M.J. Maheronnaghsh[1]
Department of Computer Engineering,
Sharif University of Technology
email: m.j.maheronnaghsh@gmail.com

M. Gheidi[1]
Department of Computer Engineering,
Sharif University of Technology
email:

A. Younesi[1]
Department of Computer Engineering,
Sharif University of Technology
email:

M.A. Fazli[1]
Department of Computer Engineering,
Sharif University of Technology
email: fazli@sharif.edu

Leveraging AI and ML Techniques for Stock and Currency Price Prediction: A Benchmark

In the dynamic world of financial markets, accurate price predictions are essential for informed decision-making. This research proposal outlines a comprehensive study aimed at forecasting stock and currency prices using state-of-the-art Artificial Intelligence (AI) and Machine Learning (ML) techniques. By delving into the intricacies of models such as Transformers, Ichimoku Cloud, LSTM, Simple RNN, NHits, and NBeats, we seek to contribute to the realm of financial forecasting, offering valuable insights for investors, financial analysts, and researchers. This article provides an in-depth overview of our methodology, data collection process, model implementations, evaluation metrics, and potential applications of our research findings.

Keywords: Machine Learning, Finance, Stock Price Prediction

1 Introduction

The financial landscape is marked by its complexity and rapid changes, making precise price predictions a formidable challenge yet a crucial endeavor. Recent advancements in AI and ML, particularly in the domains of Transformers, LSTM, Simple RNN, NHits, and NBeats, exhibit promising results in various time-series forecasting tasks. This research proposal aims to thoroughly investigate and compare the performance of these cutting-edge techniques in the prediction of stock and currency prices.

2 Related Works

3 Methodology (The method name)

3.1 Data Collection. Our study will rely on historical stock and currency price data sourced from reputable financial databases, APIs, and institutions. This rich dataset forms the foundation for training and evaluating our models.

3.2 Data Preprocessing. Before feeding the data into our models, a rigorous preprocessing phase will be conducted. This includes addressing missing values, handling outliers, and applying standardization or normalization techniques to ensure consistent scaling across diverse features.

3.3 Data Partitioning. The collected and preprocessed data will be partitioned into distinct sets: a training set for model parameter learning, a validation set for hyperparameter tuning, and a test set for unbiased evaluation.

3.4 Model Implementation. Six distinct models will be implemented:

a. Transformers: The powerful Transformer architecture will be realized using libraries such as TensorFlow or PyTorch, incorporating attention mechanisms and positional encoding for effective sequence modeling.

b. Ichimoku Cloud: The implementation involves intricate calculations of cloud components like Tenkan-sen, Kijun-sen, Senkou Span A, Senkou Span B, and Chikou Span based on historical price data.

c. LSTM: Long Short-Term Memory (LSTM) models will be established, with careful consideration given to the number of layers, hidden units, and dropout rates.

d. Simple RNN: The Simple RNN model will be set up with a specific number of recurrent units and tailored hyperparameters.

e. NHits: An ensemble forecasting model, NHits, will be implemented, amalgamating various time-series forecasting methods to generate composite predictions.

f. NBeats: The NBeats deep learning architecture will be realized, designed to effectively capture intricate time-series patterns.

3.5 Model Training, Validation, and Evaluation. a. Training: Each model will undergo training using the training dataset, employing suitable optimization algorithms like Adam or RMSprop with carefully selected learning rates.

b. Validation: The validation dataset will be employed to fine-tune hyperparameters, striking a balance between learning rates, batch sizes, and epochs to optimize model performance.

c. Evaluation Metrics: The efficacy of our models will be evaluated using widely accepted metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

3.6 Comparative Analysis. a. Performance Metrics: A rigorous quantitative comparison of models will be conducted, relying on the aforementioned evaluation metrics. This analysis will aid in identifying the most accurate and reliable model for financial prediction.

b. Statistical Analysis: Statistical tests such as t-tests or ANOVA will be performed to ascertain the statistical significance of observed differences in model performance.

4 Problem Formulation

4.1 Problem Definition. The problem at hand pertains to the accurate prediction of price movements in the context of financial markets, specifically for cryptocurrency and stock assets. This problem revolves around the inherent challenge of anticipating the future price changes of these volatile assets, which are influenced

[1]Corresponding Author.
Version 1.18, August 31, 2023

by multifaceted factors including market sentiment, economic indicators, and global events. The task of predicting these price fluctuations carries significant importance for traders, investors, and financial institutions seeking to optimize their decision-making processes.

To address this problem, this paper explores the utilization of machine learning (ML) algorithms to predict price trends in the dynamic realm of cryptocurrency and stock markets. The application of ML algorithms offers a data-driven approach that leverages historical price data and potentially relevant features to make informed predictions. The inherent ability of ML algorithms to detect patterns, learn from historical trends, and adapt to changing market dynamics presents a promising avenue for enhancing price prediction accuracy.

The primary objective of this paper is to comprehensively assess and benchmark a selection of six distinct ML algorithms in the realm of price prediction. These algorithms include Ichimoku, N-BITS, N-HEATS, RNN, LSTM, and Transformers. By analyzing these algorithms' performance, strengths, and limitations, this research aims to provide valuable insights into the effectiveness of various ML approaches for addressing the intricate challenges of cryptocurrency and stock price prediction.

Through rigorous experimentation and evaluation, this paper seeks to benchmark each algorithm against a standardized set of evaluation metrics. These metrics encompass accuracy, precision, recall, F1-score, and Mean Squared Error (MSE), among others. The evaluation process involves training each algorithm on historical price data and testing its predictive prowess on unseen data. By quantifying the algorithms' predictive capabilities and their capacity to capture intricate market dynamics, this study aims to provide a comparative analysis that assists practitioners in selecting the most suitable algorithm for their specific use cases.

In summary, this paper endeavors to define and address the pivotal problem of cryptocurrency and stock price prediction through the lens of ML algorithms. By assessing and benchmarking Ichimoku, N-BITS, N-HEATS, RNN, LSTM, and Transformers, this research strives to offer insights that aid in understanding the efficacy of different ML techniques for accurate price prediction, thus contributing to improved decision-making strategies in the financial domain.

4.2 Algorithm Discussion.

4.2.1 Ichimoku Algorithm. The Ichimoku Kinko Hyo, commonly known as Ichimoku, is a versatile trading strategy originating from Japan. It combines multiple indicators to provide a comprehensive view of price trends, support and resistance levels, and potential entry and exit points. The key components of Ichimoku are the Tenkan-sen (Conversion Line), Kijun-sen (Base Line), Senkou Span A and B (Leading Span A and B), and Chikou Span (Lagging Span). These components collectively offer insights into trend strength, momentum, and potential reversal zones.

Ichimoku Cloud Trading Explained. Investopedia.

4.2.2 N-BITS Algorithm. N-BITS (Neural Basis Expansion Analysis for Time Series) is a neural network-based model designed for time series forecasting. It utilizes a stack of fully connected neural networks to capture both local and global patterns within a time series. N-BITS architecture involves iterative forecast updates, and it's well-suited for multi-step forecasting tasks.

Borovykh, A., Bohte, S., & Oosterlee, C. (2018). Conditional Time Series Forecasting with Neural Basis Expansion Analysis. NeurIPS.

4.2.3 N-HEATS Algorithm. N-HEATS (Neural Hierarchical Time Series) is a hierarchical approach for time series forecasting. It involves encoding time series data using convolutional neural networks (CNNs) to capture local patterns and then combining them through recurrent neural networks (RNNs) to capture global dependencies. This hierarchical structure aids in improving the model's ability to capture complex temporal patterns.

Lai, G., Chang, W., Yang, Y., & Liu, H. (2018). Modeling Long-and Short-Term Temporal Patterns with Deep Neural Networks. KDD.

4.2.4 RNN (Recurrent Neural Network) Algorithm. RNNs are a class of neural networks designed for sequence modeling. They maintain an internal state (hidden state) that captures past information and utilizes it for making predictions at each time step. However, traditional RNNs suffer from vanishing gradient problems. More advanced variants like LSTM and GRU were introduced to address these issues.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation.

4.2.5 LSTM (Long Short-Term Memory) Algorithm. LSTM is an improved variant of the RNN architecture designed to mitigate the vanishing gradient problem. LSTM cells incorporate memory cells, input, output, and forget gates to control the flow of information. This enables LSTMs to capture long-range dependencies in time series data, making them effective for forecasting tasks.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. Neural Computation.

4.2.6 Transformers Algorithm. Transformers are a type of neural network architecture introduced for natural language processing tasks. They utilize self-attention mechanisms to capture contextual relationships between input elements. This architecture has also been successfully applied to time series forecasting, where the self-attention mechanism enables capturing global dependencies and patterns within sequences.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. NeurIPS.

Table 1 Caption

epoch number = 50						
Method	5	10	15	20	25	30
NBits	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NHeats	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNN	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
LSTM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Transformer	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 2 Caption

epoch number = 100						
Method	5	10	15	20	25	30
NBits	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NHeats	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNN	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
LSTM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Transformer	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

5 Results

6 Discussion of Results

The crux of our study lies in the comprehensive discussion of our findings. We will elucidate the strengths and weaknesses of each model, providing deep insights into their ability to capture intricate price patterns. The robustness of these models across different market conditions will also be evaluated, shedding light on their suitability for stock and currency price prediction.

Table 3 Caption

epoch number = 150						
Method	5	10	15	20	25	30
NBits	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NHeats	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNN	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
LSTM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Transformer	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 4 Caption

epoch number = 200						
Method	5	10	15	20	25	30
NBits	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NHeats	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNN	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
LSTM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Transformer	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 5 Caption

epoch number = 500						
Method	5	10	15	20	25	30
NBits	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NHeats	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNN	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
LSTM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Transformer	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 6 Caption

epoch number = 500						
Method	5	10	15	20	25	30
NBits	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
NHeats	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
RNN	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
LSTM	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Transformer	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

7 Potential Limitations

The journey of research is often marred by challenges. We will transparently address any encountered limitations, such as data quality issues, intricacies in model interpretability, and potential computational constraints.

8 Trading Bot Implementation

In addition to our model evaluation, we have extended our research to implement a trading bot that leverages the power of the developed models for real-world financial trading. The trading bot, referred to as the "TradingHelper bot," provides predictions based on the trained AI and ML models, aiding traders and investors in making informed decisions. Below, we present an overview of the trading bot's implementation along with the relevant code.

8.1 Trading Bot Architecture. The TradingHelper bot is designed to predict price movements of selected indices and execute trades accordingly. It is built upon the foundation of AI and ML models, utilizing their forecasting capabilities to guide trading decisions. The bot is capable of handling multiple models simultaneously, enhancing its prediction accuracy through the collective intelligence of diverse algorithms.

8.2 Implementation Details. The implementation of the TradingHelper bot involves the integration of the models trained in this research. The bot receives input in the form of desired models and indices for prediction. Based on this input, the bot queries historical price data from the market using the TvDatafeed library. It then preprocesses the data and feeds it into the selected models for prediction.

8.3 Predictive Analysis and Decision Making. The bot's predictive analysis involves generating forecasts based on historical data using the selected models. The results of these predictions provide insights into the potential future price movements. Traders and investors can then utilize these insights to make informed trading decisions, optimizing their strategies for market success.

8.4 Integration of Trading Bot with Models. The TradingHelper bot synergizes the prowess of AI and ML models with real-time trading activities. By continually updating its prediction models and adapting to changing market conditions, the bot enables dynamic and responsive trading strategies.

9 Conclusion

Our research endeavors culminate in a succinct yet impactful conclusion. We will summarize the key findings, placing a spotlight on the model that has demonstrated superior performance in the realm of stock and currency price prediction. Moreover, we will underline the implications of our research and suggest avenues for future exploration, including the potential integration of hybrid models and external market indicators to further enhance predictive capabilities.

With the implementation of the TradingHelper bot, our research not only contributes to the field of financial forecasting but also extends its impact to real-world trading scenarios. The bot's ability to harness the predictive capabilities of AI and ML models opens doors to automated, data-driven trading strategies that can potentially yield superior results in the dynamic landscape of financial markets.

10 Future Directions

Our research serves as a foundation for future endeavors in financial forecasting. Exploring hybrid models that combine the strengths of different techniques could further enhance predictive accuracy. Additionally, the integration of external market indicators might provide valuable insights for even more robust predictions.

The integration of the TradingHelper bot represents a significant step towards automating trading decisions using advanced AI and ML techniques. Moving forward, we envision the possibility of refining the bot's decision-making algorithms, exploring deeper integration with hybrid models, and incorporating external market indicators for enhanced accuracy.

11 Acknowledgment

We extend our gratitude to Mr. Abolfazl Younesi.

List of Figures

List of Tables