Vietnamese-German University

COPYRIGHT WARNING

VIETNAMESE-GERMAN UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE

Frankfurt University of Applied Sciences
Faculty 2: Computer Science and Engineering

BACHELOR THESIS

MACHINE LEARNING IN FINANCE AND ECONOMICS: COMPARATIVE ANALYSIS OF MACHINE LEARNING AND ECONOMETRIC APPROACHES IN FORECASTING VIETNAMESE STOCK MARKET

Student: Nguyễn Quốc Trung
Matriculation number: 1370166
Supervisor: Dr. Đinh Quang Vinh
Co-supervisor: Dr. Lê Văn Hà

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF BACHELOR
ENGINEERING IN STUDY PROGRAM COMPUTER SCIENCE,
VIETNAMESE-GERMAN UNIVERSITY, 2022

Binh Duong, Vietnam

Disclaimer

I hereby declare that the information reported in the paper is the result of my own, original, individual work, except where references are made. I also certify that this undergraduate dissertation has not been previously or concurrently submitted to other universities.

Nguyễn Quốc Trung

TABLE OF CONTENTS

APPENDIXES

ARIMA - Auto Regressive Integrated Moving Average

AR - Auto Regressive

I - Integrated (differencing)

MA - Moving Average

LSTM - Long Short-Term Memory

XGBoost - Extreme Gradient Boosting

RNN - Recurrent Neural Network

CLF - Classification Learner Framework

ML - Machine Learning

DL - Deep Learning

MSE - Mean Squared Error

RMSE - Root Mean Squared Error

MAPE - Mean Absolute Percentage Error

MAE - Mean Absolute Error

NSE - Nash-Sutcliffe Efficiency

FSD - Fraction of Standard Deviation

SMA - Simple Moving Average

EMA - Exponential Moving Average

ABSTRACT

The Vietnamese stock market has undergone a remarkable transformation in recent years, marked by significant expansion, notable fluctuations, and increased global recognition. This evolution has piqued the interest of a wide range of stakeholders, including local and international investors looking for opportunities, policymakers concerned with market stability and financial development, and researchers attempting to unravel the complexities of this burgeoning financial landscape. In response to the dynamic and evolving nature of the Vietnamese stock market, this thesis undertakes an extensive and multifaceted exploration. It centers on the proactive pursuit of accurate stock market performance prediction in Vietnam, a pursuit that is poised to address the growing demand for insight and foresight. The core methodology driving this research involves an in-depth comparative analysis of two distinctive, yet complementary, forecasting approaches: established econometric models and state-of-the-art deep learning techniques. This multifaceted approach recognizes the many facets of stock market dynamics in Vietnam, with the goal of not only anticipating future movements but also elucidating the underlying forces at work in this thriving economic ecosystem.

The findings of this comprehensive thesis represent a significant advance in the field of stock market prediction, significantly contributing to the growing body of knowledge in this domain. They offer a complex and nuanced understanding of the multifaceted Vietnamese stock market, including its distinct dynamics and sensitivities. This study not only identifies their individual impact but also elucidates their interdependence by dissecting and elucidating the complex web of factors that influence stock market movements, ranging from macroeconomic variables to sociopolitical influences and global interconnectedness. Furthermore, a thorough comparative analysis of the efficacy of both traditional econometric models and cutting-edge deep learning techniques in the Vietnamese context provides a detailed assessment of each approach's strengths and limitations. This, in turn, provides invaluable guidance for future prediction tool refinement, allowing for the development of more accurate and reliable forecasting models that can be tailored specifically to the Vietnamese stock market's peculiarities. Finally, these findings are a valuable resource for investors, financial analysts, and policymakers, helping them make better decisions and contributing to strategy optimization in the dynamic and ever-changing Vietnamese stock market landscape.

Keywords: Machine learning, Deep Learning, Data Science, Sequence Model, Time Series Econometrics, Financial Econometrics

ACKNOWLEDGEMENT

I would like to thank Dr. Dinh Quang Vinh for his unwavering support and invaluable guidance during the development of the Machine Learning model component of this thesis. Dr. Vinh's expertise, mentorship, and insights were critical in shaping the direction of this research and ensuring the machine learning aspects' success. His commitment to furthering knowledge in this field has been a constant source of inspiration.

I am also grateful to Dr. Le Van Ha for his excellent advice and expertise in the econometric portion of this study. Dr. Ha's deep understanding of econometric modeling and commitment to rigorous financial data analysis have been critical in bringing depth and clarity to this research. His mentorship and support were instrumental in helping me navigate the complexities of this aspect of the study.

Dr. Vinh and Dr. Ha have been more than mentors; they have been true partners in this academic journey, not only providing academic guidance but also unwavering encouragement and support. Their commitment to academic excellence and willingness to share their knowledge were critical in making this thesis a reality.

1. INTRODUCTION

1.1 Motivation and thesis scope

Vietnam's stock market, once a modest financial entity, has rapidly transformed into a dynamic and burgeoning marketplace, attracting the attention of both domestic and international investors and analysts. The expansion of the Vietnamese stock market has been marked by significant volatility, reflecting the complex interplay of economic, political, and social factors that underpin its operations. As a result, the ability to forecast stock market performance in Vietnam has become critical for investors, policymakers, and researchers.

This thesis begins a thorough investigation into the prediction of stock market performance in Vietnam, with the goal of meeting the urgent need for insights and reliable forecasting tools. It combines traditional econometric models with cutting-edge machine learning techniques to predict the stock market. This study aims to provide a comprehensive view of stock market forecasting through a comparative analysis of these methodologies, with a particular emphasis on the Vietnamese context.

1.2 Thesis Structure

This thesis includes five main parts::

- Part 1: Introduction: Gives a general overview and motivation of a thesis
- Part 2: Methodology: Introduce both the econometrics model, Machine Learning model and the hybrid model, capturing the advantages of both model
- Part 3: Data Generating Process: Display the data source and the data preprocessing procedure.
- Part 4: Results: Show all the results of one-step forecasting and multi-step forecasting in comparison
- Part 5: Conclusion: Briefly sum up everything

2. METHODOLOGY

2.1 ARIMA

The ARIMA model, first presented by Box and Jenkins in 1976, is widely used in statistical linear modeling to predict univariate time series. Time series can be broken down into three basic components: current, historical, and random mistakes. Therefore, moving average MA(q) (q random errors), auto-regression AR(p) (an additive linear function of p prior observations), and d (an integer that indicates that a series is stationary) combine to form ARIMA. The ARIMA (p, q, d) can be represented as:

$$\Delta^d y(t) = c + \sum_{j=1}^{p} \alpha_j \times y(t-j) + \epsilon(t) + \sum_{j=1}^{q} \beta_j \times \epsilon(t-j)$$

Formula 1: ARIMA (p, d, q)

ARIMA models consist of three main components:

The AR component represents the model's autoregressive part, which captures the relationship between the current observation and its past values. It denotes that the current value is linearly dependent on previous values with a time lag, which is denoted by "p" in ARIMA (p, d, q). The number of lags used in the model is represented by the parameter 'p'.

Integrated (I) Component: The differencing component, denoted by the letter "I," tells us how many differences are required to make the time series stationary. Because it makes the modeling process easier, stationarity is an essential property for time series analysis. To achieve stationarity, the number of differences needed is indicated by the differencing parameter 'd'.

Moving Average (MA) Component: The MA component symbolizes the model's moving average portion, which simulates the relationship between the current value and earlier

errors or white noise. It demonstrates that there is a linear dependence between the current value (represented by the letter "q" in ARIMA (p, d, q) and the earlier error words. The number of lags of the error terms used in the model is represented by the parameter "q."

2.2 XGBoost

Extreme Gradient Boosting, or XGBoost, is an ensemble learning algorithm that has gained a lot of recognition in the machine learning community for its remarkable predictive powers. Fundamentally, XGBoost makes use of a gradient boosting framework, which iteratively builds an ensemble model by fusing together several weak learners, most often decision trees. It performs exceptionally well in the iterative model improvement process for optimizing a user-specified loss function, successfully identifying complex patterns in the data. Technical features of XGBoost include sparsity-aware split finding, which improves efficiency when handling sparse data, and weighted quantile sketch, which optimizes histogram-based splitting. Additionally, by incorporating second-order gradient information, it employs robust and adaptive regularization techniques to enhance model precision and decrease overfitting. XGBoost's ability to handle a wide range of data types—from categorical to numerical further demonstrates its adaptability.

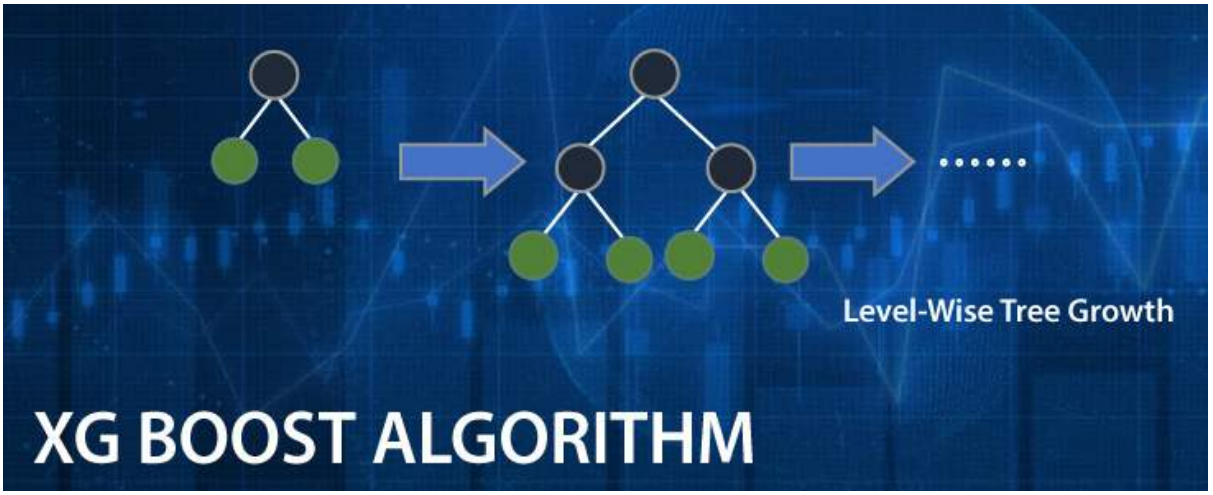

Figure 2: XGBoost theme

2.3 LSTM

Recurrent neural network (RNN) architecture in the form of Long Short-Term Memory (LSTM) has completely changed the field of sequential data processing and prediction. LSTMs are perfect for tasks like speech recognition, natural language processing, time series analysis, and more because they are excellent at recognizing and learning long-range dependencies LSTMs can store and update data over long sequences while reducing the vanishing gradient issue because, in contrast to conventional RNNs, they have memory cells and a system of gating mechanisms that regulate the flow of information. Ent problem. As a cornerstone of deep learning, the architecture's capacity to recognize and retain both short- and long-term patterns within sequential data has opened the door for major developments in a variety of applications requiring the modeling of intricate temporal relationships.

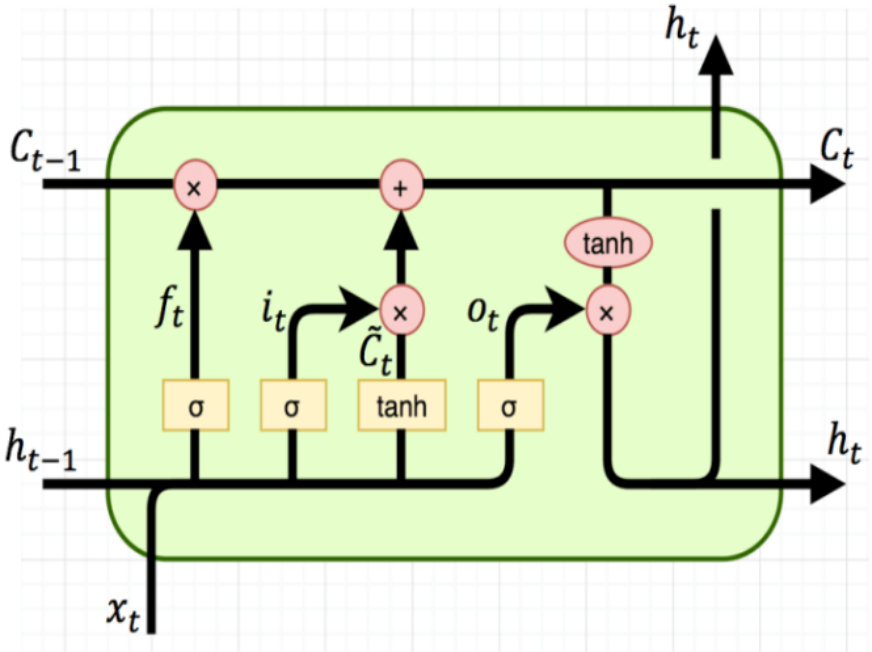

Figure 2: Structure of a LSTM cell

2.4 Attention

When processing sequential or structured data, attention mechanisms are used to concentrate on particular portions of the input data. Various elements in the input sequence are given importance weights by attention layers according to how relevant they are to the current context. This enables the model to dynamically focus on the most informative elements. Because of their capacity for selective attention, attention mechanisms have greatly enhanced the performance of many models, including time series forecasting.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

The Query (Q), Key (K), and Value (V) concepts are the foundation upon which the Attention mechanism bases its dynamic attention allocation. Which portions of the input data are given more attention during the computation depends critically on these three factors.

What the model is searching for or attempting to comprehend from the input data is represented by the Query (Q). It is a vector that is obtained from the model's present state and serves as a query for particular data. The elements of the input data that the model compares the query to are represented by the Key (K). Keys are extracted from the input data and are essential for assessing the significance of various data elements. The details pertaining to every element in the input data are contained in the Value (V). In order to provide the content that needs to be addressed based on the query and key, values are also obtained from the input data.

2.5 Hybrid model

Data from time series has both linear and nonlinear components (Zhang, 2003). There is no single model that can capture the complex mix of linear and nonlinear relationships found in time series data. Linear statistical models, such as ARIMA, are good at modeling linear aspects but fall short when it comes to nonlinear relationships (Omer Faruk, 2010; Valenzuela et al., 2008). On the other hand, nonparametric machine learning models, including XGBoost and deep learning models like LSTM and Attention-LSTM, can flexibly model nonlinear patterns as universal approximators. Therefore, to improve forecasting performance, we propose hybrid models that separately model the linear and nonlinear components of time series data. Before delving into the hybrid approach in depth, we explain why we chose the specific algorithms used to create this novel modeling technique. The hybrid approach aims to leverage the strengths of both linear and nonlinear models to better fit the full range of dynamics in time series data.

We can derive the model into two terms using the formula below:

$$y(t) = L(t) + N(t)$$

where the time series' linear and nonlinear components are denoted by L(t) and N(t), respectively. Both components (i.e. L(t), N(t)) must be estimated from time series data.

The proposed approach contains three main steps: (i) linear modeling, (ii) linear-nonlinear modeling, and (iii) forecasting future values. First, an ARIMA model is applied to extract the linear component of the time series. Next, the residuals from the ARIMA model, along with the lagged values of the original time series, are used as inputs to train machine learning models to capture nonlinear patterns. Finally, future values are forecasted from the different hybrid models. The details of these stages are as follows:

- Linear Modeling

An ARIMA model is fitted to the entire time series data to obtain predicted values L^(t) and residuals. The residuals from the ARIMA model are used as part of the input in the second hybrid modeling stage. Specifically, the ARIMA order p is used to determine the optimal lag time for the inputs to the hybrid models. This transforms the one-dimensional time series into p-dimensional data so that machine learning methods can be applied to forecast the univariate series.

Importantly, the order p is derived in a data-driven manner from the ARIMA fitting process using ACF and PACF rather than chosen heuristically. Calculating p via ARIMA is fast, eliminating the need for expensive tuning of p for each machine learning algorithm.

- Linear-Nonlinear Modeling

The ARIMA residuals are useful for selecting appropriate linear models since residuals containing nonlinear components indicate the model is not completely linear. However, statistical methods cannot detect nonlinear autoregressive patterns in residuals (Mousavi-Mirkalaei and Banihabib, 2019). Therefore, machine learning models are applied to the residuals to uncover nonlinear relationships.

The ARIMA residual at time t is:

$$\epsilon(t) = y(t) - \hat{L}(t)$$

Where ε(t) is the residual and L^hat(t) is the ARIMA forecast at time t. To discover nonlinear time series dynamics, ε(t) is modeled using machine learning algorithms like XGBoost and deep learning models like LSTM and Attention-LSTM

- Forecasting: The developed models in the previous step are used to forecast stock market

3. DATA GENERATING PROCESS

3.1 Data Collecting

3.1.1 VNQuant Package

For the Vietnam stock market, this study uses historical stock price data from reputable financial and investment firm VNDirect. One consistent source of clear, organized market data is the developer API offered by VNDirect. For the constituent stocks of the VN30 from April 2017 to November 2023, specific data such as daily opening, high, low, close, and trading volumes were obtained. The top 30 equities on the Vietnamese stock exchanges are represented by the VN30, respectively.

During the same time period, VNDirect gathered daily macroeconomic data for Vietnam, which included GDP growth, interest rates, money supply, exchange rates, inflation, and other pertinent economic factors, in addition to stock price data. Efficiently gathering stock time series data and related macroeconomic data in an analytically-ready structured format is made possible by the VNDirect API. Pandas DataFrames were used to locally store all of the data for future preprocessing, feature engineering, and modeling.

By utilizing VNDirect's vast data resources, this study seeks to create precise prediction models for the Vietnam stock market based on reliable, real-time data. The experiment design and VNDirect data preprocessing are covered in the next section.

Figure 3: VNDirect banner

3.1.2 Stock Selection

From the VN30, Asia Commercial Bank (ACB) was chosen for detailed modeling and forecasting. ACB is a top-tier retail bank in Vietnam and a component of the VN30 index. This stock was selected based on a statistical analysis of the VN30 from April 21, 2013 to November 5, 2023. ACB, in particular, had the highest trading volume, dollar value traded, and volatility as measured by standard deviation of daily returns over the last 10.5 years. Concentrating on a single major stock provides a rigorous test case for predictive modeling.

ACB represents an important banking sector that has an impact on the Vietnamese economy. By using this high volume, volatile VN30 stock, rigorous predictive modeling on a key market sector can be performed. Individual stock forecasting is also a more difficult test of model skill than aggregate index forecasting. The following section goes over the forecasting experiments for ACB that were conducted using the collected VN30 dataset from April 21, 2013 to November 5, 2023. Model performance on high-impact constituent equities will be demonstrated by isolating ACB.

Figure 4: ACB logo

3.2 Data Preprocessing

The close price of ACB was used as the primary time series for prediction in the modeling. Each day, the close price reflects the most recently traded price and is a common target for forecasting models. Weekends, holidays, and other non-trading days were excluded from the time series. This ensures that the data accurately reflects the continuous evolution of ACB's stock price over the course of active trading sessions.

All preprocessing and feature engineering were applied to the filtered close price time series from April 21, 2013 to November 5, 2023 across the entire ACB dataset. Imputing missing values, scaling the data, and generating lagged returns and technical indicators were all important steps.

This study intends to accurately forecast the progression of ACB's stock price across regular trading days by focusing the modeling on the close price series with non-trading days removed. The preprocessed close price data provides the foundation for subsequent linear and nonlinear modeling approaches.

3.2.1 Train - Test Splitting

To evaluate model performance, the ACB close price time series was divided into training and test sets. The data was split 80/20 into contiguous training and test periods for one-step forecasting. The models were fitted on the first 80% of the data and then used to forecast the close price one day ahead during the 20% test period. This metric measures basic directional accuracy over short time periods.

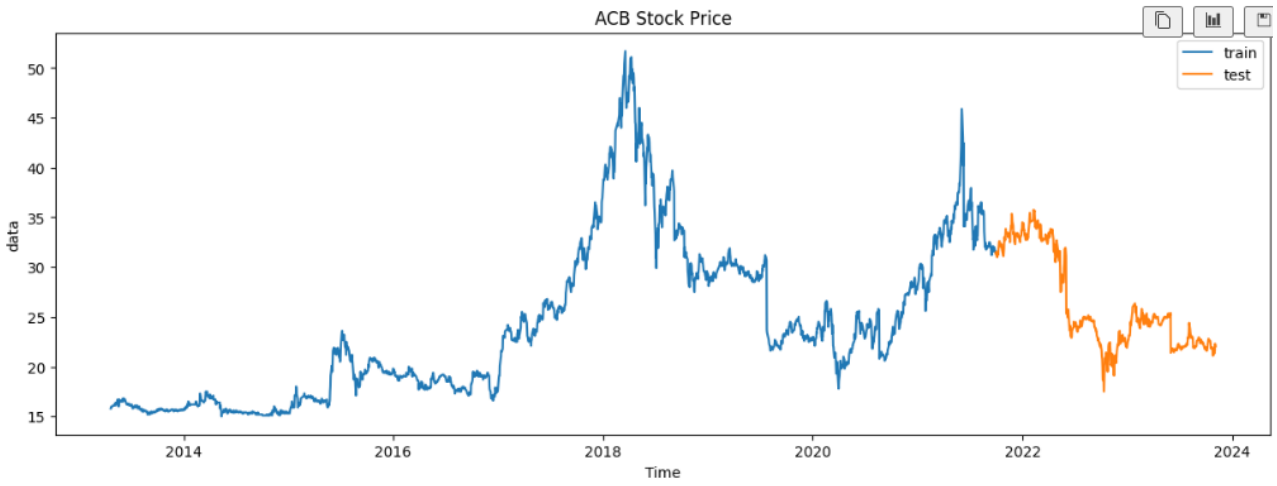

Figure 5: Train-Test Data for one-step forecasting

The last 40 observations, representing a two-month test period, were kept for multi-step forecasting. The models were trained without this test set and then used to forecast the close price over a 40-day period. A longer two-month multi-step test allows for a more rigorous evaluation of performance over longer forward-looking horizons.

Figure 6: Train-Test data for multi-step forecasting

3.2.2 Featurizing for the Machine Learning Model

In addition to the close price time series, technical indicators were developed as machine learning model features. The following moving averages were calculated: 9-day exponential moving average (EMA_9), 5-day simple moving average (SMA_5), 10-day simple moving average (SMA_10), 15-day simple moving average (SMA_15), and 30-day simple moving average (SMA_30).

These engineered features, as well as lagged close price data, were used to train statistical machine learning models such as XGBoost, LSTM, and LSTM-Attention. On top of the raw close price, the technical indicators provide additional nonlinear signals for the models to learn from. By providing interpretable price trend data, this expanded feature set aims to improve machine learning forecasting performance.

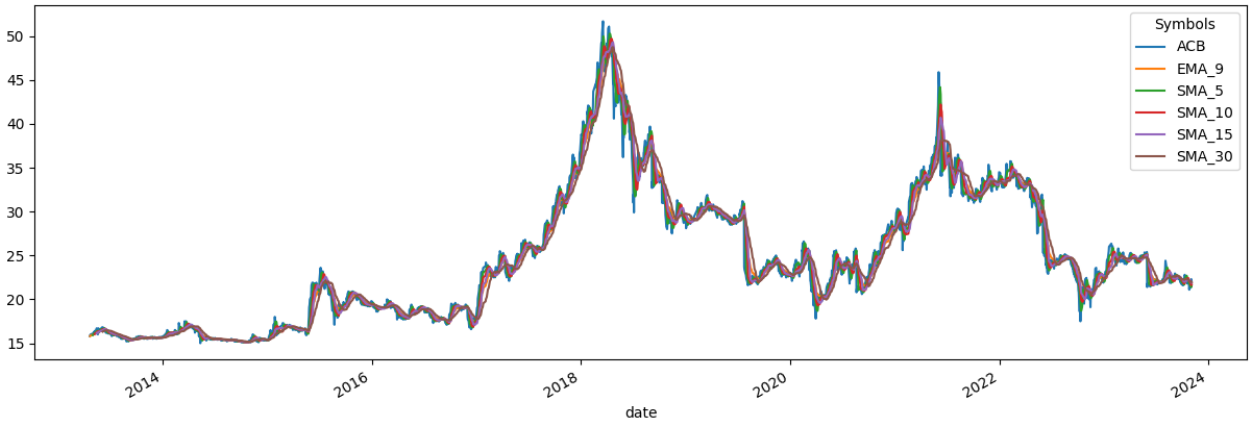

Figure 7: Actual Price and some MA line of ACB

4. Results and Discussion

4.1 Performance evaluation indicators

4.1.1 Mean Squared Error

The average squared difference between the forecasted values and the true values y is defined as the Mean Squared Error (MSE). This metric measures the precision or accuracy of predictions. In general, the most effective forecasting method will achieve the lowest MSE.

$$MSE = \sum_{i \in I_{test}} \frac{1}{N} (y_i - \widehat{y_i})^2$$

MSE effectively captures both model bias and variability. Large errors are penalized more heavily than small errors by squaring the errors, reflecting the real-world preference for consistent accuracy. Lower MSE indicates better performance, with 0 indicating perfect forecasts.

4.1.2 Mean Absolute Error

The average absolute difference between the forecasted values and the true values y is defined as the Mean Absolute Error (MAE). This metric, like MSE, measures prediction accuracy. In general, the model with the lowest MAE performs the best.

$$MAE = \sum_{i \in I_{test}} \frac{1}{N} |y_i - \widehat{y_i}|$$

MAE calculates the magnitude of errors without squaring, which means that all errors are weighted equally. In comparison to MSE, this provides a more natural and interpretable scale. MAE penalizes consistent inaccuracy, even though it is less sensitive to outliers.

4.1.3 Root Mean Squared Error

The square root of the average squared difference between predicted and actual values y is defined as the Root Mean Squared Error (RMSE). The interpretation of MSE is retained, but on the same scale as the original data.

$$RMSE = \sqrt{\sum_{i \in I_{test}} \frac{1}{N} (y_i - \widehat{y_i})^2}$$

Lower RMSE indicates better performance, with 0 indicating perfect accuracy. RMSE avoids over-weighing large errors as severely as MSE does by taking the square root. RMSE is expressed in the same units as the quantity being predicted, making interpretation easier.

4.1.4 Mean Absolute Percentage Error

The Mean Absolute Percentage Error (MAPE) measures forecast accuracy as a percentage of actual values. It is defined as the average absolute percent difference between predicted and true values y.

$$MAPE = \sum_{i \in I_{test}} 100\% \frac{1}{N} \frac{|y_i - \widehat{y_i}|}{y_i}$$

MAPE converts errors to a relative percentage scale, allowing for comparisons between datasets of varying scales or magnitudes. It is scale-independent and interpretable. Lower MAPE indicates improved model performance, with 0% indicating perfect accuracy.

4.1.5 Fraction of Standard Deviation

Another method for quantifying feature importance for machine learning models is the Fraction of Standard Deviation (FSD).

It expresses the change in model performance as a fraction of the feature's standard deviation when a feature is randomly shuffled. Larger FSD values indicate characteristics that are more important for maintaining predictive skill.

$$FSD = 2 * \frac{|SD(y) - SD(\hat{y})|}{SD(y) + SD(\hat{y})}$$

4.1.6 R Score

The R Score adds a new dimension to the significance of features in machine learning models. It calculates the strength of the relationship between a feature and the model's predictions.

Features with higher absolute R Score values correlate better with predictions and are thus deemed more important. The sign of the R Score indicates the direction of the relationship.

4.1.7 Nash Sutcliffe efficiency

The Nash-Sutcliffe Efficiency (NSE) is a predictive performance metric for hydrological models. It measures the goodness of fit between modeled and observed values.

NSE measures how well the model predicts compared to simply taking the mean of the observed data. It has a value between - and 1, with higher values indicating better model fit. An NSE of 1 indicates a perfect match between the modeled and observed values.

$$\mathrm{NSE} = 1 - \frac{\Sigma(y_i - \hat{y_i})^2}{\Sigma(y_i - \bar{y_i})^2}$$

4.2 Data representation

For one-step and multi-step forecasting experiments, the ACB close price time series data was divided into training and test sets.

The training set contains 2104 daily observations from 2013 to 2022, and the test set contains 526 observations from 2022 to 2023 for one-step forecasting.

The training set for multi-step forecasting consists of 2570 daily observations from 2013 to 2023, with a 40-day test set beginning at the end of 2023.

The data show overall rising price trends from 2013 to 2023, but no consistent seasonal patterns at the daily level.

The data is collected at a daily close price for trading days only, with non-trading days excluded.

The 10-year ACB dataset, with upward trends but no seasonality, provides a difficult benchmark for evaluating model flexibility and lag-based relationships for nonstationary financial time series.

ACB Dataset	Train observation	Test observation	Seasonal (Y/N)	Trend (Y/N)	Frequency	Period
1 Step	2104	526	N	N	1 day	2013-2023
Multi Step	2570	40	N	N	1 day	2013-2023

Table 1: Characteristics of the ACB close price time series

4.3 Model Representation

Type of Forecast			Method	
	ARIMA(p,d,q)	XGBoost	LSTM	Attention_LSTM
1 step	(2,1,0)	n_estimators = 200 gamma = 0.005 lr = 0.01 random_state = 42	total_params = 5381 batch_size = 32 optimizer = "Adam" loss = MSE	total_params = 5417 batch_size = 32 optimizer = "Adam" loss = MSE
Multistep	(4,1,3)	n_estimators = 200 gamma = 0.005 lr = 0.01 random_state = 42	total_params = 5381 batch_size = 32 optimizer = "Adam" loss = MSE	total_params = 5417 batch_size = 32 optimizer = "Adam" loss = MSE
		ARIMA_XGBoost	ARIMA_LSTM	ARIMA_Attention_LSTM
1 step	(2,1,0)	n_estimators = 200 gamma = 0.005 lr = 0.01 random_state = 42	total_params = 75 batch_size = 32 optimizer = "Adam" loss = MSE	total_params = 75 batch_size = 32 optimizer = "Adam" loss = MSE
Multistep	(4,1,3)	n_estimators = 2000 gamma = 0.01 lr = 0.01 random_state = 42	total_params = 5061 batch_size = 32 optimizer = "Adam" loss = MSE	total_params = 4123 batch_size = 32 optimizer = "Adam" loss = MSE

Table 2: Selected parameters of various forecasting method

This thesis employs auto_arima to find the best ARIMA(p,d,q) model for one-step and multi-step training splits. For each forecasting horizon, the autoregressive (p), differencing (d), and moving average (q) terms are data-adaptively tuned. The CLF method in Claraframework searches for machine learning hyperparameters such as the number of LSTM nodes, XGBoost estimators, and LSTM-Attention layers.

Nonlinear machine learning models such as XGBoost, vanilla LSTM, and attention-augmented LSTM are trained using lagged values and engineered features. The ARIMA and ML forecasts are then combined in hybrid models.

This experiment design provides a comprehensive framework for systematically benchmarking linear, non-linear, simple, and ensemble architectures by jointly tuning p,d,q orders and ML parameters while evaluating blended approaches. The following section examines the accuracy results.

4.4 Results of forecasting

4.4.1 Results of one step forecasting

In terms of MSE (0.3152), RMSE (0.5614), MAE (0.3431), MAPE (1.3379), and lowest FSD (0.0005), the ARIMA_LSTM hybrid model outperformed both standalone ARIMA and LSTM models (Table 2). ARIMA_LSTM also received the highest R Score (0.9929). The improved performance from combining ARIMA and LSTM forecasts demonstrates the utility of this ensemble approach for the ACB stock forecasting problem.

While the linear ARIMA model outperformed the nonlinear LSTM model on its own, combining them captured complementary linear and nonlinear predictive components. Though subpar on its own, the LSTM contributed nonlinear representations that helped the hybrid model outperform ARIMA alone in terms of overall skill. This demonstrates the advantages of modeling raw ARIMA and LSTM forecasts together within an ensemble framework. The combination of strengths from both methodologies improved accuracy.

Method	MSE	RMSE	MAE	MAPE	R Score	FSD	NSE
ARIMA	0,3159	0,5620	0,3447	1,3434	0,9928	0,0011	0,9857
XGBoost	2,9603	1,7205	1,5074	5,8135	0,9761	0,0242	0,8663
ARIMA_XGBoost	0,3379	0,5810	0,3672	1,4382	0,9926	0,0013	**0,9847**
LSTM	1,2928	1,137	0,7157	2,8072	0,9742	0,0189	0,9486
ARIMA_LSTM	**0,3152**	**0,5614**	**0,3431**	**1,3379**	**0,9929**	**0,0005**	0,9857
LSTM_Attention	1,6998	1,3037	0,8546	3,3696	0,9670	0,0045	0,9335
ARIMA_LSTM_Attention	0,3159	0,5620	0,3442	1,3417	0,9670	0,001	0,9857

Table 3: Results of 1 step forecasting

4.4.2 Results of multi step forecasting

Based on table 4, the linear ARIMA model outperformed all other models for multi-step predictions, with the lowest errors and highest skill scores. A high negative NSE, on the other hand, indicates that there is room for improvement. Over longer sequences, XGBoost and LSTM Attention struggled with exploding errors. Ensembling ARIMA with ML approaches no longer produces gains over standalone ARIMA.

One important factor is that the ARIMA residuals on the training set are very small, indicating that the most predictable dynamics have already been captured. As a result, there is little signal left in the residuals for the hybrid models to improve on. The LSTM and XGBoost have little additive value over long horizons because ARIMA extracts nearly all linear dependency. When a model is nearly exhaustive individually, managing multi-step uncertainty will most likely necessitate more advanced fusion or residual correction techniques.

Overall, multi-step forecasting proved more difficult, but an examination of training residuals and model errors reveals clear paths to improve sequential prediction. The residual interpretation explains why the hybrids could not outperform ARIMA in this case.

Method	MSE	RMSE	MAE	MAPE	R Score	FSD	NSE
ARIMA	**0,4817**	**0,6940**	**0,5609**	**2,5766**	0,2024	1,8065	**-1,4951**
XGBoost	29,2250	5,4060	4,8810	22,1541	0,4071	**1,3954**	-150,3714
ARIMA_XGBoost	10,4958	3,2397	2,8106	12,7316	**0,4074**	1,3973	-53,3631
LSTM	1,0569	1,0280	0,9251	4,2296	-0,2393	1,7217	-4,4744
ARIMA_LSTM	0,4837	0,6955	0,5623	2,5829	0,2125	1,8069	-1,5055
LSTM_Attention	69,4864	8,3358	8,3230	37,7300	-0,2404	1,4525	-358,9054
ARIMA_LSTM_Attention	0,4831	0,6951	0,5621	2,5823	0,2125	1,8062	-1,5026

Table 4: Results of multi step forecasting

4.4.3 Prediction plot

4.4.3.1 One step prediction

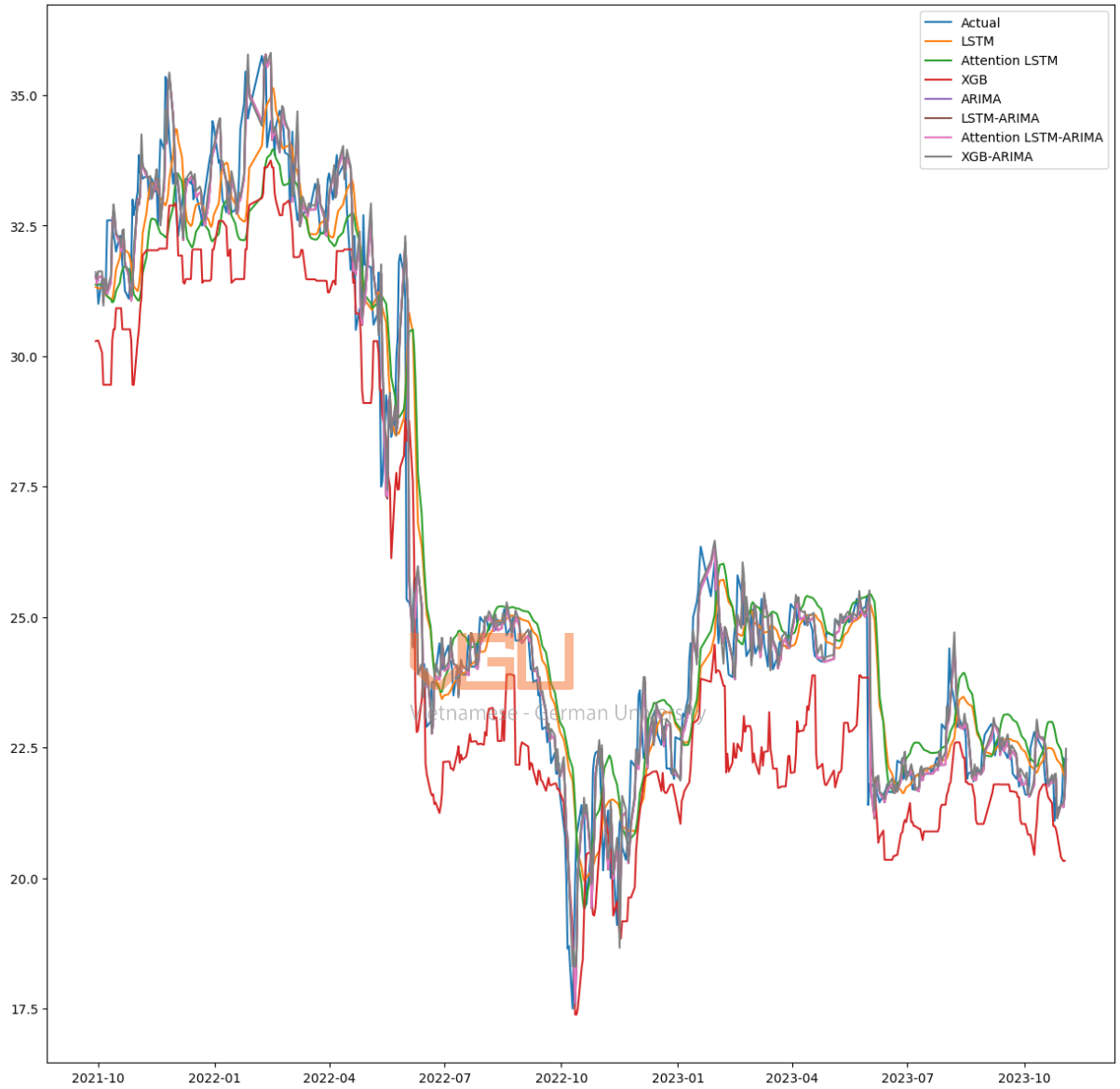

Figure 8: Visual comparison of one-step-ahead predicted values of ACB stock

Figure 9: Best Model (LSTM_ARIMA) and actual values

Figure 10: Prediction of ARIMA versus Attention_LSTM_ARIMA

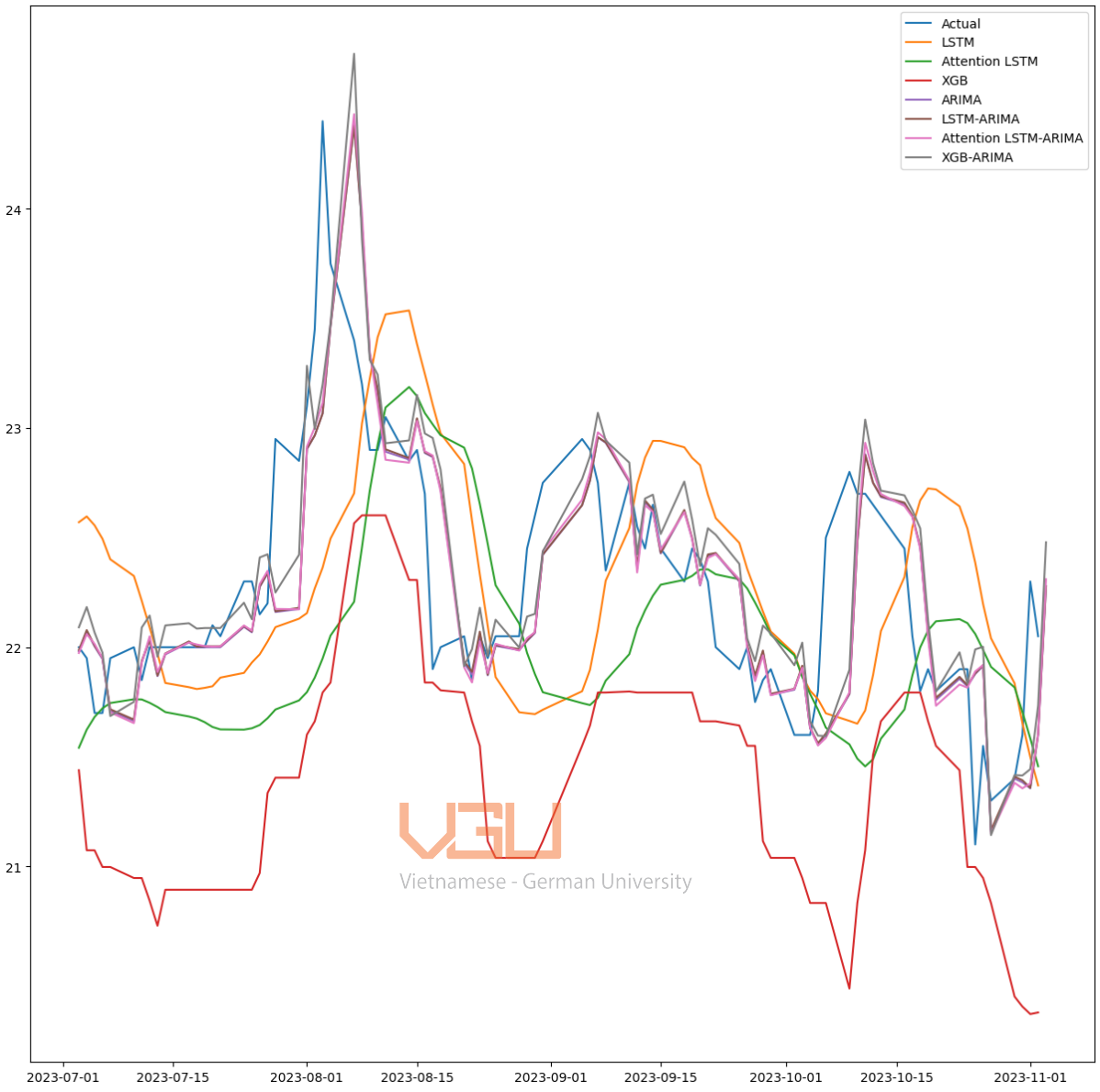

Figure 11: Visual comparison of one-step-ahead predicted values of ACB stock in recent day (from 2023-07-01 to 2023-11-05)

Figure 12: Visual comparison of one-step-ahead predicted values of ACB stock in recent day (from 2023-07-01 to 2023-11-05) between ARIMA and ARIMA_LSTM

Figure 13: Visual comparison of one-step-ahead predicted values of ACB stock in recent day (from 2023-07-01 to 2023-11-05) between ARIMA and LSTM

4.4.3.2 Multi step prediction

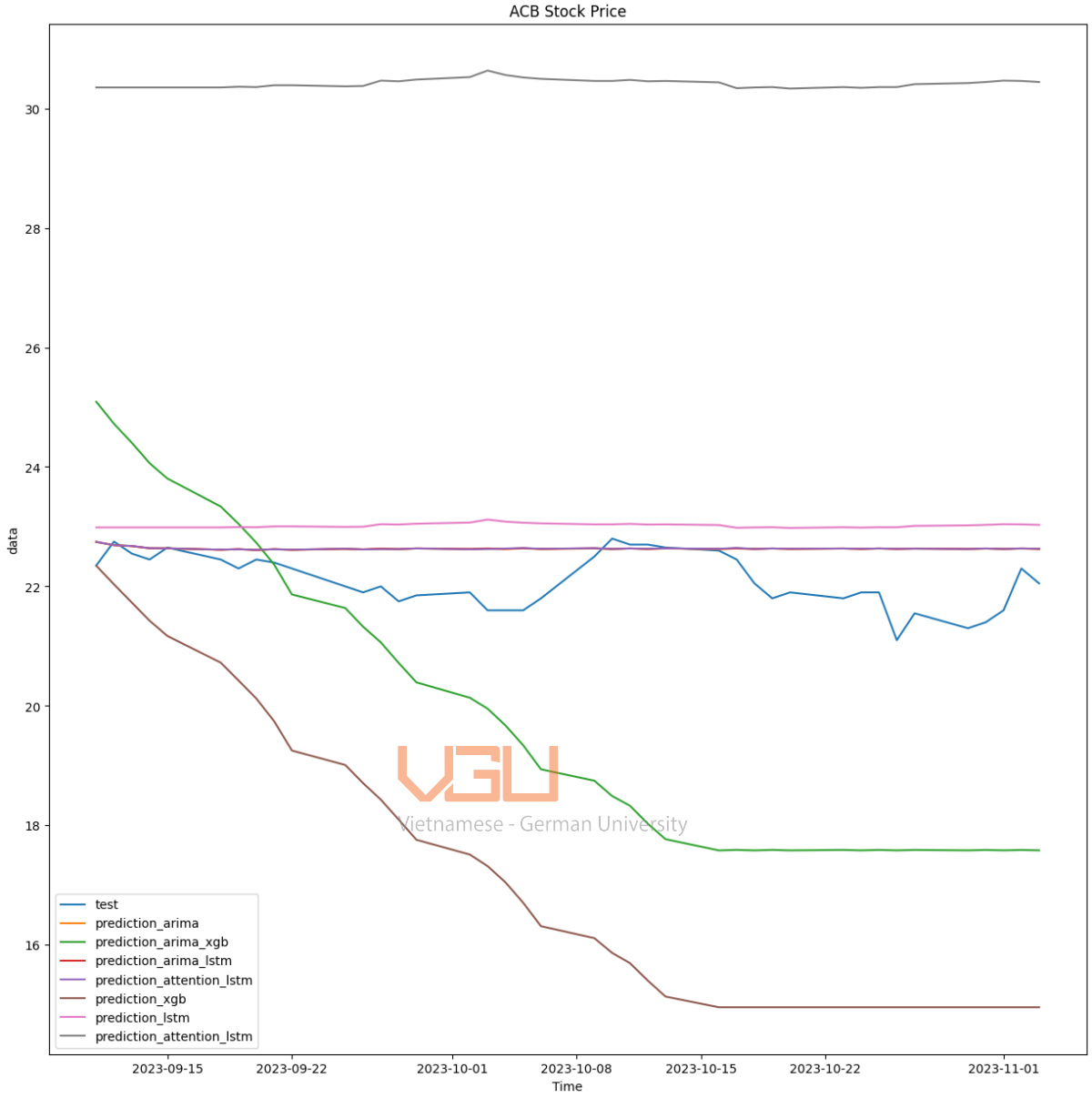

Figure 14: Visual comparison of multi-step-ahead predicted values of ACB stock in 40 days

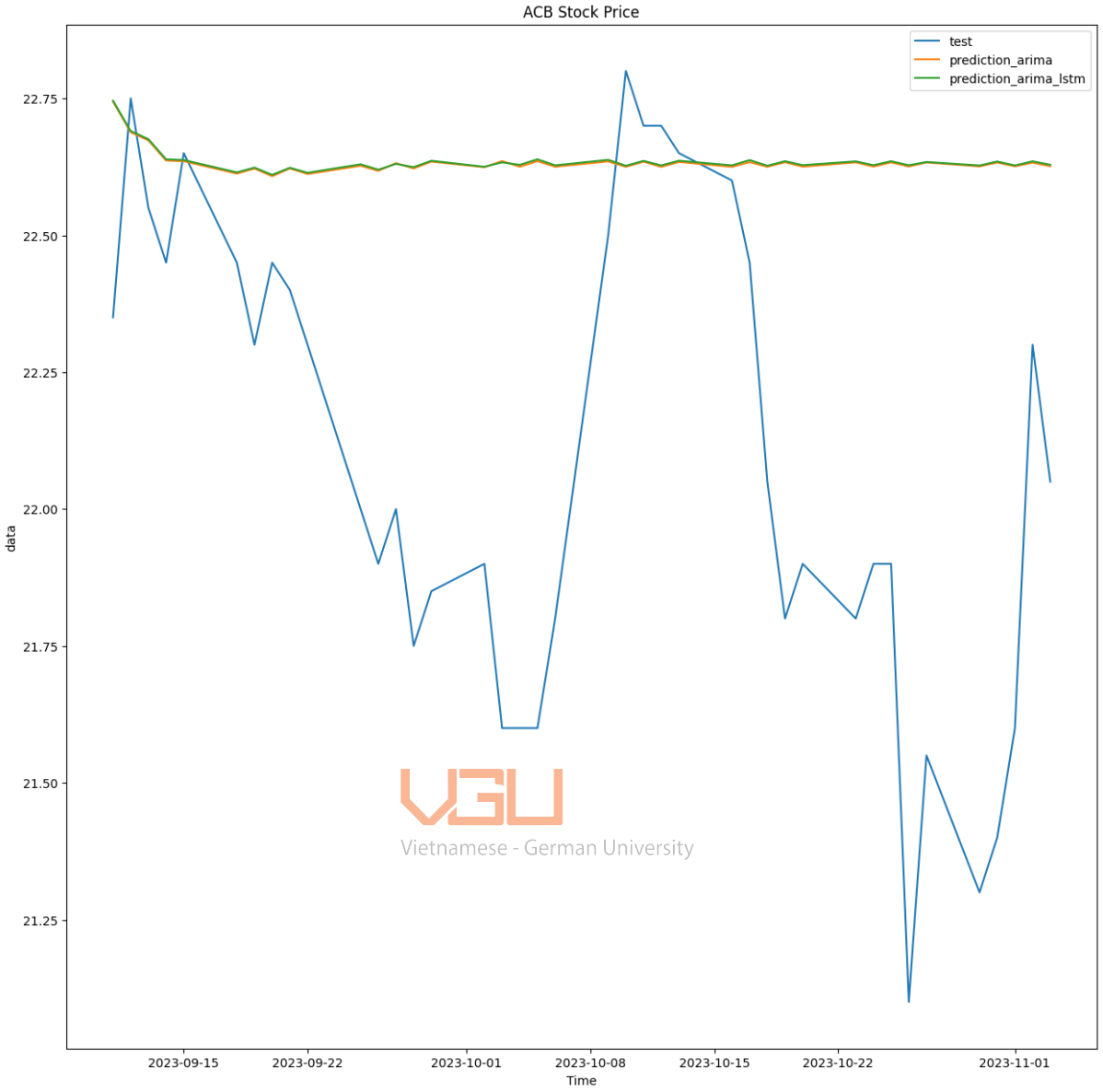

Figure 15: Visual comparison of multi-step-ahead predicted values of ACB stock in 40 days between ARIMA and ARIMA_LSTM

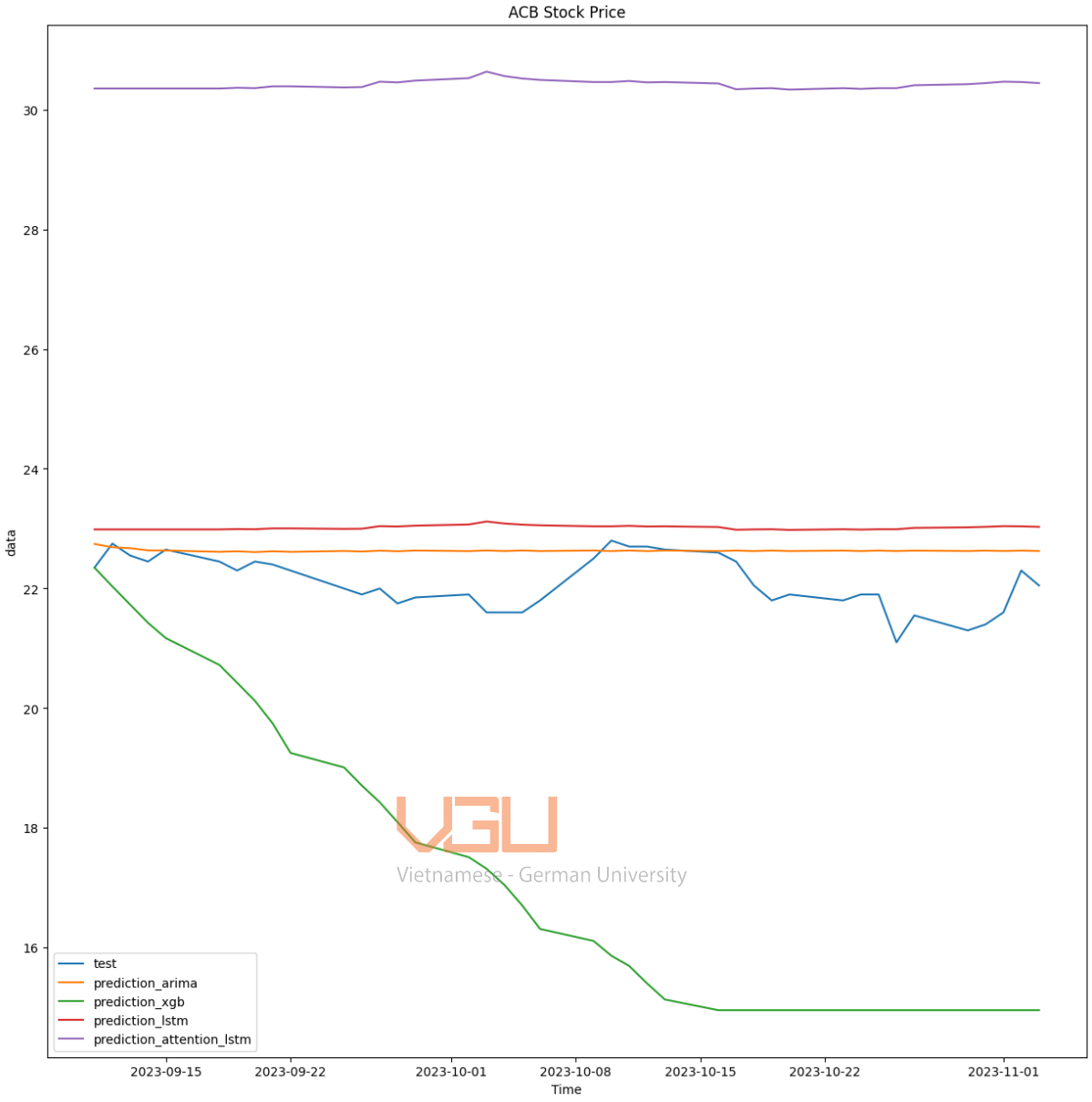

Figure 16: Visual comparison of multi-step-ahead predicted values of ACB stock in 40 days between ARIMA and Machine Learning Model

4.5 Limitations

Data Limitations

This thesis only utilized a single stock's (ACB) daily price data for analysis. Extending the modeling to include more Vietnamese stocks and economic indicators may improve the generalizability of the results. Furthermore, leveraging higher frequency intraday data may reveal additional predictive signals outside of the daily scale. In addition to technical indicators, fundamental financial ratios and macroeconomic variables can help improve explanatory power.

Methodological Limitations

For time series problems, the classical statistical and machine learning models evaluated were successful. However, modern deep learning architectures such as Transformers and Graph Neural Networks may uncover novel nonlinear relationships that traditional approaches have missed. Point estimates and one-step ahead metrics were used to evaluate performance, whereas probabilistic forecasting and density estimation can quantify uncertainty. Finally, the fundamental direct hybridization mechanisms that have been implemented could be improved to better integrate constituent model strengths.

Practical Limitations

The accuracy metrics under consideration do not take into account the real-world trading impacts of transaction fees, slippage, liquidity constraints, and other practical factors. practical viability was out of scope but is essential to translate to operational forecasting success rather than solely empirical performance.

4.6 Future Works

This thesis demonstrated that linear ARIMA models outperform machine learning techniques for forecasting Vietnamese stock prices. ARIMA-LSTM hybridization improved one-step accuracy, demonstrating the promise of blended approaches. There are several promising extensions:

- To expand market coverage and financial impact, use the hybrid frameworks to model additional individual stocks and economic indicators in addition to ACB. This allows model architectures to be tailored to different series dynamics.
- More advanced ensemble techniques, such as stacked regressors and autoregressive neural network fusion, can be used to improve predictive accuracy by better incorporating diverse method strengths.
- Create residual correction mechanisms to address error accumulation issues in long-term sequences, resulting in better real-world usability.
- Applying the Vietnam findings to stocks in developed economies will help to quantify generalization and fine-tune differences.
- Examine modern uncertainty-aware forecasting architectures such as Transformer networks and probabilistic models.

By evaluating more Vietnamese stocks, assessing transferability to other markets, and developing multi-horizon modeling, significant progress toward hybrid forecasting for stock analysis and quantitative finance can be made. The developed robust approach provides a reliable foundation for future growth.

5. Conclusion

For forecasting Vietnamese stock prices, this thesis investigated linear, nonlinear, and ensemble modeling approaches. On the ACB daily close price series, the linear ARIMA model outperformed sophisticated machine learning methods such as XGBoost, LSTM neural networks, and attentional LSTM variants across both one-step and multi-step prediction horizons.

In contrast, combining ARIMA with basic LSTM neural networks in a hybrid framework resulted in additional accuracy gains over standalone ARIMA for one-step ahead forecasts. This demonstrates how collaborative modeling can be used to leverage complementary strengths. For longer-term multi-step predictions, managing uncertainty and dependencies remains an open challenge.

The analysis not only demonstrated the linear ARIMA framework's capabilities for financial forecasting, but it also demonstrated future directions by combining multiple techniques. Adapting hybrid modeling approaches to additional stocks and indicators, as well as other sectors and economic time series, is an area of future research that will drive forecasting advances in Vietnam and emerging markets.

6. REFERENCES

O¨mer Faruk, D., 2010. A hybrid neural network and arima model for water quality time series prediction. Eng. Appl. Artif. Intell. 23, 586–594.

Box, G., Jenkins, G.M., 1976. Time Series Analysis: Forecasting and Control. Holden-Day.

Nash, J.E., Sutclie, J.V., 1970. River flow forecasting through conceptual models part I A A discussion of principles. Journal of Hydrology 10, 282–290.

Peng, T., Zhou, J., Zhang, C., Fu, W., 2017. Streamflow Forecasting Using Empirical Wavelet Transform and Artificial Neural Networks. Water 9, 40

Phan, T.T.H, Nguyen, X.H, 2020. Combining Statistical Machine Learning Models with ARIMA for Water Level Forecasting: The Case of the Red River