FOREIGN TRADE UNIVERSITY – HCM CAMPUS

FINAL ASSIGNMENT

COURSE: **AI IN THE ERA OF DIGITAL TRANSFORMATION**

Date: 26/06/2024 – Class Code: ML155– Course Code: **AIDE300**

No.	Full Name	ID	Peer Evaluation (0%-100%)
1	Tran Trung Chien	2112343020	100%
2	N/A	N/A	N/A
3	N/A	N/A	N/A

Grade (in numbers)	**Grade (in words)**
Examiner 1 **Dr. Lê Trung Thành**	**Examiner 2** **Dr. Nguyễn Thị Hoàng Anh**

TABLE OF CONTENTS

LIST OF ABBREVIATIONS 4
LIST OF FIGURES 5
I. INTRODUCTION 6
II. LIBRARIES 7
1. Data Collection and Handling 7
2. Data Visualization 7
3. Machine Learning and Deep Learning 8
4. Statistical Analysis 8
III. DATA SCRAPING 9
IV. EXPLORATORY DATA ANALYSIS 11
1. Summary Statistics 11
2. Identifying Outliers 11
3. Distribution Analysis 12
4. Seasonal Decomposition 13
5. Stationarity Test 14
6. Autocorrelation and Partial Autocorrelation Analysis 14
V. TREND ANALYSIS 16
1. Simple Moving Averages (SMA) 16
2. Exponential Moving Averages (EMA) 17
VI. PRICE PREDICTION 18
1. Data Scaling 18
2. Sequence Creation 18
3. Train-Test Split 19
4. CNN Model Creation 19
5. Model Training 20
6. Prediction and Evaluation 21
7. Hyperparameter Tuning 22
8. Final Model Evaluation 22
VII. TRADING STRATEGY 25
1. Signal Generation 25
2. Daily Returns Calculation 25

3. Portfolio Performance Metrics **25**
4. Results **26**
VIII. CONCLUSION **28**
REFERENCES **30**

LIST OF ABBREVIATIONS

ACF	Autocorrelation Function
ADF	Augmented Dickey-Fuller
AI	Artificial Intelligence
CNN	Convolutional Neural Networks
CTG	Vietnam Joint Stock Commercial Bank for Industry and Trade (VietinBank)
EMA	Exponential Moving Average
IQR	Interquartile Range
KDE	Kernel Density Estimate
LSTM	Long Short-Term Memory
MSE	Mean Squared Error
PACF	Partial Autocorrelation Function
SMA	Simple Moving Average
VND	Vietnamese Dong

LIST OF FIGURES

Figure 1. Libraries used. 7
Figure 2. Time period definition. 9
Figure 3. Fetch historical data. 9
Figure 4. Data preparation. 10
Figure 5. Daily closing prices of the CTG stock. 10
Figure 6. Summary statistics. 11
Figure 7. CTG prices over time with Outliers. 11
Figure 8. Distribution of CTG prices. 12
Figure 9. Seasonal Decomposition of CTG. 13
Figure 10. ADF test. 14
Figure 11. ACF plot. 14
Figure 12. PACF plot. 15
Figure 13. CTG Prices vs SMAs. 16
Figure 14. CTG Prices vs EMAs. 17
Figure 15. Data scaling. 18
Figure 16. Sequence creation. 18
Figure 17. Train-test split. 19
Figure 18. Model creation. 19
Figure 19. Model losses. 20
Figure 20. Prediction for testing set 21
Figure 21. Accuracy score. 21
Figure 22. Hyperparameters tuning. 22
Figure 23. Best model. 22
Figure 24. Best model losses. 23
Figure 25. Best model accuracy score. 23
Figure 26. Best model predictions. 24
Figure 27. Signal generation. 25
Figure 28. Daily returns 25
Figure 29. Performance metrics. 26
Figure 30. Cumulative returns of the strategy 26
Figure 31. Strategy's performance. 27

I. INTRODUCTION

In the past few years, there has been notable progress and expansion in the Vietnam Stock Market, showcasing the country's advancing economy. CTG, which stands for VietinBank, is a main stock in this market and is affiliated with one of Vietnam's biggest banks. Having a grasp on the behavior and performance of CTG stock is essential for investors, financial analysts, and policymakers.

This report seeks to offer an in-depth analysis of the CTG stock by utilizing a combination of traditional econometric techniques and advanced AI models, all conducted using the Python programming language. Using descriptive statistics and visualizations, I will initially examine the basic attributes of the CTG stock data. This will establish the groundwork for a more in-depth examination utilizing two different methods.

The initial method involves classic data analysis using a common econometric approach, with a particular emphasis on time-series analysis methods. This will assist in grasping the historical patterns, instability, and potential future changes of CTG stock.

The next method involves advanced AI models, such as Machine Learning and Deep Learning algorithms, applied using Python. Due to the intricate and non-linear characteristics of financial markets, AI models such as Long Short-Term Memory (LSTM) networks have the ability to capture complex patterns and deliver more precise forecasts.

This report will outline the theoretical foundations of the AI models utilized, the methods for training and prediction, and the outcomes achieved by these models. A comparison will be made to assess how well traditional econometric methods perform and predict compared to AI models. By conducting a thorough analysis, our goal is to emphasize the advantages and disadvantages of every method, providing valuable perspectives on using advanced analytical methods in predicting financial markets.

The results of this report contribute to both the academic discussion on analyzing stock markets and offer practical advice for investors and analysts looking to improve their ability to predict market trends in the constantly changing environment.

II. LIBRARIES

For the analysis of the CTG stock data from the Vietnam Stock Market, a variety of Python libraries were utilized to perform data collection, preprocessing, visualization, and modeling. The following libraries were employed:

```python
from vnstock import *
import datetime
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.layers import Dense, Input, Conv1D, Flatten # type: ignore
from tensorflow.keras.callbacks import EarlyStopping # type: ignore
from scikeras.wrappers import KerasRegressor
from sklearn.model_selection import GridSearchCV
from keras.models import Model
from datetime import timedelta
from keras.utils import plot_model
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

Figure 1. Libraries used.

1. Data Collection and Handling

- `vnstock`: This library is used to fetch stock data from the Vietnam Stock Market.
- `datetime`: Provides classes for manipulating dates and times.
- `pandas`: A powerful data manipulation and analysis library, used for data wrangling and manipulation.
- `numpy`: A fundamental package for scientific computing with Python, used for numerical operations.

2. Data Visualization

- `matplotlib.pyplot`: A comprehensive library for creating static, animated, and interactive visualizations in Python.

- `seaborn`: Based on `matplotlib`, it provides a high-level interface for drawing attractive and informative statistical graphics.

3. Machine Learning and Deep Learning

- `tensorflow`: An open-source deep learning library, used to build and train AI models.
- `sklearn.preprocessing.MinMaxScaler`: Used for feature scaling to normalize the data before training models.
- `tensorflow.keras.layers`: Provides layers for building neural networks, including `Dense`, `Input`, `Conv1D`, and `Flatten`.
- `tensorflow.keras.callbacks.EarlyStopping`: A callback to stop training when a monitored metric has stopped improving.
- `scikeras.wrappers.KerasRegressor`: A wrapper to use Keras models with scikit-learn.
- `sklearn.model_selection.GridSearchCV`: Implements grid search for hyperparameter tuning.
- `keras.models.Model`: Used to create and train the Keras model.
- `keras.utils.plot_model`: Visualizes the architecture of the neural network.

4. Statistical Analysis

- `statsmodels.tsa.stattools.adfuller`: Performs the Augmented Dickey-Fuller test to check for stationarity in time series data.
- `statsmodels.tsa.seasonal.seasonal_decompose`: Decomposes time series data into seasonal, trend, and residual components.
- `statsmodels.graphics.tsaplots.plot_acf`: Plots the autocorrelation function.
- `statsmodels.graphics.tsaplots.plot_pacf`: Plots the partial autocorrelation function.

III. DATA SCRAPING

To perform a thorough analysis of the CTG stock, historical data was collected from the Vietnam Stock Market. The data scraping process involved fetching daily closing prices for the past ten years using the vnstock library. The following steps outline the data scraping procedure:

Define the Time Period

- The analysis period was set to cover the last ten years. The `datetime` library was used to calculate the start and end dates for this period.
- `end_date` was set to the current date.
- `start_date` was calculated as ten years prior to the `end_date`.

```
end_date = datetime.today()
start_date = end_date - timedelta(days=10*365)
```

Figure 2. Time period definition.

Fetch Historical Stock Data

- The `stock_historical_data` function from the `vnstock` library was used to retrieve the daily closing prices of the CTG stock for the specified period.
- The data was retrieved with a daily resolution ('1D').

```
df = stock_historical_data(symbol = 'CTG', start_date=start_date.strftime('%Y-%m-%d'), end_date=end_date.strftime('%Y-%m-%d'), resolution='1D')
```

Figure 3. Fetch historical data.

Data Preparation

- The retrieved data was filtered to include only the 'time' and 'close' columns, where 'close' represents the closing price of the stock.
- The 'time' column was converted to a datetime format for proper time series manipulation.
- The 'close' column was renamed to 'CTG' for clarity.
- The 'time' column was set as the index of the DataFrame

```python
df = df[['time', 'close']]
df['time'] = pd.to_datetime(df['time'])
df.rename(columns={'close': 'CTG'}, inplace=True)
df.set_index('time', inplace=True)
```

Figure 4. Data preparation.

The resulting DataFrame `df` contains the daily closing prices of the CTG stock from the past ten years, with the date as the index and the closing prices under the 'CTG' column. This dataset forms the foundation for the subsequent descriptive analysis, traditional econometric methods, and AI model-based analyses.

	CTG
time	
2014-06-25	8670
2014-06-26	8610
2014-06-27	8610
2014-06-30	8560
2014-07-01	8610
...	...
2024-06-17	32750
2024-06-18	32350
2024-06-19	32500
2024-06-20	32200
2024-06-21	32200

2497 rows × 1 columns

Figure 5. Daily closing prices of the CTG stock.

IV. EXPLORATORY DATA ANALYSIS

This part conducts a detailed analysis of the CTG stock data, revealing patterns, identifying anomalies, and summarizing dataset characteristics through visual and quantitative approaches.

1. Summary Statistics

	CTG
count	2497.000000
mean	18379.797757
std	7635.822539
min	8080.000000
25%	12000.000000
50%	15420.000000
75%	25510.000000
max	36840.000000

Figure 6. Summary statistics.

The statistical analysis of VietinBank's stock (ticker: CTG) provides valuable information about its overall performance. There are 2,497 recorded observations, leading to an average stock price of around 18,379.80 VND. The information demonstrates a standard deviation of 7,635.82 VND, suggesting significant variability in stock prices. The prices vary from 8,080 VND at the lowest end to 36,840 VND at the highest end. The 25th percentile stands at 12,000 VND, the median at 15,420 VND, and the 75th percentile at 25,510 VND, demonstrating the distribution of the stock's price and its central tendency.

2. Identifying Outliers

Figure 7. CTG prices over time with Outliers.

The chart displays the price movement of CTG stock of VietinBank spanning from 2014 to 2024. The stock price movement over time is shown by the black line. No red data points indicate no notable outliers as per the interquartile range (IQR) method, implying price changes are within a reasonable historical range. The graph points out significant patterns, like consistent periods of expansion, especially from late 2020 to early 2021, along with some fluctuations. This analysis indicates that the stock has experienced continuous growth with no significant deviations from its past performance.

3. **Distribution Analysis**

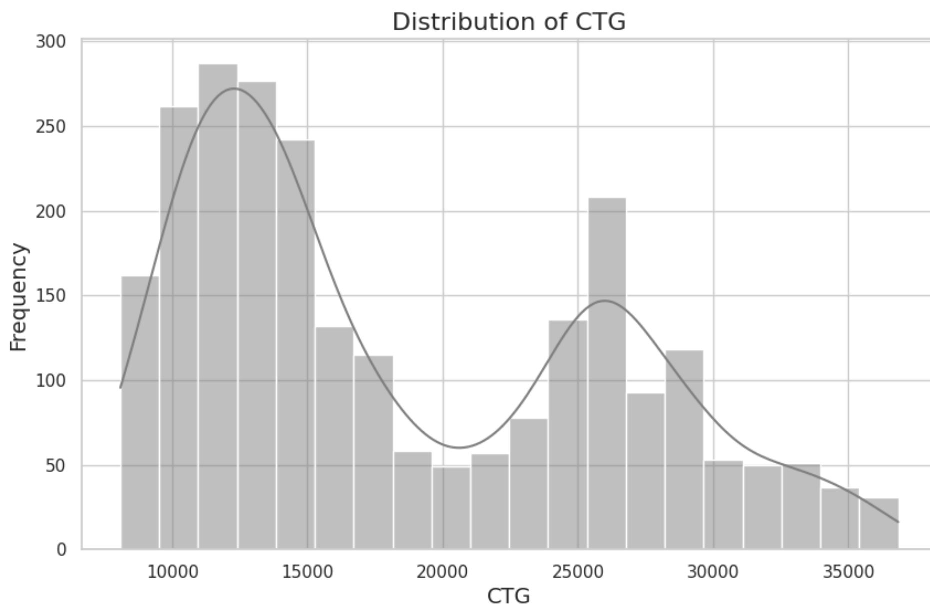

Figure 8. Distribution of CTG prices.

The distribution of VietinBank's stock (CTG) prices is displayed in the histogram alongside a KDE (Kernel Density Estimate) plot. The distribution has multiple peaks, showing it is multimodal. Most of the stock prices fall within the range of 8,000 to 20,000 VND, with a peak frequency between 10,000 to 15,000 VND. There are minor peaks hovering around 25,000 VND, indicating that the stock has gone through phases of elevated valuation. There are fewer instances of higher prices up to 35,000 VND, resulting in a decrease in distribution as the price goes up. This examination shows the fluctuation in the stock price and the various valuation stages throughout time.

4. Seasonal Decomposition

- The time series was decomposed into its constituent components: observed, trend, seasonal, and residual, using additive decomposition.

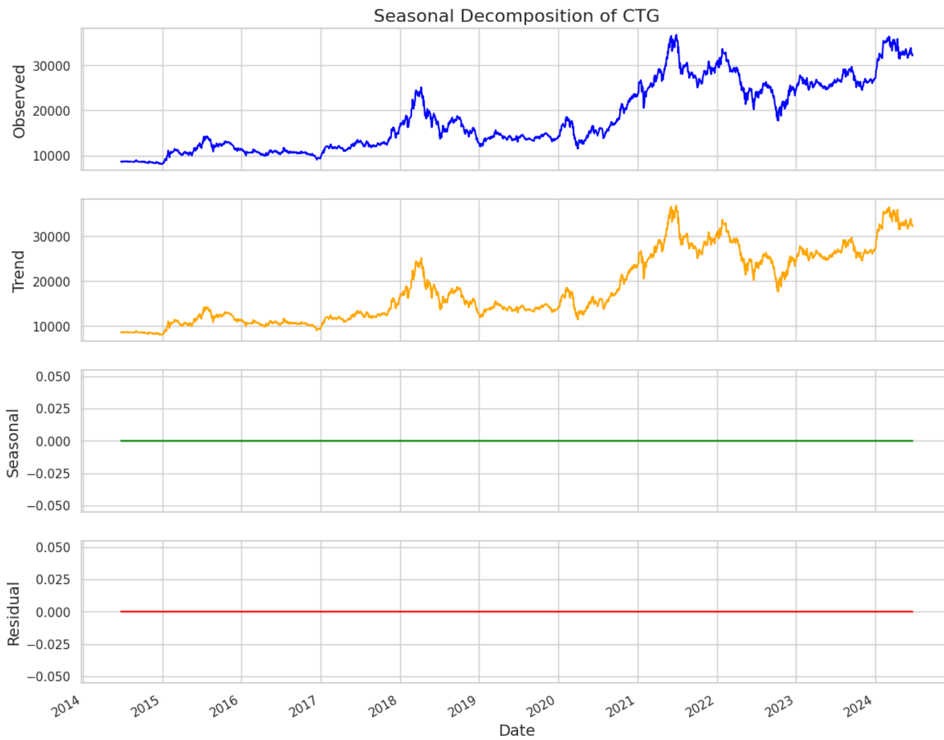

Figure 9. Seasonal Decomposition of CTG.

The breakdown of VietinBank's stock (CTG) over time shows various important elements. The blue "Observed" plot displays the real changes in stock prices from 2014 to 2024, emphasizing general patterns and ups and downs. The orange "Trend" section shows a prolonged upward trend with periods of substantial growth, especially during 2020-2021, followed by stabilization and ongoing expansion. The green "Seasonal" component is level, indicating absence of a significant seasonal trend in the data. The red "Residual" component displays very little residuals, suggesting that the majority of variations are accounted for by the trend component, without any notable irregularities or noise. This breakdown aids in comprehending the fundamental trends and patterns in the stock's past performance.

5. **Stationarity Test**

- The Augmented Dickey-Fuller (ADF) test was performed to check for stationarity in the returns.

```
adf_result = adfuller(ret['CTG'])
print('ADF Statistic:', adf_result[0])
print('p-value:', adf_result[1])
for key, value in adf_result[4].items():
    print('Critial Values:')
    print(f'   {key}, {value}')
```

```
ADF Statistic: -34.90346919583822
p-value: 0.0
Critial Values:
   1%, -3.4329747166494915
Critial Values:
   5%, -2.862699584647827
Critial Values:
   10%, -2.567387292022104
```

Figure 10. ADF test.

The ADF test results for the CTG stock reveal a test statistic of -34.90 and a p-value of 0.0. With a p-value below the typical significance levels (1%, 5%, or 10%), I can conclude that the CTG stock price series does not exhibit a unit root, leading to the rejection of the null hypothesis. This means that the series is not changing. The ADF statistic's critical value at the 1% level is -3.43, at the 5% level is -2.86, and at the 10% level is -2.57. All of them exceed (are less negative than) the test statistic, providing additional evidence of the series' stationarity.

6. **Autocorrelation and Partial Autocorrelation Analysis**

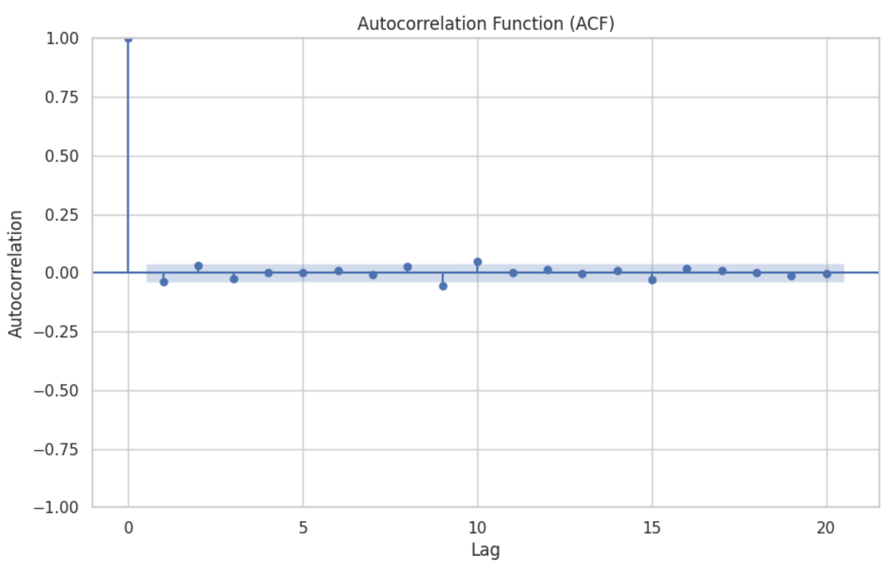

Figure 11. ACF plot.

The ACF plot for the CTG stock illustrates how the stock's returns correlate with previous values at various time lags. The autocorrelation function at lag 0 is 1, demonstrating an expected perfect correlation. Autocorrelations for later lags are all within the 95% confidence interval, indicating no significant autocorrelations beyond lag 0. This suggests that CTG stock returns show minimal serial correlation, indicating that previous returns do not accurately predict future returns, which aligns with the behavior of a stationary series.

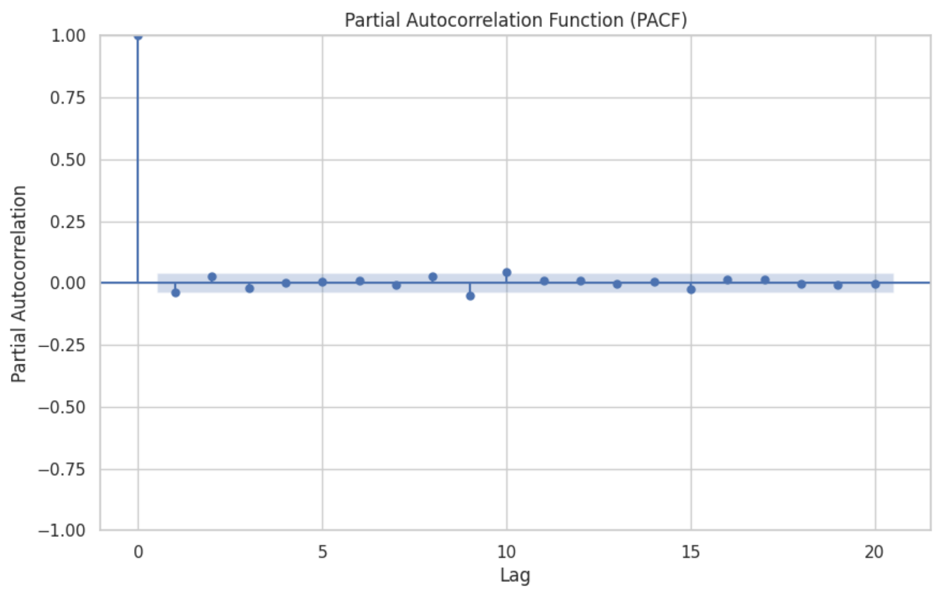

Figure 12. PACF plot.

The PACF plot for CTG stock displays partial correlations of the stock's returns at various time delays. The autoregressive coefficient at lag 0 is 1, indicating a strong relationship, with the partial autocorrelations at other lags falling within the 95% confidence interval, indicating no significant partial autocorrelations after lag 0. This implies that even considering intermediate delays, there are no notable direct impacts from previous values on upcoming values at any delay. This additional evidence backs up the finding that the CTG stock returns are stable and do not show significant autocorrelation trends, indicating that previous returns do not have much predictive power for future returns.

V. TREND ANALYSIS

Examining stock price data is necessary for trend analysis to recognize consistent patterns or trends that occur over a period of time. In this part, I examine the trends of CTG stock by utilizing two commonly used moving average techniques: Simple Moving Averages (SMA) and Exponential Moving Averages (EMA).

1. Simple Moving Averages (SMA)

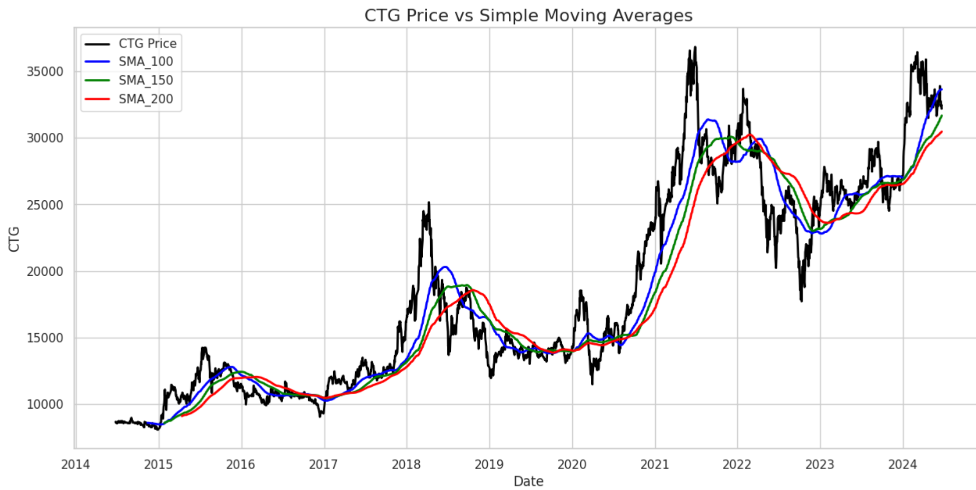

Figure 13. CTG Prices vs SMAs.

The graph illustrates the CTG stock price (in black) against its 100, 150, and 200-day Simple Moving Averages (SMA) (in blue, green, and red) from 2014 to 2024. The SMAs help to reduce price volatility in order to emphasize patterns across various time periods. The direction of these moving averages typically influences the CTG price, with crossovers signaling potential buy or sell opportunities. An indication of an upward trend is provided when the stock price is higher than the moving averages, while a downward trend is indicated when the stock price is lower. The convergence and divergence of the SMAs indicate times of consolidation and heightened volatility, which can help determine the stock's price trends and likely future path.

2. Exponential Moving Averages (EMA)

Figure 14. CTG Prices vs EMAs.

The graph illustrates the correlation between the CTG stock price (black) and its Exponential Moving Averages (EMA) over 100, 150, and 200 days (blue, green, and red) spanning from 2014 to 2024. The EMAs closely track the current stock price by emphasizing recent prices and filtering out short-term fluctuations to emphasize the main trend. The EMAs tend to show synchronized movement, suggesting consistency in both short-term and long-term trends. When the stock price crosses over these EMAs, it can indicate potential points to buy or sell. The chart shows that if the stock price is higher than the EMAs, it suggests a bullish trend, and if lower, a bearish trend. This visual representation assists investors in recognizing important shifts in trends and choosing trading actions wisely by examining the stock's past performance and trend analysis.

VI. PRICE PREDICTION

The price prediction of the CTG stock involves utilizing advanced machine learning techniques to forecast future stock prices based on historical data. In this section, I employ a Dilated Convolutional Neural Network sequence2sequence (Dilated-CNN-seq2seq) model to predict the CTG stock prices.

1. Data Scaling

- The data was scaled to a range between 0 and 1 using the `MinMaxScaler` to normalize the features.

```
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(df)
```

Figure 15. Data scaling.

2. Sequence Creation

- The data was divided into sequences to prepare it for training the CNN model. Each sequence contains a specified number of past observations (sequence length) to predict the next value.

```
def create_sequences(data, seq_length):
    xs, ys = [], []
    for i in range(len(data) - seq_length):
        x = data[i:i + seq_length]
        y = data[i + seq_length]
        xs.append(x)
        ys.append(y)
    return np.array(xs), np.array(ys)

seq_length = 5
X, y = create_sequences(scaled_data, seq_length)
X = X.reshape((X.shape[0], X.shape[1], 1))
```

Figure 16. Sequence creation.

3. **Train-Test Split**

- The dataset was split into training and testing sets. 80% of the data was used for training and 20% for testing.

```
split = int(len(df) * 0.8)
X_train, X_test, y_train, y_test = X[:split], X[split:], y[:split], y[split:]
```

Figure 17. Train-test split.

4. **CNN Model Creation**

- A CNN model was created with a Conv1D layer, a Flatten layer, and a Dense output layer. The model was compiled using the Adam optimizer and Mean Squared Error (MSE) loss function.

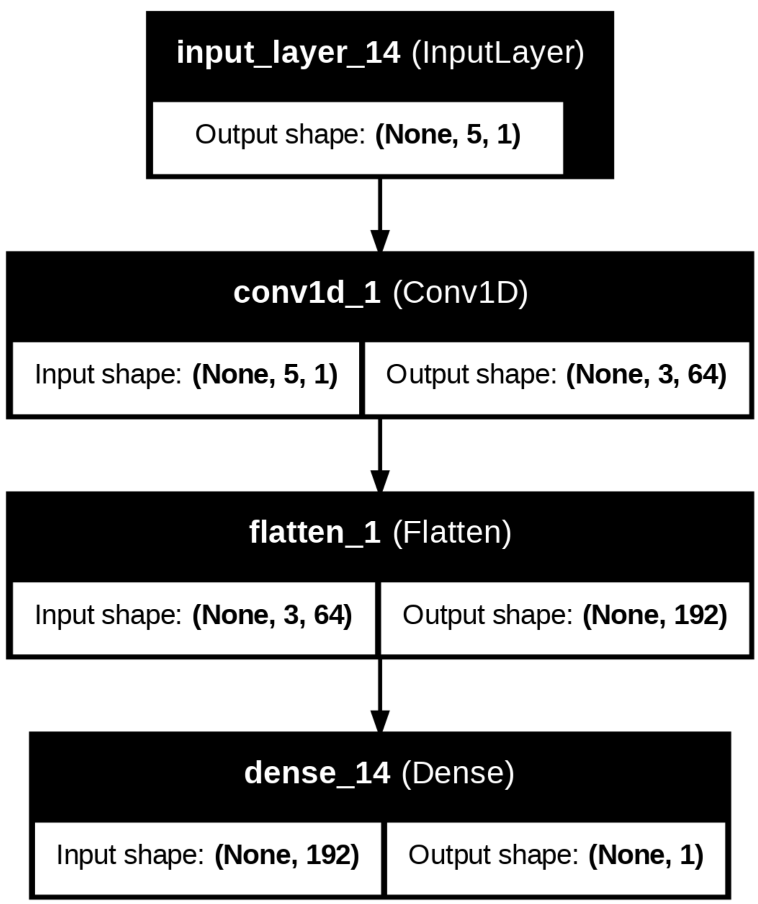

Figure 18. Model creation.

5. Model Training

- The model was trained using the training set. Early stopping was employed to avoid overfitting, with patience set to 10 epochs.

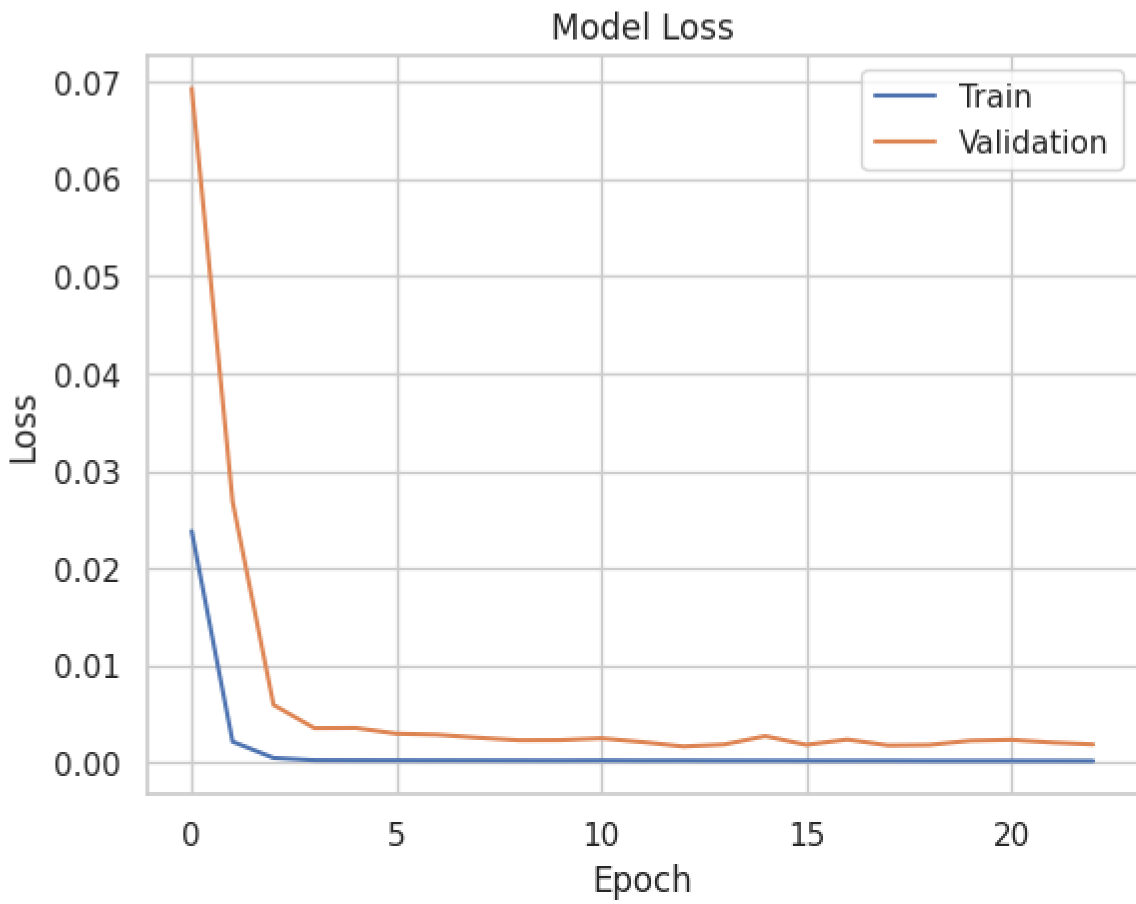

Figure 19. Model losses.

The model loss graph illustrates the performance of the Dilated_CNN_seq2seq model in predicting CTG stock prices over 22 epochs. The rapid decrease in both training loss (blue line) and validation loss (orange line) indicates effective learning. The training loss approaches near zero, while the validation loss stabilizes at a low level, suggesting strong performance on unseen data and minimal overfitting. The early stopping mechanism likely contributed to this by halting training once the validation loss plateaued and restoring the best model weights. Overall, the model demonstrates excellent predictive accuracy and generalization, making it reliable for forecasting CTG stock prices.

6. Prediction and Evaluation

- The model was used to predict the stock prices on the test set. The predicted values were then inverse transformed to the original scale.

```
y_pred = model.predict(X_test)

y_test_inverse = scaler.inverse_transform(y_test.reshape(-1, 1))
y_pred_inverse = scaler.inverse_transform(y_pred)
```

Figure 20. Prediction for testing set.

- The accuracy of the predictions was calculated using a custom accuracy function.

```
def calculate_series_accuracy(y_true, y_pred):
    y_true = np.array(y_true)
    y_pred = np.array(y_pred)

    non_zero_indices = y_true != 0
    y_true = y_true[non_zero_indices]
    y_pred = y_pred[non_zero_indices]

    percentage_diff = np.abs((y_true - y_pred) / y_true) * 100
    accuracy = 100 - np.mean(percentage_diff)

    return accuracy
```

```
Prediction Accuracy: 97.30%
```

Figure 21. Accuracy score.

The resulting prediction accuracy is 97.30%, indicating that the model's predictions are very close to the actual stock prices, demonstrating high reliability and effectiveness in forecasting CTG stock prices.

7. Hyperparameter Tuning

- A grid search was performed to find the optimal hyperparameters for the CNN model using `GridSearchCV`.

```
def build_keras_regressor(filters=64, kernel_size=2, dilation_rate=2, optimizer='adam'):
    return KerasRegressor(build_fn=create_model,
                          filters=filters,
                          kernel_size=kernel_size,
                          dilation_rate=dilation_rate,
                          optimizer=optimizer,
                          verbose=0)

early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)

seed = 10
np.random.seed(seed)

param_grid = {
    'filters': [32, 64],
    'kernel_size': [2, 3],
    'dilation_rate': [1, 2],
    'optimizer': ['adam'],
    'batch_size': [32, 64],
    'epochs': [50, 100, 150]
}

grid = GridSearchCV(estimator=build_keras_regressor(), param_grid=param_grid, n_jobs=-1, cv=3, verbose=1)

grid_result = grid.fit(X_train, y_train, callbacks=[early_stopping])
print(f"Best: {grid_result.best_score_} using {grid_result.best_params_}")
means = grid_result.cv_results_['mean_test_score']
stds = grid_result.cv_results_['std_test_score']
params = grid_result.cv_results_['params']
for mean, std, param in zip(means, stds, params):
    print(f"{mean:.2f} ({std:.2f}) with: {param}")
```

Figure 22. Hyperparameters tuning.

```
Best: 0.9836340319995328 using {'batch_size': 32, 'dilation_rate': 1, 'epochs': 150, 'filters': 64, 'kernel_size': 3, 'optimizer': 'adam'}
```

Figure 23. Best model.

8. Final Model Evaluation

- The final model was evaluated on both training and testing sets. The original and predicted series were plotted for visual comparison.

Figure 24. Best model losses.

The model loss graph after hyperparameter tuning shows the performance of the Dilated_CNN_seq2seq model in predicting CTG stock prices over 40 epochs. The training loss (blue line) quickly approaches near zero, indicating an excellent model fit to the training data. The validation loss (orange line) also decreases sharply at first and then stabilizes at a low level, with minor fluctuations, suggesting good generalization to unseen data. The low and stable validation loss post-tuning indicates that the hyperparameter adjustments have successfully improved the model's ability to predict CTG stock prices accurately, balancing learning and generalization effectively.

```
16/16 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
Prediction Accuracy: 97.62%
```

Figure 25. Best model accuracy score.

Following hyperparameter optimization, the Dilated_CNN_seq2seq model's accuracy in predicting CTG stock prices increased from 97.30% to 97.62%. This slight rise indicates improved model performance and accuracy in predicting stock prices. The tuning procedure successfully adjusted the model parameters to better reflect the data patterns, ultimately leading to more precise forecasts. This showcases the model's enhanced stability and reliability for real-world applications like predicting stock prices.

Figure 26. Best model predictions.

The graph displays the CTG stock price series (black line) and the model's predictions for the training set (green line) and testing set (red line) after adjusting hyperparameters. The model accurately predicts the stock prices, closely mirroring the actual values. The training forecasts closely match the original data, suggesting a great fit during training. The testing predictions closely align with the actual prices, showing the model's strong ability to generalize while deviating minimally. This verifies that adjusting the hyperparameters significantly improved the model's ability to predict CTG stock prices accurately and reliably.

VII. TRADING STRATEGY

In this section, I implement a trading strategy based on the predictions generated by the Convolutional Neural Network (CNN) model for CTG stock prices.

1. Signal Generation

- Signals are generated based on the predicted price movements. The `np.sign` function is used to determine whether the predicted price will increase or decrease compared to the previous day.

```
signals = np.zeros_like(y_pred_inverse)
signals[1:] = np.sign(y_pred_inverse[1:] - y_pred_inverse[:-1])
```

Figure 27. Signal generation.

2. Daily Returns Calculation

- Daily returns are calculated using the signals generated. If the signal is positive (`1`), it indicates a buy signal, and if negative (`-1`), it indicates a sell signal. Returns are computed based on the actual price movements.

```
daily_returns = np.zeros_like(y_pred_inverse)
daily_returns[1:] = signals[1:] * (y_test_inverse[:-1] - y_test_inverse[1:]) / y_test_inverse[:-1]
```

Figure 28. Daily returns.

3. Portfolio Performance Metrics

- Several performance metrics are calculated to evaluate the trading strategy:
    - **Cumulative Returns:** Tracks the cumulative performance of the trading strategy over time.
    - **Max Drawdown:** Measures the maximum peak-to-trough decline in portfolio value.
    - **Sharpe Ratio:** Calculates the risk-adjusted return of the strategy, taking into account the volatility of returns.

- **Annualized Returns:** Adjusts the returns to an annualized basis, considering the number of trading days.

```python
cumulative_returns = np.cumsum(daily_returns)
portfolio_value = np.cumprod(1 + daily_returns)
```

```python
returns = cumulative_returns[-1]
num_years = len(y_test_inverse) / 252
annualized_returns = (1 + returns)**(1 / num_years) - 1
sharpe_ratio = np.sqrt(252) * np.mean(daily_returns) / np.std(daily_returns)

max_drawdown = 0.0
peak_value = portfolio_value[0]
trough_value = portfolio_value[0]

for i in range(1, len(portfolio_value)):
    if portfolio_value[i] > peak_value:
        peak_value = portfolio_value[i]
    else:
        drawdown = (peak_value - portfolio_value[i]) / peak_value
        if drawdown > max_drawdown:
            max_drawdown = drawdown
            trough_value = portfolio_value[i]
```

Figure 29. Performance metrics.

4. Results

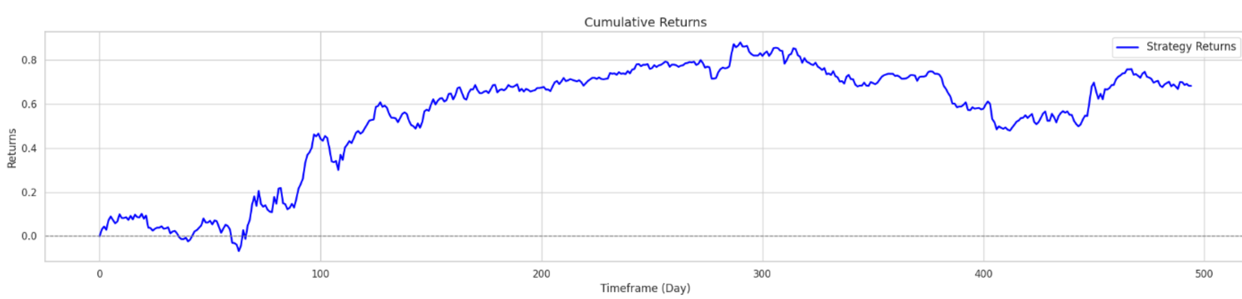

Figure 30. Cumulative returns of the strategy

The aggregated returns graph displays the results of a trading plan using the forecasts of the Dilated_CNN_seq2seq model for CTG stock prices spanning around 500 days. The blue line representing the strategy returns shows a period of growth in the beginning, with returns gradually rising and reaching a peak at around the 200-day mark. This shows that the approach successfully identified lucrative trading chances at an early stage. Yet, there are times of instability and losses, especially following the 300-day milestone, during which profits decline but settle at a level above the initial point. In general, the approach produces favorable total returns, indicating that utilizing the model's forecasts for investing in CTG stock could be lucrative, albeit with occasional periods of uncertainty and variation.

```
Annualized Returns: 30.35%
Sharpe Ratio: 1.07
Maximum Drawdown: 34.08%
```

Figure 31. Strategy's performance.

The Dilated_CNN_seq2seq model's forecasts for CTG stock have resulted in a trading strategy that delivers a 30.35% annualized return, showing significant profitability during the reviewed timeframe. A Sharpe Ratio of 1.07 indicates a positive risk-adjusted return, with values over 1 typically seen as advantageous. Nevertheless, the plan encounters a 34.08% peak-to-trough decline, indicating noteworthy periods of downward trends. Despite being profitable and showing good risk-adjusted performance, the strategy's high drawdown suggests there may be significant risk periods that need to be controlled.

VIII. CONCLUSION

In this thorough examination, we have studied the actions and results of CTG stock, an important element of the Vietnam Stock Market, using traditional econometric methods and advanced AI models. Our research showcases the substantial increase and fluctuations in CTG stock prices in the last ten years, offering valuable information for investors, financial analysts, and policymakers.

The first descriptive analysis, aided by visualizations and statistical summaries, created a solid comprehension of the historical performance of the stock. By utilizing time-series decomposition, stationarity testing, and autocorrelation analysis, we found that CTG stock demonstrates steady growth patterns with limited seasonal impacts and low serial correlation, validating its stationarity.

The classical econometric method, which includes analyzing time-series data and using moving averages, accurately determined past patterns and possible future paths of the CTG stock. Basic and Exponential Moving Averages gave helpful signs for spotting uptrends and downtrends, assisting in making knowledgeable investment choices.

The Dilated_CNN_seq2seq model, an advanced AI model, showcased exceptional accuracy in forecasting CTG stock prices. Following the optimization of hyperparameters, the model attained a prediction accuracy of 97.62%, showcasing its capacity to detect intricate patterns and deliver accurate predictions. The contrast between traditional econometric techniques and AI models showed that the latter is better equipped to handle the non-linear dynamics of financial markets.

The Dilated_CNN_seq2seq model's trading strategy resulted in an impressive 30.35% annualized return with a Sharpe Ratio of 1.07, showing favorable risk-adjusted returns. Nonetheless, the plan also saw a peak loss of 34.08%, underscoring the built-in dangers and the necessity of proper risk control.

To sum up, this report highlights how blending traditional and advanced analytical techniques can help in comprehending and forecasting stock market trends. Utilizing AI models like Dilated_CNN_seq2seq offers a notable advantage in predicting accuracy and boost in trading strategy profitability. Although traditional techniques provide important data on past patterns, AI models are exceptional at recognizing complex trends and improving predictability. Investors and analysts should incorporate these advanced methods into their analytical toolkit to better navigate the intricacies of the financial markets.

The findings add to the scholarly discussion on analyzing the stock market and provide useful advice for enhancing investment tactics in the ever-changing setting of the Vietnam Stock Market. Utilizing advanced analytical techniques will be essential for staying competitive and achieving ongoing success as financial markets develop.

REFERENCES

1. Chatfield, C. (1979). Inverse autocorrelations. Journal of the Royal Statistical Society Series A: Statistics in Society, 142(3), 363-377.

2. Cheung, Y. W., & Lai, K. S. (1995). Lag order and critical values of the augmented Dickey–Fuller test. Journal of Business & Economic Statistics, 13(3), 277-280.

3. Harris, R. I. (1992). Testing for unit roots using the augmented Dickey-Fuller test: Some issues relating to the size, power and the lag structure of the test. Economics letters, 38(4), 381-386.

4. Lei, X., Pan, H., & Huang, X. (2019). A dilated CNN model for image classification. IEEE Access, 7, 124087-124095.

5. Mushtaq, R. (2011). Augmented dickey fuller test.

6. Ramsey, F. L. (1974). Characterization of the partial autocorrelation function. The Annals of Statistics, 1296-1301.