Hanoi University of Science and Technology

**School of Information and Communication Technology**

**REPORT**

*TOPIC:*

Time-serial prediction – Stock prediction

**Class:**

139402 (

IT4110E

)

**Lecturer:**

Prof. Vũ Văn Thiệu

**Students**

Nguyễn Thế Minh Quân

20215237

Nguyễn Văn Lâm

20215219

Phạm Trung Hiếu

20215205

Nguyễn Thành Đạt

20215194

Nguyễn Tiến Thành

20215243

Hanoi, 2023



**Table of Contents**

[ABSTRACT 3](#_Toc139492402)

[CHAPTER 1: INTRODUCTION 4](#_Toc139492403)

[I. History of time-serial prediction 4](#_Toc139492404)

[II. Customer classifications 4](#_Toc139492405)

[III. Problems description 5](#_Toc139492406)

[CHAPTER 2: MODEL & ALGORITHMS 6](#_Toc139492407)

[I. Data Collection and Data preprocessing 7](#_Toc139492408)

[1. Data sources 7](#_Toc139492409)

[2. Processing steps 7](#_Toc139492410)

[II. Sequential Model 8](#_Toc139492411)

[III. LSTM 9](#_Toc139492412)

[1. Forget Gate 10](#_Toc139492413)

[*2.* Input Gate 11](#_Toc139492414)

[3. Output Date 12](#_Toc139492415)

[IV. Building and training model 12](#_Toc139492416)

[CHAPTER 3: CONCLUSIONS 14](#_Toc139492417)

[I. Evaluation 14](#_Toc139492418)

[II. Experimental results 14](#_Toc139492419)

[III. Future development 15](#_Toc139492420)

[REFERENCES 17](#_Toc139492421)

‘

# ABSTRACT

Time series prediction is the process of forecasting future values of a sequence of observations that are ordered in time. This is a common problem in various fields such as finance, economics, meteorology, and engineering.

The "Scientific Computing – IT4110E" course gave us fundamental knowledge about this technology. In our project, we aimed to predict the stock price using Sequential model and LSTM algorithms.

This report will explain the theoretical basis of the algorithms we used and how we implemented them. Then, we will analyze the effectiveness of the models at predicting stock price based on trained model using LSTM.

We experimented with Deep Learning algorithms Long short-term memory (LSTM) and supported by the Sequential model. It is a variety of recurrent neural networks (RNNs) that are capable of learning long-term dependencies, especially in sequence prediction problems. We trained and tested the models on a dataset containing stock data files of META company. We evaluated the models based on their accuracy and ability to guess the stock price correctly. Some challenges we faced included the need for a larger and more diverse dataset to improve performance.

# CHAPTER 1: INTRODUCTION

## I. History of time-serial prediction

In the 1980s and 1990s, artificial neural networks (ANNs) were developed and used for time series prediction. ANNs are a type of machine learning algorithm that can learn patterns and relationships in data. They were found to be effective in modeling nonlinear relationships in time series data.

In the early 2000s, support vector machines (SVMs) were introduced for time series prediction. SVMs are a class of powerful machine learning algorithms that can handle high-dimensional data and nonlinear relationships.

In recent years, deep learning techniques such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have been applied to time series prediction. RNNs, in particular, have been shown to be effective in modeling sequential data and have achieved state-of-the-art results in various time series prediction tasks.

One of the most significant developments in time series prediction in recent years has been the growth of big data and the availability of powerful computing resources. This has enabled the use of more complex models and the analysis of larger and more diverse dataset.

## **II**. Customer classifications

There are various ways customers can use stock prediction to inform their investment decisions. One approach is to use technical analysis, which involves analyzing past market data and patterns to predict future market trends. Technical analysis can include using chart patterns, trend lines, and technical indicators to identify buying and selling opportunities.

Another approach is to use fundamental analysis, which involves analyzing the financial health and performance of a company to determine its intrinsic value and potential for growth. Fundamental analysis can include analyzing financial statements, earnings reports, and industry trends to identify undervalued or overvalued stocks.

Customers can also use machine learning models and algorithms to predict stock prices and identify profitable investment opportunities. Machine learning models such as neural networks, random forests, and support vector machines can be trained on historical market data to predict future stock prices and identify investment opportunities.

## III. Problems description

The overarching aim of this research is to develop a model with the ability to indicate future direction of stocks price with the given information. The model will be trained on a large dataset consisting of thousands of historical stock prices and volumes for each stock. The model will perform the calculations bases on the given data. After that, it will display the line of stock price (including price prediction).

The intended functionality of the final model will be to output key price metrics for any given input data, including:

* Time
* Volume of stocks
* Price (Open, Close, High, Low)

Through extensive training and testing, the goal is to optimize the model's performance metrics such as the accuracy when any type of stock price is predicted. This designed functionality has promising applications for use cases such as investment decisions based on future market trends.

# CHAPTER 2: MODEL & ALGORITHMS

To train the model for indicate the stock price changing in the future, we need to gather and process the data, followed by model construction and training.

1. Data collection: We will collect a diverse dataset consisting of historical stocks data. This dataset should adequately represent the characteristics and variations of each stock.
2. Data preprocessing: The collected data will undergo preprocessing steps such as data calculation, and data augmentation techniques to enhance the quality and variety of the dataset. This ensures that the model can generalize well to unseen data.
3. Model construction: We will design and build a suitable model. This architecture may involve various layers, such as LSTM layers , followed by fully connected layers for classification.
4. Model training: The constructed model will be trained using the preprocessed dataset. The training process involves feeding the model with input data and adjusting its internal parameters through an optimization algorithm (e.g., stochastic gradient descent). The goal is to minimize the difference between the model's predictions and the ground truth labels in the training dataset.

During training, it's crucial to validate the model's performance on a separate validation dataset to monitor its progress and prevent overfitting. Fine-tuning the model's hyperparameters, such as learning rate and regularization techniques, may also be necessary to optimize its performance.

## Data Collection and Data preprocessing

### Data sources

We collect data of historical stocks prices from these sources:

* Investing.com
* StockInvest.us
* market.sh
* marketwatch.com, etc.

### Processing steps

After collecting data from these data sources, the raw data are transformed to the .csv data file that we can use to train our deep learning model.

The raw data has form like

A screenshot of a computer

Description automatically generated

Firstly, because price in our datasets update by days, so that we need to calculate the mean and the standard deviation of price in 1 week, 2 weeks and 3 weeks each to get the most accurate datasets. We also add [‘H-L’] and [‘O-C’] columns in data file to represent the change of price.

### Then we Calculate the simple moving averagement (SMA) with the window size of each SMA , ma\_1, ma\_2 and ma\_3 á define in the below code

A white screen with text

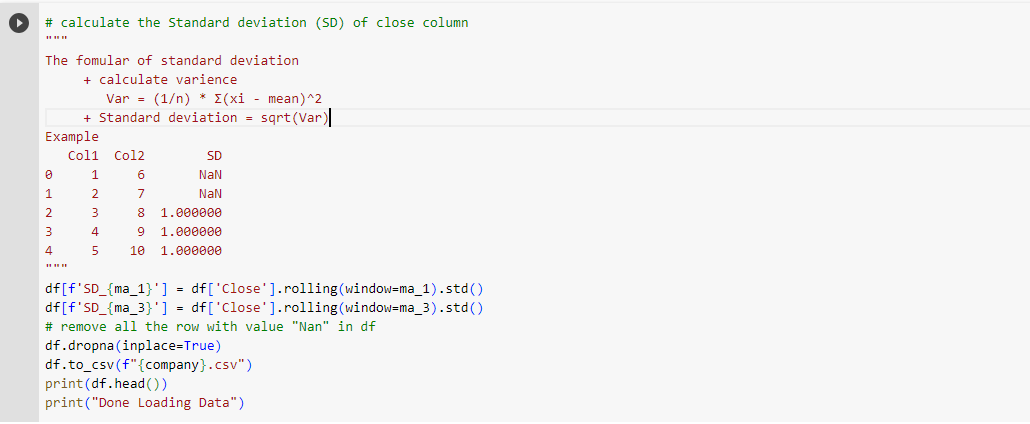
Description automatically generated A screenshot of a computer

Description automatically generated

The next part is that we calculate the standard deviation

Var = (1/n) \* Σ(xi - mean)^2

SD= sqrt(Var) where SD is the standard deviation



Then, to make the data usable for training model, we continue process the output data from **MinMaxScaler**function. From there, we fit the data to 2 arrays. We split the data set into 2 groups: traing and testing follow by 80% and 20% of total.

A white background with colorful text

Description automatically generated A close-up of a white screen

Description automatically generated

After completing all these steps, we can finally train our deep learning model with the processed data.

## Sequential Model

A sequential model is a type of neural network in deep learning that is composed of a linear stack of layers. Each layer in the stack receives input from the layer below it and passes its output to the layer above it. In other words, the data flows sequentially through the layers of the network.

Sequential models are very popular in deep learning because they are easy to build and understand, and they can be used for a wide range of tasks such as classification, regression, and sequence prediction.

In a sequential model, each layer can have a different type of activation function, number of neurons, and type of regularization, among other properties. The most common types of layers used in sequential models include:

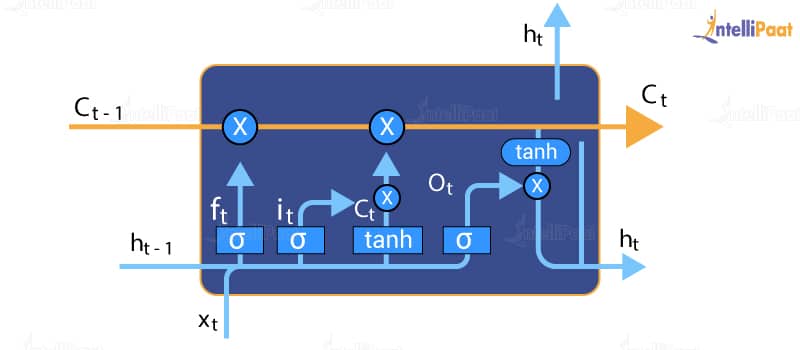
* Dense layers: These are fully connected layers where each neuron in one layer is connected to every neuron in the previous layer.
* Convolutional layers: These are used for processing images and other types of spatial data. They apply a set of filters to the input data and extract features from it.
* Recurrent layers: These are used for processing sequential data such as time series data and natural language processing. They can maintain a state that carries information from one time step to the next.
* Pooling layers: These are used to reduce the size of the input data and extract the most important features.

Sequential models are typically trained using gradient descent algorithms such as stochastic gradient descent (SGD) or Adam. The goal of training is to optimize the weights and biases of the neural network so that it can accurately predict the output for a given input.

## LSTM

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNNs) that is specifically designed to handle sequential data, such as time series, speech, and text. LSTM networks are capable of learning long-term dependencies in sequential data, which makes them well suited for tasks such as language translation, speech recognition, and time series forecasting.

LSTM has a chain structure that contains four neural networks and different memory blocks called cells.



### Forget Gate

The information that is no longer useful in the cell state is removed with the forget gate. Two inputs (input at the particular time) and (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for output 1, the information is retained for future use. The equation for the forget gate is:

where:

is the forget gate value at time step t

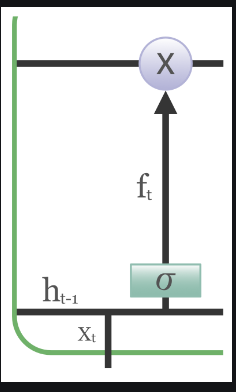
σ is the sigmoid activation function

is the weight matrix for the forget gates

is the previous hidden state of the LSTM

is the input at time step t.

is the bias vector for the forget gate.

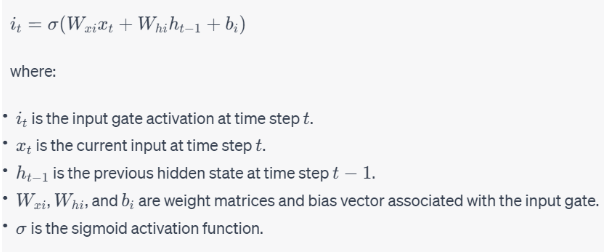


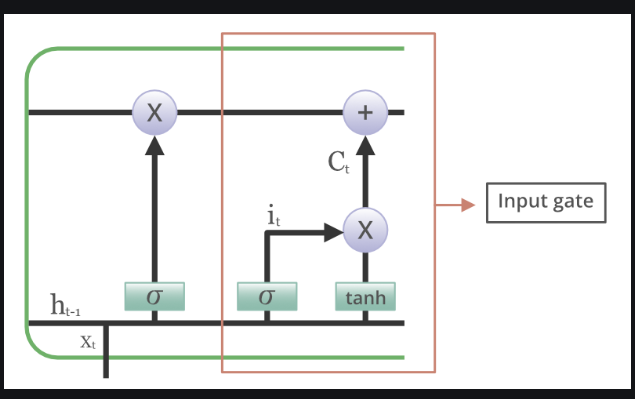
### Input Gate

The purpose of input gate

* + - The input gate determines how much new information should be stored in the cell state.
    - It is computed based on the current input and the previous hidden state.
    - It decides which parts of the input should be updated and added to the cell state.

The addition of useful information to the cell state is done by the input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs and . Then, a vector is created using tanh function that gives an output from -1 to +1,which contains all the possible values from and . At last, the values of the vector and the regulated values are multiplied to obtain the useful information. The equation for the input gate is:





* + - The formula for the sigmoid function is:

σ(x) = 1 / (1 + e^(-x))

1. The way we calculate the factor in input gate formula

Z\_i = W\_i \* [h\_(t-1), x\_t]

* Z\_i is the weighted sum of the inputs.
* W\_i is the weight matrix for the input gate.
* h\_(t-1) is the previous hidden state.
* x\_t is the current input.
* [\*] denotes concatenation of vectors.

.

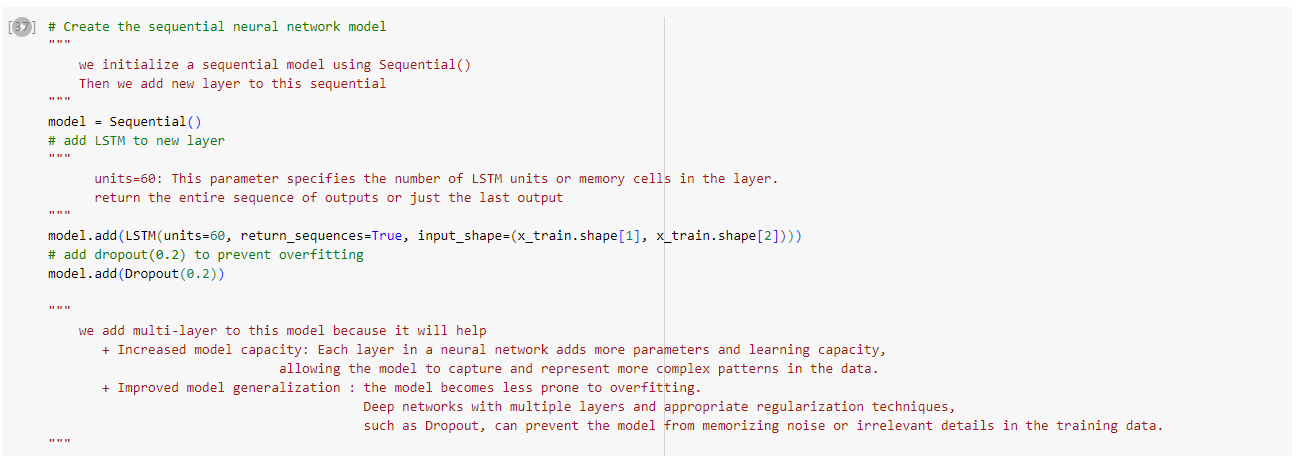
### Output Date

The task of extracting useful information from the current cell state to be presented as output is done by the output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter by the values to be remembered using inputs and At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell. The equation for the output gate is:

Where F(t) is the result of forget gate, it may be 1 or 0, respectively accept, save or ignore data from . I(t) is the result of input gate. And we have, . Is defined as the potential gate.

## Building and training model

We create an instance of the Sequential class, which is used to build a linear stack of layers for the neural network model. Then we add 5 layers of LSTM to the model with randomly sets 20% of the input units to 0 during training to prevent overfitting of each layer. After that, we add a dense layer with the number of units equal to the length of the output feature ( to connect 5 layers together.

 A screen shot of a computer

Description automatically generated

We compile the model and sets the optimizer and loss function. The optimizer used here is Adam, which is a widely used optimization algorithm in deep learning, and the loss function is mean squared error, which is commonly used for regression problems. After that, we train the model on the training data. The is the input data, is the target data, epochs is the number of times the model will be trained on the entire training dataset , is the number of times the model will be trained on each epoch (we set epochs = 120 and = 40), and is a flag that enables the use of multiple CPUs for parallel processing during training.

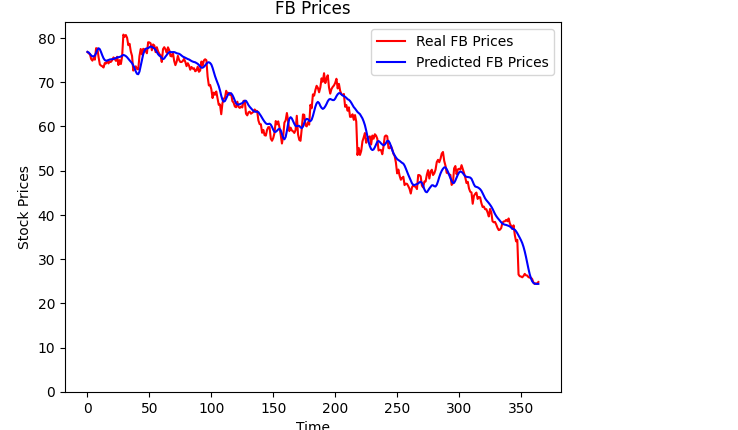
# CHAPTER 3: CONCLUSIONS

## I. Evaluation

To assess the effectiveness of our stock price prediction model, we intend to use a dataset of 10000 day of each stock with known information. This dataset will be divided into two sets: a training set for model training and a test set for performance evaluation. We will evaluate the model's performance after training, with a particular focus on its ability to identify the price changing. We will analyze one main key metric: Accuracy.

## II. Experimental results

For the sake of visualization, here is our demo app:



Although the accuracy of the model on the test dataset exceeds 90%, when tested with various images with different qualities, the accuracy of the model does not meet our expectations. However, it still maintains a level of accuracy above 70%.

## III. Future development

Stock prediction is an area of active research and development in the field of finance and machine learning. Here are some potential future developments in this field:

* Integration of alternative data sources: One potential development is the integration of alternative data sources, such as social media, news articles, and satellite imagery, to improve the accuracy of stock predictions. This approach is known as "alternative data" or "big data" analysis, and it has the potential to provide new insights into market trends and consumer behavior.
* Deep learning models: Another potential development is the use of deep learning models, such as convolutional neural networks (CNNs) and long short-term memory (LSTM) networks, to improve the accuracy of stock predictions. These models are capable of learning complex patterns in time-series data and have shown promising results in other areas of machine learning.
* Reinforcement learning: Reinforcement learning is a type of machine learning that involves training an agent to interact with an environment and maximize a reward signal. One potential application of reinforcement learning in stock prediction is to train agents to make trading decisions based on historical data and real-time market conditions.
* Explainable AI: As machine learning models become more complex, there is a growing need for explainable AI, which refers to the ability to understand and interpret the decisions made by these models. In the context of stock prediction, explainable AI could help investors and traders understand the factors that are driving the model's predictions and make more informed decisions.
* Online learning: Online learning is a type of machine learning that involves updating a model in real-time as new data becomes available. One potential application of online learning in stock prediction is to update the model as new market data becomes available, allowing investors and traders to make more timely and accurate predictions.

# REFERENCES

1. "*Deep Learning for Time Series Forecasting: A Survey*" by Wenqiang Feng et al., published in IEEE Transactions on Neural Networks and Learning Systems in 2020.

2. "*A Brief Survey of Deep Learning for Time Series Prediction*" by Xiaodong Gu et al., published in IEEE Transactions on Neural Networks and Learning Systems in 2018.

3. "*Time Series Forecasting with Deep Learning: A Survey*" by Xiaojun Xu et al., published in IEEE Intelligent Systems in 2020.

4. "*Time Series Forecasting: A Comprehensive Review*" by Viktoriya Semenova and Andrey Gavrilov, published in Applied Soft Computing in 2021.

5. "*Time Series Prediction: A Review of the Literature*" by Jason Brownlee, published on the Machine Learning Mastery website in 2019.

6. "*A Survey on Time Series Forecasting"* by Anastasia Borovykh et al., published in arXiv in 2017.