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

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:

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, 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 intricate and ever-shifting terrain of the financial world presents an intricate tapestry of variables that collectively influence the trajectories of stock and currency prices. Amidst this intricate web of factors, the endeavor to accurately predict price movements emerges as a formidable challenge that resonates across industries and reverberates in decision-making chambers. The quest for such predictive accuracy is not a mere academic exercise; it is a strategic necessity that underpins the ability of market participants, investors, and financial institutions to navigate the dynamic ebbs and flows of the financial market with informed foresight.

Recent strides in the realms of Artificial Intelligence (AI) and Machine Learning (ML) have cast new light on the age-old quest for accurate price predictions. The emergence of transformative technologies has opened a Pandora's box of possibilities, breathing fresh vigor into the realm of forecasting. Among the notable stars in this technological constellation are the luminous entities of Transformers, Long Short-Term Memory (LSTM), Simple Recurrent Neural Networks (Simple RNN), NHits, and NBeats. These algorithms, which reside at the crossroads of data science and financial acumen, hold the promise of deciphering patterns from the intricate dance of historical data and casting projections onto the uncertain canvas of the future.

The Transformers, originally conceived for natural language processing tasks, have stepped beyond their linguistic confines and found a new canvas upon which to paint their predictive prowess. Their unique self-attention mechanisms, capable of discerning subtle relationships across time steps, are now poised to decipher the temporal symphony woven by financial data. Alongside, LSTM, with its ability to capture long-range dependencies, and the nimble Simple RNN, which unveils patterns in compact sequences, have embarked on a mission to decode the enigma of financial markets.

In this electrifying convergence of technology and finance, NHits and NBeats stand as torchbearers of innovation. NHits, driven by its inherent ability to imbibe multiple time scales, equips itself to tame the chaos of financial data. NBeats, on the other hand, dances to a different rhythm, embracing the very uncertainty that has bedeviled predictions. It thrives on the ebb and flow, capturing the evolving complexity with an uncanny flair.

Against this vibrant backdrop, this research proposal unfurls its sails, driven by an overarching objective: to unfurl the map of these cutting-edge techniques and navigate through the seas of prediction. The compass of this proposal points toward a thorough, systematic investigation—a comparative odyssey through these algorithmic landscapes. As we set forth, our intent is not merely to unravel the tapestry of these methodologies but also to weave our own thread of understanding. With each algorithm as a guide, we shall traverse through historical data, replete with past market whispers, in the hopes of discerning patterns that may whisper the secrets of the future.

In the chapters that follow, each algorithm will receive its spotlight, meticulously scrutinized under the magnifying lens of evaluation metrics. Our voyage will be guided by accuracy, precision, recall, the ever-nuanced F1-score, and the hauntingly recurrent Mean Squared Error (MSE). Through a series of empirical experiments, we shall unleash these algorithms on historical data, watching as they grapple with the task of predicting the unpredictable.

In conclusion, as the financial realm continues to expand and transform, the alliance between AI, ML, and finance promises to yield new vistas of understanding. The algorithms that form the bedrock of this research proposal are not merely tools; they are digital oracles that extend a hand into the unknown. Through this exploration, we hope to illuminate the contours of their predictive prowess and contribute a nuanced brushstroke to the canvas of financial understanding.

2 Related Works

The domain of time-series prediction for stock and currency price forecasting has garnered significant research attention, driven by the pressing need for accurate predictions in the financial sector. This section presents an overview of relevant studies that have contributed to advancing predictive techniques and methodologies in this area.

Traditional Statistical Models: Numerous traditional statistical models have been employed for time-series prediction in financial markets. Notably, the autoregressive integrated moving average (ARIMA) model has been widely used for capturing linear dependencies in financial time series data [8]. Similarly, the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model has been effective in modeling volatility clustering [15].

[1]Corresponding Author.
Version 1.18, September 2, 2023

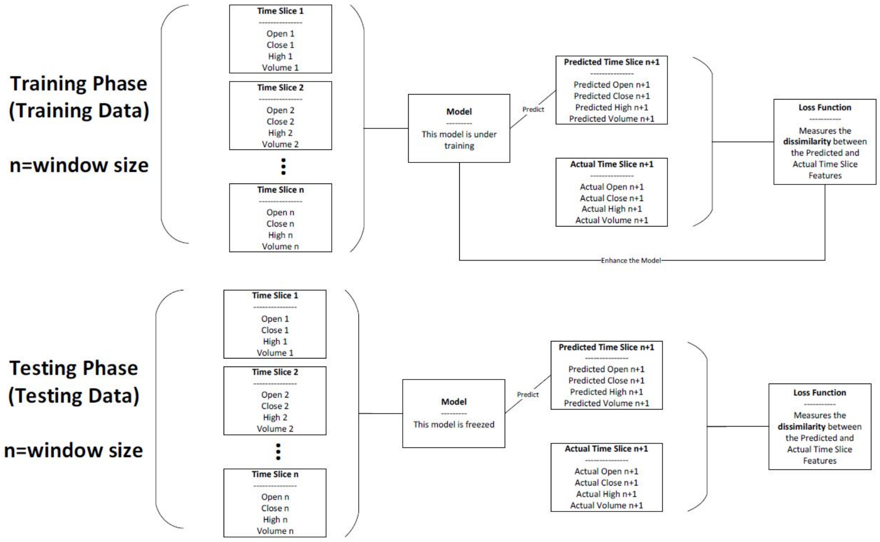


Fig. 1 This figure shows the overall process of the paper.

Machine Learning Approaches: Machine learning techniques have gained prominence due to their ability to capture complex patterns and relationships in financial time-series data. Research has explored the efficacy of Support Vector Machines (SVMs) in predicting stock prices [13]. Additionally, Random Forests have been employed to handle the non-linear dynamics of financial time series [3].

Deep Learning Techniques: With the advent of deep learning, neural networks have emerged as potent tools for time-series prediction. Long Short-Term Memory (LSTM) networks have demonstrated superior performance in capturing long-range dependencies [2]. Convolutional Neural Networks (CNNs) have been utilized for extracting spatial features from time-series data [11].

Hybrid Approaches: Hybrid models that combine multiple techniques have also been explored. The combination of ARIMA and GARCH with neural networks has demonstrated improved prediction accuracy [10]. Hierarchical hybrid models, such as the integration of LSTM and CNN, have been employed for capturing both local and global patterns [1].

Transformer-Based Approaches: The Transformer architecture, originally designed for natural language processing, has been adapted to time-series forecasting. Self-Attention mechanisms have shown efficacy in capturing temporal dependencies and handling irregularities in financial time-series data [7].

Ensemble Methods: Ensemble methods, such as the combination of multiple models for prediction, have been investigated. The fusion of multiple forecasting models, each specialized in different aspects of time series, has shown promising results in improving prediction accuracy [12].

The aforementioned studies collectively contribute to the rich landscape of time-series prediction for stock and currency markets. As the field continues to evolve, incorporating novel techniques and leveraging the advancements in machine learning and deep learning, it is essential to critically evaluate and adapt these methodologies to address the dynamic nature of financial data and enhance prediction accuracy.

3 Benchmark Methodology

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

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

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

e. 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 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 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 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 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 [5].

4.2.2 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 [6].

4.2.3 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 [9].

4.2.4 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 [4].

4.2.5 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 [14].

5 Results

Here you see the results in the table 1.
These are the plots out of the predictions 2.

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.

7 Potential Limitations

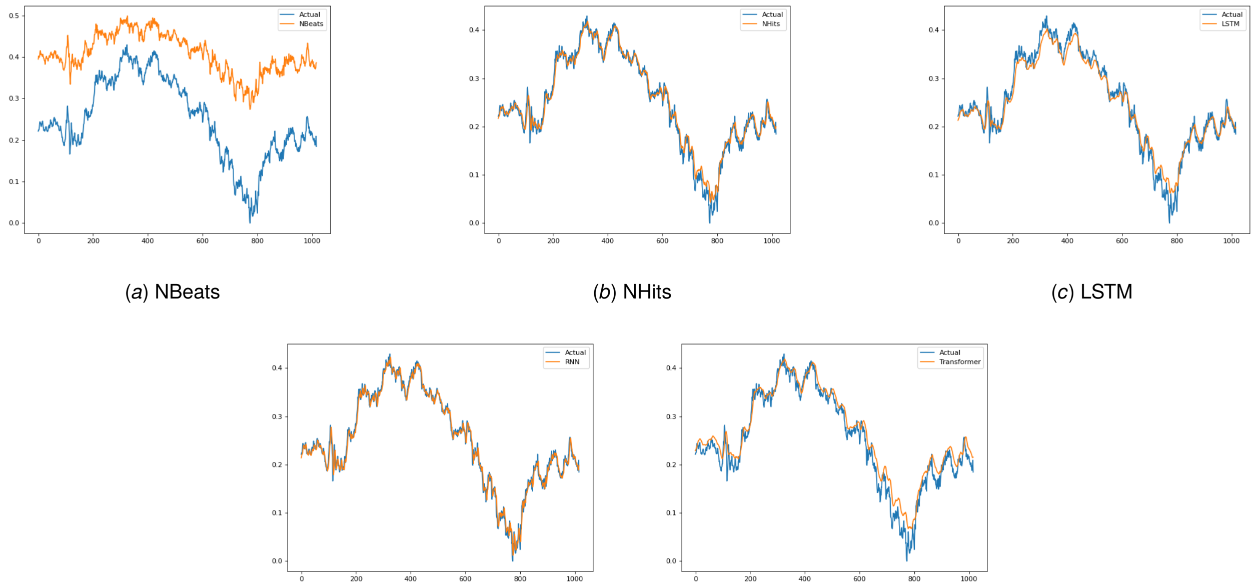
The journey of research is often marred by challenges. We will transparently address any encountered limitations, such as data quality issues, intricacies in model interpretability, and potential computational constraints.

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.



(*a*) NBeats (*b*) NHits (*c*) LSTM

(*d*) RNN (*e*) Transformer

Fig. 2 Here you can see the Close Predictions in the setting of epoch_num = 10 and sequence_length = 10.

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.

References

[1] Lai, g., chang, w., yang, y., & liu, h. (2018). modeling long-and short-term temporal patterns with deep neural networks. kdd.
[2] Hochreiter, s., & schmidhuber, j. (1997). long short-term memory. neural computation.
[3] Breiman, l. (2001). random forests. machine learning.
[4] Gers, f. a., schmidhuber, j., & cummins, f. (2000). learning to forget: Continual prediction with lstm. neural computation.
[5] Borovykh, a., bohte, s., & oosterlee, c. (2018). conditional time series forecasting with neural basis expansion analysis. neurips.
[6] Lai, g., chang, w., yang, y., & liu, h. (2018). modeling long-and short-term temporal patterns with deep neural networks. kdd.
[7] Vaswani, a., shazeer, n., parmar, n., uszkoreit, j., jones, l., gomez, a. n., ... & polosukhin, i. (2017). attention is all you need. neurips.
[8] Box, g. e. p., jenkins, g. m., reinsel, g. c., & ljung, g. m. (2015). time series analysis: Forecasting and control.
[9] Hochreiter, s., & schmidhuber, j. (1997). long short-term memory. neural computation.
[10] Zhang, x., & wu, y. (2018). time series prediction with arima-garch hybrid model. international journal of computer applications.
[11] Yao, q., cohn, t., vasilakos, a. v., & yong, j. (2019). deep learning for time-series analysis. acm computing surveys.
[12] Kuncheva, l. i., & whitaker, c. j. (2003). measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. machine learning.
[13] Zhang, g., patuwo, b. e., & hu, m. y. (1998). forecasting with artificial neural networks: The state of the art. international journal of forecasting.
[14] Vaswani, a., shazeer, n., parmar, n., uszkoreit, j., jones, l., gomez, a. n., ... & polosukhin, i. (2017). attention is all you need. neurips.
[15] Bollerslev, t. (1986). generalized autoregressive conditional heteroskedasticity. journal of econometrics.

Table 1 Results table

Model	Sequence Length	Epochs	Errors		
			MSE	MAE	RMSE
NBeats	2	10	0.0097	0.0824	0.0865
NBeats	2	50	0.000179	0.00883	0.0103
NBeats	2	100	4.17e-05	0.00476	0.00611
NBeats	2	200	5.17e-05	0.00513	0.00629
NBeats	5	10	0.00765	0.0671	0.0716
NBeats	5	50	8.66e-05	0.00708	0.00889
NBeats	5	100	7.86e-05	0.00662	0.00811
NBeats	5	200	6.49e-05	0.00592	0.00717
NBeats	10	10	0.0154	0.0901	0.0971
NBeats	10	50	0.000122	0.00870	0.0108
NBeats	10	100	9.89e-05	0.00724	0.0093
NBeats	10	200	7.10e-05	0.00616	0.00762
NHits	2	10	0.00012	0.00533	0.00833
NHits	2	50	4.19e-05	0.00420	0.00575
NHits	2	100	8.73e-05	0.00692	0.00855
NHits	2	200	9.48e-05	0.00498	0.00762
NHits	5	10	0.000202	0.00777	0.0116
NHits	5	50	0.000122	0.00698	0.00946
NHits	5	100	6.49e-05	0.00521	0.007
NHits	5	200	0.000188	0.00665	0.0102
NHits	10	10	0.000126	0.00807	0.0108
NHits	10	50	0.000134	0.00680	0.01
NHits	10	100	0.000103	0.00721	0.00954
NHits	10	200	0.000170	0.00603	0.00963
RNN	2	10	0.000187	0.00975	0.0127
RNN	2	50	0.000108	0.00787	0.00969
RNN	2	100	9.70e-05	0.00748	0.00878
RNN	2	200	0.000237	0.0117	0.0127
RNN	5	10	0.000168	0.00978	0.0123
RNN	5	50	0.000152	0.0109	0.0121
RNN	5	100	5.31e-05	0.00572	0.00693
RNN	5	200	4.78e-05	0.00542	0.00654
RNN	10	10	9.02e-05	0.00722	0.00931
RNN	10	50	0.000296	0.0143	0.0153
RNN	10	100	7.06e-05	0.00644	0.00759
RNN	10	200	5.86e-05	0.00537	0.00656
LSTM	2	10	8.83e-05	0.00603	0.00767
LSTM	2	50	7.56e-05	0.00625	0.00764
LSTM	2	100	7.12e-05	0.00634	0.00757
LSTM	2	200	6.06e-05	0.00563	0.00695
LSTM	5	10	0.000111	0.00741	0.00968
LSTM	5	50	9.81e-05	0.00820	0.00957
LSTM	5	100	4.75e-05	0.00489	0.00616
LSTM	5	200	4.70e-05	0.00497	0.0061
LSTM	10	10	0.000172	0.00928	0.0121
LSTM	10	50	5.16e-05	0.00515	0.00667
LSTM	10	100	4.61e-05	0.00478	0.00608
LSTM	10	200	4.01e-05	0.00458	0.00577
Transformer	2	10	0.000782	0.0162	0.0227
Transformer	2	50	0.000211	0.00924	0.0124
Transformer	2	100	0.000235	0.00966	0.0136
Transformer	2	200	0.000353	0.0135	0.0166
Transformer	5	10	7.64e-05	0.00609	0.00804
Transformer	5	50	0.000271	0.0118	0.0149
Transformer	5	100	0.000169	0.0078	0.0117
Transformer	5	200	8.07e-05	0.00583	0.00789
Transformer	10	10	0.000282	0.0127	0.0153
Transformer	10	50	0.000409	0.0143	0.0181
Transformer	10	100	0.000165	0.00842	0.0116
Transformer	10	200	6.07e-05	0.00503	0.00704

List of Figures

List of Tables