \left[ \frac{1}{\tau} + f(\mathbf{x}(t), \mathbf{I}(t), t, \theta) \right] \mathbf{x}(t) + f(\mathbf{x}(t), \mathbf{I}(t), t, \theta)A \quad (1)$$

The above equation is derived by (Funahashi and Nakamura 1993) and may be used to determine a more stable continuous-time recurrent neural network:

$$(\text{CT-RNN}): \frac{dx(t)}{dt} = - \frac{x(t)}{\tau} + f(\mathbf{x}(t), \mathbf{I}(t), t, \theta) \quad (2)$$

where the phrase  $-\frac{x(t)}{\tau}$  helps the autonomous system attain an equilibrium state with a time constant  $\tau$ .  $\mathbf{x}(t)$  is the hidden state,  $\mathbf{I}(t)$  is the input,  $t$  represents time, and  $f$  is parametrized by  $\theta$ .

Offering an alternative formulation: let the hidden state flow of a network be proclaimed by a system of linear ODEs of the form:

$$\frac{dx(t)}{dt} = - \frac{x(t)}{\tau} + S(t) \quad (3)$$

Let  $S(t) \in \mathbb{R}^M$  represent the following nonlinearity specified, with parameters  $\theta$  and  $A$  [6].

$$S(t) = f(\mathbf{x}(t), \mathbf{I}(t), t, \theta)(A - \mathbf{x}(t)), \quad (4)$$

### C. Related Work

For the past 35 years, we have developed probabilistic models that make predictions based on data and learned parameters. Each neuron acts as a logistic regression unit. By using backpropagation, which adjusts the neuron's parameters to minimize errors, the network learns and improves its predictions. Neural networks face certain challenges today; they excel in unified tasks, but cannot generalize knowledge across tasks. Additionally, they process data non-sequentially, making them inefficient at handling real-time data.

Stock market prediction has been a longstanding area of both scholarly and real-world interest, bringing together economic principles and sophisticated computational methods. A landmark contribution from Thakkar and Chaudhari (2021) [7] on time series analysis has driven numerous follow-up studies in market forecasting techniques. More recently, machine learning approaches have been adopted to boost predictive accuracy and manage ever-growing datasets.

Neural networks, in particular, have drawn considerable attention in the realm of stock market prediction. For instance, Zhang et al. (2017) [8] showed how deep learning could outperform traditional statistical techniques in forecasting stock prices. Their study was instrumental in revealing the power of neural networks to capture the complex behavior of financial markets.

Meanwhile, more recent research has pivoted toward cutting-edge neural network models and hybrid approaches. One example is Dixon et al. (2016) [9], who combined machine learning algorithms with quantitative finance models to predict stock returns, shedding light on the fusion of

established financial theory with modern computational insights.

## III. ARCHITECTURE & METHODOLOGY

This section outlines the methodology employed in this study, encompassing data collection, exploratory data analysis (EDA), feature engineering, model architecture, and evaluation metrics. Our objective is to leverage historical stock data from Yahoo Finance for Apple (AAPL) and Tesla (TSLA) to predict future stock prices using advanced machine learning techniques.

### A. Data Collection

The dataset consists of daily stock prices for AAPL and TSLA retrieved from Yahoo Finance using the Python library **yfinance**. The data spans from January 1, 2010, for AAPL, and from June 29, 2010, for TSLA, up to the day prior to the current date. Each dataset includes the following features: Open, High, Low, Close, Adjusted Close, and Volume.

### B. Exploratory Data Analysis (EDA)

Initial data exploration was conducted to understand underlying patterns and distributions:

1. Candlestick Plot with Moving Averages (20, 50, 100 days): Visualizes price movements over time alongside key moving averages.
2. Correlation Heatmap: Assesses the relationships between different stock attributes.
3. Box Plot: Identifies outliers within the dataset.
4. Histogram of Daily Price Changes: Examines the distribution of daily price fluctuations.

### C. Feature Engineering

To enhance model performance, several technical indicators were computed, resulting in a total of 28 attributes [10]. Table I presents the feature engineering done on the model.

The equations for each attribute are found in [11].

TABLE I. FEATURE ENGINEERING DONE ON THE MODEL

Attributes		Target Output
Open, High, Low, Close, Volume	14 & 21 Day Moving Average Convergence Divergence	Adjusted Close
Daily Returns	Pivot Point	
5-Day Momentum	On Balance Volume	
14-Day Bollinger Bands	14-Day Average True Range	
14-Day Fast, Slow & Smoothed Slow Stochastic Indicators	7, 14 & 21 Day Up & Down Trending Fibonacci Retracement at 38.2, 50 & 61.8%	
Past 8 Weekly & Past 2 Monthly Returns	3 Day Rate of Change	
14 Day Simple & Exponential (k=2/5) Moving Average	14-Day Relative Strength Index	

Normalization was applied using MinMaxScaler to scale all input features to a [0, 1] range, facilitating model convergence and improving training stability. The MinMaxScaler method is chosen for the normalization of the input dataset, and the formula is as follows [12]:

$$X_{new} = \frac{X_{original} - X_{min}}{X_{max} - X_{min}} \quad (5)$$

#### D. Dataset Construction

The training and testing datasets are constructed using a **look\_back** parameter of 10 days, creating sequences from the time series data that represent the past 10 days of trading to predict the next day's adjusted close price [13].

The dataset is split into 80% for training and 20% for testing to evaluate the models' performance.

#### E. Model Architecture

##### 1) Liquid Neural Networks (LNN)

The LNN model utilized the LTCCell class, configured with 150 units incorporating dropout and regularization (L1 and L2) to prevent overfitting and enhance generalization [14].

##### 2) Incremental Learning (MLP)

The MLP architecture consisted of three dense layers with 256, 128, and 64 units respectively, followed by a dropout layer and a final dense layer outputting the predicted adjusted close price. The activation function used for the hidden layers is the **Relu** activation function.

#### F. Evaluation Metrics

The models were assessed using the following metrics:

1. Mean Squared Error (MSE): Measures how predicted values deviate from actual values by squaring the differences and then taking the average. Because of the squaring, bigger mistakes get penalized more than smaller ones. A lower MSE means the model is better in fitting the data.

$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (6)$$

2. Root Mean Squared Error: RMSE is the square root of the MSE, which keeps it in the same units as the original data and makes it easier to interpret. Like MSE, it also emphasizes larger errors more heavily. A lower RMSE indicates stronger model performance.

$$\sqrt{\sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}} \quad (7)$$

3. Mean Absolute Error (MAE): MAE calculates the average absolute difference between predicted and actual values. It gives all differences the same weight, making it a straightforward measure of prediction accuracy. A smaller MAE indicates a more accurate model.

$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (8)$$

4. Mean Absolute Percentage Error (MAPE): MAPE takes the average absolute percentage difference between actual and predicted values, showing accuracy as a percentage. This makes it handy when comparing performance across different scales. Still, it can be less reliable if actual values are very small or zero because the percentage calculation can inflate errors in those cases.

$$\frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (9)$$

5. Directional Accuracy: Directional Accuracy checks whether the model correctly predicts whether values will go up or down, regardless of how much they change. A higher score here means the model is better at capturing the correct trend or movement direction [15].

$$\frac{1}{n} \sum_t 1_{\text{sign}(y_t - y_{t-1}) == \text{sign}(\hat{y}_t - y_{t-1})} \quad (10)$$

Where,  $y_i$  and  $\hat{y}_i$  in each metric indicate the actual price and the predicted price at time  $t$ , respectively;  $n$  represents the number of observations in the dataset.

## IV. RESULTS & EXPLANATION

### A. User Interface

A user interface was developed for the stock market prediction application. The application leverages Liquid Neural Network (LNN) and Incremental Learning techniques to predict stock prices [16].

### B. Results

Fig. 5 illustrates how the Liquid Time-Constant (LTC) model forecasts closing prices for both Apple and Tesla. Its strong performance in projecting different stock values is evident in the close alignment between predicted outcomes and actual historical data during both training and testing. This indicates that the model is well-tuned and capable of delivering accurate market predictions. These findings emerge as a key takeaway from the Liquid Neural Network (LNN) discussion.

The model also captures the overall trend in the testing phase but exhibits some inaccuracies during volatile periods. These results indicate that the LTC model is effective in predicting general trends in stock prices. This analysis aligns with the objectives and findings presented in the Liquid Neural Network (LNN) presentation.

### C. Common Results

- X-axis (Time Steps): Represents the sequential time steps, which could correspond to days, weeks, or any other time interval used in the dataset.
- Y-axis (Adjusted Close Price): Represents the adjusted closing price of stock.
- Blue Line (True Adj Close): This line shows the actual historical adjusted closing prices of each stock.

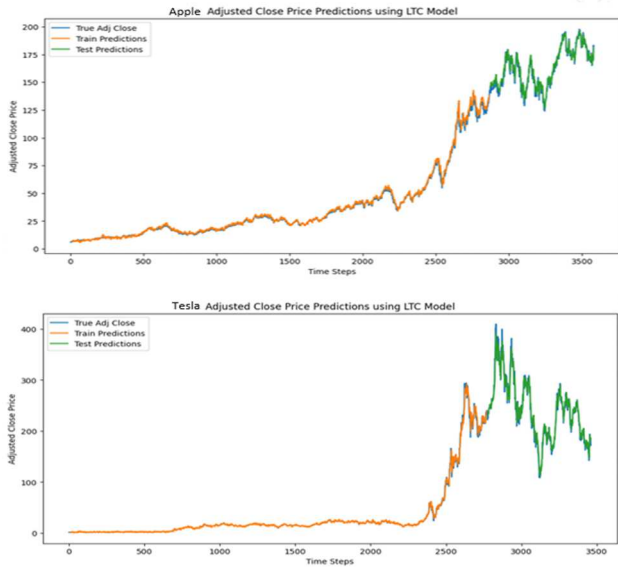


Fig. 4. Apple & Tesla Stock Market Predictions

- Orange Line (Train Predictions): This line shows the model's predictions for the training dataset, indicating how well the model fits the historical data it was trained on.
- Green Line (Test Predictions): This line shows the model's predictions for the testing dataset, which is used to evaluate the model's performance on unseen data.

#### D. Loss Function

Figure 6 shows the training and validation loss for models that predict Apple and Tesla's adjusted closing prices, tracked over multiple epochs of training. Both models clearly learn effectively, as evidenced by the sharp decline in loss from start to finish. The close alignment between training and validation losses signals strong generalization, indicating the models are not simply memorizing the training data. Across both datasets, the graphs reveal low and stable losses, confirming that the models have been well-trained. While Tesla's higher volatility poses slightly more challenges, the overall results suggest both models are robust for forecasting stock prices.

#### E. Evaluation Metrics

Table II evaluates the performance of two predictive models—LTC Model, and Incremental Learning—on Tesla's stock market metrics. Across all evaluated metrics, the LTC Model significantly outperforms the others, demonstrating the lowest Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), which indicates it predicts having a minimal error. It also has the lowest Mean Absolute Percentage Error (MAPE) at 4.87%, showing highly accurate percentage predictions. Additionally, it achieves the highest Directional Accuracy (DA) at 51.25%, making it the most effective model in predicting the direction of stock price movements.

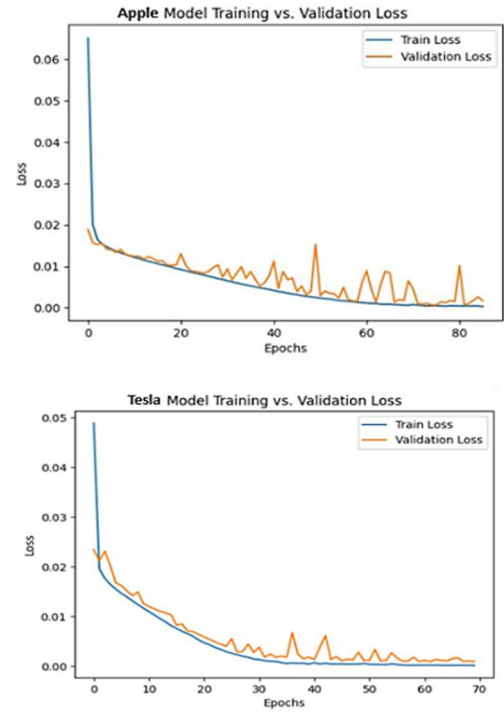


Fig. 5. Apple & Tesla Model Training Loss

Table III summarizes the performance metrics of two models used for predicting apple prices. The LTC Model excels in most metrics, demonstrating the lowest Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), indicating highly accurate predictions with minimal errors in both magnitude and percentage terms.

TABLE II. EVALUATION METRICS FOR TESLA STOCK MARKET

Metrics	Models	
	LTC Model	Incremental Learning
Mean Squared Error (MSE)	0.002488	0.337814
Root Mean Squared Error (RMSE)	0.04988	0.581218
Mean Absolute Error (MAE)	0.03179	0.561843
Mean Absolute Percentage Error (MAPE)	4.87%	94.79%
Directional Accuracy (DA)	51.25%	14.68%

TABLE III. EVALUATION METRICS FOR APPLE STOCK MARKET

Metrics	Models	
	LTC Model	Incremental Learning
Mean Squared Error (MSE)	0.00317	0.54355
Root Mean Squared Error (RMSE)	0.0178	0.73725
Mean Absolute Error (MAE)	0.014131	0.73158
Mean Absolute Percentage Error (MAPE)	1.80%	89.39
Directional Accuracy (DA)	49.36%	28.37%

Overall, the LTC Model consistently shows superior performance across all metrics, making it the most effective model for predicting Tesla and Apple stock market metrics compared to the Incremental Learning models.

## V. CONCLUSIONS AND FUTURE WORK

### A. Conclusions

This study demonstrates the application and comparative effectiveness of Liquid Neural Networks (LNN) and Incremental Learning methods in predicting the stock prices of Tesla and Apple. LNN outperformed incremental learning in almost all tested metrics, offering a more accurate and reliable forecasting tool for stock market investors. LNN's adaptability and dynamic structure make it exceptionally suited for capturing the complex, nonlinear patterns observed in financial markets like those of Tesla and Apple. The study's findings emphasize the potential of Liquid Neural Networks not only in enhancing the prediction but also in increasing the robustness of financial analytics tools. As markets continue to evolve, the adaptability and precision of models like LNNs will be crucial in developing tools that can leverage vast amounts of data for real-time decision-making, ultimately benefiting the broader field of financial technology.

### B. Future Work

Liquid Neural Networks (LNN) have shown considerable promise in predicting stock prices and open up a range of future research possibilities. One avenue involves creating hybrid models by combining LNN with other advanced machine learning methods, while another involves incorporating broader and more varied datasets—such as macroeconomic indicators or social media sentiment—to better capture the many factors driving stock prices. Researchers could also experiment with different LNN configurations, including adjustments to the liquid states and regularization approaches, to optimize performance for specific markets and prediction horizons. Additionally, real-time data streaming and incremental updates could enhance high-frequency trading strategies, and improving model interpretability through techniques like visualizing the liquid states can foster trust and encourage real-world adoption.

## REFERENCES

- [1] Polanki, K., Roa Purimetla, M., Roja D, Khader Basha Sk., Thommandru, R., & Javvadi, S. (2023, December 18). Predictions of Tesla Stock Price Based on Machine Learning Model.
- [2] Mahtab, M. T., Zaman, A. G. M., Mahin, M. R., Mia, M. N., & Islam, M. T. (2022, December 31). Stock Price Prediction: An Incremental Learning Approach Model of Multiple Regression. the AIUB Journal of Science and Engineering.
- [3] Botunac, I., Bosna, J., & Matetić, M. (2024, February 28). Optimization of Traditional Stock Market Strategies Using the LSTM Hybrid Approach.
- [4] Zhao, L., Kong, S. and Shen, Y., 2023, August. DoubleAdapt: A Meta-learning Approach to Incremental Learning for Stock Trend Forecasting. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (pp. 3492-3503).
- [5] Singh, T., Kalra, R., Mishra, S., Satakshi and Kumar, M., 2023. An efficient real-time stock prediction exploiting incremental learning and deep learning. *Evolving Systems*, 14(6), pp.919-937.
- [6] Hasani, R., Lechner, M., Amini, A., Rus, D. and Grosu, R., 2021, May. Liquid time-constant networks. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 9, pp. 7657-7666).
- [7] Thakkar, A. and Chaudhari, K., 2021. A comprehensive survey on deep neural networks for stock market: The need, challenges, and future directions. *Expert Systems with Applications*, 177, p.114800.
- [8] Zhang, L., Aggarwal, C., & Qi, G.-J. (2017). Stock Price Prediction via Discovering Multi-Frequency Trading Patterns. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- [9] Dixon, M., Klabjan, D., & Bang, J. H. (2016). Classification-based Financial Markets Prediction using Deep Neural Networks. *Algorithmic Finance*, 6(3-4), 67-77.
- [10] Sethia, A. and Raut, P., 2019. Application of LSTM, GRU and ICA for stock price prediction. In *Information and Communication Technology for Intelligent Systems: Proceedings of ICTIS 2018, Volume 2* (pp. 479-487). Springer Singapore.
- [11] Thakkar, A. and Chaudhari, K., 2020. Predicting stock trend using an integrated term frequency-inverse document frequency-based feature weight matrix with neural networks. *Applied Soft Computing*, 96, p.106684.
- [12] Zhang, H., Lin, H. and Li, Y., 2015. Impacts of feature normalization on optical and SAR data fusion for land use/land cover classification. *IEEE Geoscience and Remote Sensing Letters*, 12(5), pp.1061-1065.
- [13] Liu S., Liao G., & Ding Y., (2018). Stock transaction prediction modeling and analysis based on LSTM. In *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE; p. 2787–2790.
- [14] R. H. Memisevic, "liquid\_time\_constant\_networks," GitHub repository, 2023. [Online]. Available: [https://github.com/raminmh/liquid\\_time\\_constant\\_networks](https://github.com/raminmh/liquid_time_constant_networks)
- [15] Wikipedia contributors, Mean directional accuracy – Wikipedia, The Free Encyclopedia, 2020, [https://en.wikipedia.org/w/index.php?title=Mean\\_directional\\_accuracy&oldid=956009191](https://en.wikipedia.org/w/index.php?title=Mean_directional_accuracy&oldid=956009191); [Online; accessed 17-July-2020].
- [16] H. Jammal, "Liquid-Neural-Networks-in-Stock-Market-Prediction," GitHub repository, 2024. [Online]. Available: <https://github.com/HusseinJammal/Liquid-Neural-Networks-in-Stock-Market-Prediction>
- [17] Hasan, A., Moin, S., Karim, A., & Shamshirband, S. (2018). Machine learning-based sentiment analysis for twitter accounts. *Mathematical and computational applications*, 23(1), 11.
- [18] Chaudhary, A., Gupta, A., Pahariya, D., & Singh, S. (2023, January 1). Stock Price Prediction of Tesla & Apple using LSTM. *ITM Web of Conferences*.