Housing price prediction using machine learning

Tan, Yawen

2022

HOUSING PRICE PREDICTION USING MACHINE LEARNING

TAN YAWEN

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2022

HOUSING PRICE PREDICTION USING MACHINE LEARNING

TAN YAWEN

School of Electrical and Electronic Engineering

A DISSERTATION SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN SIGNAL PROCESSING

2022

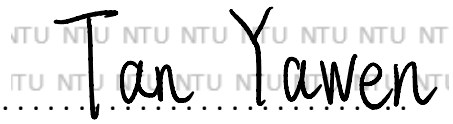
Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

2022/6/7

. . . . . .

Date



. . . . . .

TAN YAWEN

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

8June 2022

. . . . . .

Date

Lihui Chen

. . . . . .

CHEN LIHUI

Authorship Attribution Statement

This thesis **does not** contain any materials from papers published in peer-reviewed journals or from papers accepted at conferences in which I am listed as an author.

2022/6/7

. . . . . .

Date

. . . . . .

TAN YAWEN

Table of Contents

Abstract

Predicting the housing price is an enduring topic since the price change of real estate has a great relationship with the economy, policy, and market. This dissertation explored the use of deep learning models to predict the resale prices of the Housing and Development Board (HDB) flats. In this dissertation, a comprehensive study of the HDB flat transaction data in Singapore has been conducted from 3 aspects: web crawling and analysis, resale price prediction, and performance comparison. Prediction methods were divided into two-phase and single-phase. For the two-phase method, the median resale price per square meter (MRP/m^2) in one month was initially predicted by the Long Short-Term Memory (LSTM) model in the first phase, based on the data from the previous 24 months. Then the second phase models, including LSTM, Multilayer Perceptrons (MLP), and Convolutional Neural Network (CNN) were proposed to predict the resale prices of HDB flats. The first and the second phase were connected by inputting the (MRP/m^2), along with the intrinsic and external attributes of flats, to the second phase models. On the other hand, to judge the effect of the single-phase method, only the intrinsic and external attributes of flats were fed into the second phase models. Grid search with cross-validation was applied to these models. Then, the models with the optimal combination of hyper-parameters were evaluated and compared the performance on the test set. The experiment demonstrated that the two-phase methods outperformed the single-phase ones, where the collaboration of the LSTM and the MLP model achieved the minimum error and the highest accuracy.

Acronyms

AI	Artificial Intelligence
ANN	Artifical Neural Network
ARIMA	Autoregressive Integrated Moving Average
CNN	Convolutional Neural Network
FC	Fully-Connected
GPU	Graphics Processing Unit
G-SVM	Genetic Algorithm with Support Vector Machine
HDB	Housing and Development Board
KNN	K-nearest Neighbors
LSTM	Long Short-term Memory
MLP	Multilayer Perceptrons
MRP/m^2	Median Resale Price per Square Meter
MRT	Metro Rail Transit
MSE	Mean Square Error
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
ROI	Return on Investment

List of Tables

List of Figures

Chapter 1

Introduction

This chapter begins with the motivation of relevant research work and the current obstacles to existing development. With the objectives to be explored, the theme of this dissertation is to use several models based on deep learning techniques to predict housing prices. The organization of this dissertation is presented as well.

1.1 Motivation

Whether for the reasons of marriage, childbirth, or elderly, buying or selling a property is something that every family must face. Real estate often accounts for a large proportion of family assets, so for the buyers, the rise or fall of housing prices in the future always occupies their attention. On the other hand, since changes in housing prices affect economic and even social stability to a certain extent, forecasting the overall trend of housing prices will help the government to introduce timely regulatory policies. While there exist many studies on housing price forecasting, each method has its limitations and somehow lacks of consideration in some aspects. With the increase in computing power in recent years, as well as the collection and disclosure of a large amount of data, there are still demands to more accurately predict the housing prices in a specific country or region. Therefore, this project proposed and evaluates some models based on the deep learning technique to predict the resale price of Housing and Development Board (HDB) flats in Singapore.

1.2 Objectives

The purpose of this project is to develop a model for predicting the resale price of HDB flats in Singapore. The resale price is not only affected by the various attributes of flats, but also related to time factors. Changes in the Median Resale Price per Square Meter (MRP/m^2) of HDB flats, for example, are good indicators of real estate market conditions. Hence, this project predicts the resale prices in two phases: first uses the Long Short-term Memory (LSTM) model to predict the future MRP/m^2, and then combines the prediction results with the intrinsic and external attributes - such as floor area, flat type, and walking distance from Metro

Rail Transit (MRT), etc., of a certain flat unit, to predict the final resale price of this unit through LSTM, or Multilayer Perceptrons (MLP), or Convolutional Neural Network (CNN) model. In addition, by setting up multiple comparison groups, i.e., adding or not adding the predicted results of the first phase - MRP/m^2 at the time of reselling, as the input of the second phase model, whether the two-phase methods outperform the single-phase methods on predicting the resale price of the flats has been discussed, and experimental study has been conducted to justify the analysis.

1.3 Contributions

The contributions of this dissertation is to explore suitable machine learning based approach for house price prediction. It consists of three aspects: feature selections, investigation on suitable deep learning algorithms, and empirical study. First, different types of features (time-dependent or time-independent) used to predict the resale price of an HDB flat were selected and analyzed. The time-dependent feature MRP/m^2 represents the trend of the resales price, while the time-independent features like floor area, flat type, the position of the flat, etc., represent its time-fixed properties. Next, to seek out an appropriate algorithm, two kinds of methods were implemented. As shown in Figure 1.3.1, for the two-phase methods, only the time-dependent features are fed to the first phase model, and all features are used as inputs for the second phase models. In contrast, for the single-phase method, only the time-independent features are inputted into the second phase model. After comparing the predicted accuracies of these two methods, which model is more suitable for which type of features can be concluded.

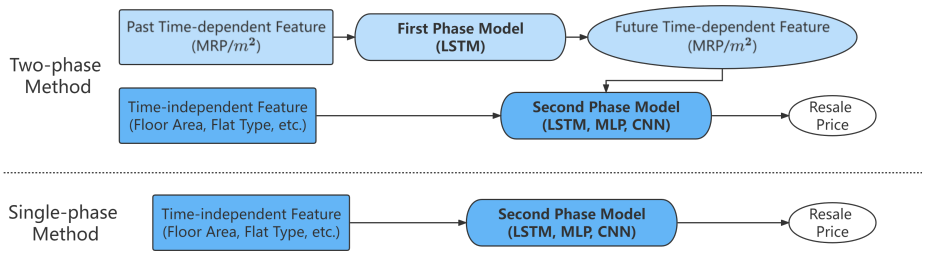


Figure 1.3.1: Two-phase v.s. single-phase methods

1.4 Organization

This report is divided into 6 chapters:

Chapter 1 consists of the motivation and objects of the dissertation.

Chapter 2 discusses the related work about housing price prediction that has been done before.

Chapter 3 describes the background information of HDB, and the main theories about MLP, CNN, and LSTM models.

Chapter 4 elaborates the implementation details of data processing, analysis, model training and prediction.

Chapter 5 provides the result and analysis of the conducted experiments.

Chapter 6 gives a conclusion and recommends some future research.

Chapter 2

Background Information

This chapter mainly provides a detailed explanation of the background knowledge that needs to be understood, beginning with the history and transaction methods of HDB flats in Singapore. The three basic models used in the dissertation: MLP, CNN, and LSTM are discussed next.

2.1 Singapore Housing and Development Board (HDB)

The Housing & Development Board (HDB) is Singapore's public housing authority [1], which provides affordable and quality homes to Singaporeans. The HDB was established on 1 February 1960 with the task to solve the housing crisis in Singapore. Nowadays, the HDB has completed more than 1,000,000 flats and housed the entire nation. About 80% of Singapore's population, spread across 23 towns and 3 estates., are dwelling in these flats.

According to HDB's official website [2], the numbers of residents and flats in each town/estate in Singapore as of 31 March 2018 are shown in Table 2.1.1.

To buy new HDB flats, the eligible buyers can apply during sales launch and wait for a computer ballot results that determine the queue positions. When their turns come, the HDB will invite them to book a flat and sign the agreement for lease. After the flat is completed, the applicants need to make full payment and collects the keys to their new flats.

The owners of HDB flats must fulfill a set of eligibility conditions, according to the mode of purchase, before selling their flats. Usually, the eligibility conditions contain the minimum occupation period, ethnic integration policy, and Singapore permanent resident quota. The prices of such resale flats are influenced by the market economy as well as the government's interventions. And that's why stakeholders, including economists, real estate practitioners, and policymaker, are interested in predicting resale flat prices.

The purpose of this dissertation is to analyze the factors that affect the resale price of HDB flats, and develop models to predict the future resale price through deep learning techniques.

Region	Town/Estate	Number of HDB Residents	Number of Flats
North	Sembawang	73,500	26,834
	Woodlands	242,500	68,153
	Yishun	196,600	62,786
North-East	Ang Mo Kio	143,800	50,733
	Hougang	179,500	54,328
	Punggol	187,800	49,909
	Sengkang	212,100	66,605
	Serangoon	68,800	21,634
East	Bedok	194,700	61,828
	Pasir Ris	108,400	29,654
	Tampines	232,700	68,812
West	Bukit Batok	114,000	40,612
	Bukit Panjang	121,100	35,325
	Choa Chu Kang	169,000	48,900
	Clementi	72,300	26,727
	Jurong East	78,000	23,897
	Jurong West	258,100	74,301
Central	Bishan	63,200	20,072
	Bukit Merah	145,700	54,423
	Bukit Timah	8,400	2,555
	Central Area	27,700	12,316
	Geylang	87,300	30,304
	Kallang/Whampoa	106,900	39,194
	Marine Parade	21,600	7,862
	Queenstown	82,500	32,678
	Toa Payoh	105,000	37,900

Table 2.1.1: The numbers of residents and flats is each town/estate in Singapore as of 31 Mar, 2018.

2.2 Deep Learning and Neural Network

Deep learning is a subset of machine learning, which is in turn a subset of Artificial Intelligence (AI). AI is a technique that enables a machine to mimic human behavior. Machine learning is a technique to achieve AI through algorithms trained with data. And finally deep learning is a type of machine learning inspired by the structure of the human brain, which is called neural network. At the heart of neural networks are several key principles: alternation of linear and nonlinear processing units called layers, and back propagation to adjust all the parameters in the neural network at once. Since 2010, massive data sets were within reach, due to the emergence of Internet-based companies, serving hundreds of millions of users online. In addition, the dissemination of cheap, high-quality sensors and cheap data storage, especially in the form of Graphics Processing Units (GPUs), readily enables large-scale computing power. For these two reasons, neural network algorithms that once seemed computationally infeasible have become popular. In this section, I will briefly introduce some neural networks that were used in this project.

2.2.1 Multilayer Perceptrons (MLP)

Multilayer Perceptron (MLP) is a feedforward ANN model that contains at least one hidden layer. Figure 2.2.1 shows a MLP with a single hidden layer with p hidden units. This MLP has d input nodes and q output nodes. Except for the input nodes, each node is a neuron using a non-linear activation function ϕ to exploit the potential of the multi-layer architecture.

To express the MLP in Figure 2.2.1 in mathematical form, denote the input layer by $\mathbf{X} \in \mathbb{R}^{n \times d}$ that represents n samples with d input features. The output of

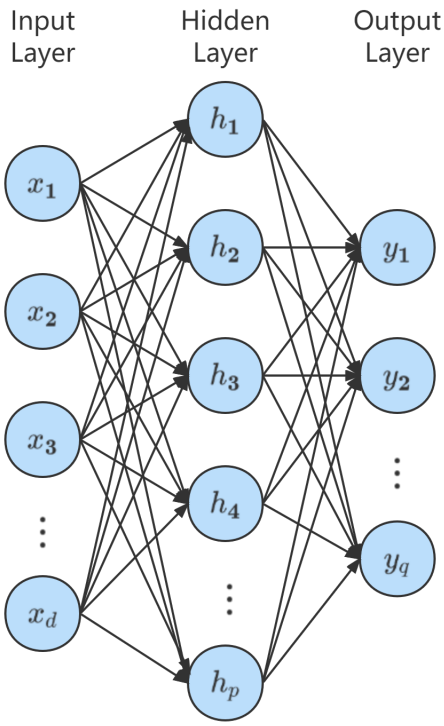


Figure 2.2.1: A MLP with a single hidden layer

the hidden layer can be represented by $\mathbf{H} \in \mathbb{R}^{n \times p}$, and the output of this MLP is $\mathbf{Y} \in \mathbb{R}^{n \times q}$:

$$\mathbf{H} = \phi(\mathbf{X}\mathbf{W}_1 + \mathbf{b}_1) \tag{2.2.1}$$

$$\mathbf{Y} = \mathbf{H}\mathbf{W}_2 + \mathbf{b}_2 \tag{2.2.2}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times p}$ and $\mathbf{b}_1 \in \mathbb{R}^{1 \times p}$ are the weight and bias parameters for the hidden layer; $\mathbf{W}_2 \in \mathbb{R}^{p \times q}$ and $\mathbf{b}_2 \in \mathbb{R}^{1 \times q}$ are the weight and bias parameters for the output layer.

To build a more general MLP, it is capable to stack more hidden layers to produce a more expressive model. Furthermore, there are many kinds of activation functions, such as *ReLU*, *sigmoid* and *tanh*, each has its pros and cons. Choices

to use which activation function should be made on a specific application.

2.2.2 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) refers to a class of neural networks that contain convolutional layers, which is usually applied to graph-structured data. Compared to MLP, it requires fewer parameters and thus operates more efficiently. Next, I will introduce the basic elements that constitute the backbone of CNN: convolutional layers and pooling layers.

2.2.2.1 Convolutional Layer

Although called a convolutional layer, it actually performs a two-dimensional cross-correlation operation on input tensors and convolution kernel weights, as shown in Figure 2.2.2. The convolution window starts at the upper left corner of the input tensor, and slides from left to right and top to bottom. At each position where the convolution window slides, the partial tensors contained in the window are multiplied element-wise by the convolution kernel weights. Then the multiplication results are summed up to obtain a single scalar value, which is the output at this position after adding a scalar bias.

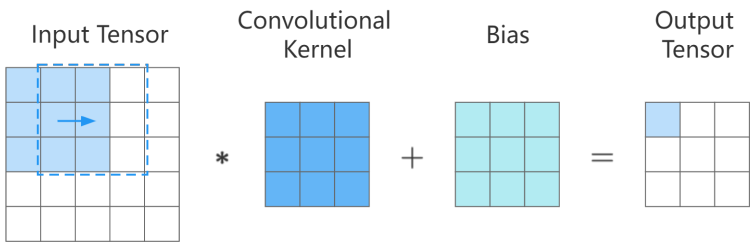


Figure 2.2.2: The 2D cross-correlation operation

The output shape of the convolutional layer depends on the input shape, the shape of the convolution kernel, as well as the padding and stride. padding is to pad elements (usually 0) at the border of the input image. stride is the number of elements that the convolution window slides each time.

Assume the input shape is $(i_h \times i_w)$, the shape of the convolution kernel is $(k_h \times k_w)$, the padding elements for rows and columns are p_h and p_w, and, the horizontal and vertical strides are s_h, s_w respectively. Then the output shape $(o_h \times o_w)$ is [3]:

$$(o_h \times o_w) = \lfloor (i_h - k_h + p_h + s_h)/s_h \rfloor \times \lfloor (i_w - k_w + p_w + s_w)/s_w \rfloor \qquad (2.2.3)$$

2.2.2.2 Pooling Layer

Similar to the convolutional layers, a pooling operator consists of a fixed-shape window called a pooling window, which slides over all regions of the input tensor according to its stride, computing an output for each position. The pooling layer contains no parameter so it is different from the 2D cross-correlation operation. Instead, the maximum or average values of all elements in the pooling window are calculated to reduce the redundancy of parameters and avoid the over-fitting problem.

2.2.3 Long Short-term Memory (LSTM)

Long short-term memory (LSTM) is a network that aims to alleviate the problem of numerical instability of Recurrent Neural Network (RNN). Therefore, to understand LSTM, it is necessary to describe RNN first.

2.2.3.1 RNN

Considering a sequence $\{x_1, x_2, \cdots, x_t, \cdots\}$ where $t \in \mathbb{Z}^+$ is the time step. It is possible to predict x_t according to the past observations $\{x_1, x_2, \cdots, x_{t-1}\}$, which in mathematical form is [3]:

$$x_t \sim P\left(x_t \,|x_{t-1}, x_{t-2}, \cdots, x_1\right) \tag{2.2.4}$$

Such method of prediction will encounter a problem that the number of input data $x_1, x_2, \cdots, x_{t-1}$ is depend on the time step t. That is, the inputs will increase with the accumulation of the data. So, an approximation is needed to make this calculation tractable. RNN use a strategy to keep some summary h_t of the past observations, which is called hidden variable. Then the prediction of x_t changes into $\hat{x}_t = P\left(x_t \,|h_{t-1}\right)$. At every time step t, h_t shall be updated according to the current input data x_t and the previous hidden variable h_{t-1}:

$$h_t = f\left(x_t, h_{t-1}\right) \tag{2.2.5}$$

Because the same definition used in the current hidden variable h_t and the previous one h_{t-1}, the calculation is recurrent. Hence, the neural network based on such recurrent calculation is called recurrent neural network, and the layer that performs recurrent calculation are called recurrent layer.

Considering n sequence samples at time step t: $\mathbf{X}_t \in \mathbb{R}^{n\times d}$, where d is the input dimensions. Then $\mathbf{H}_t \in \mathbb{R}^{n\times h}$, where h is the number of neurons in the recurrent layer, represents all the hidden variables at time step t. The calculation of the hidden variable matrix $\mathbf{H}_t$ can be seen as: Combining the current input $\mathbf{X}_t$ and the previous hidden variable matrix $\mathbf{H}_{t-1}$ by a fully-connected layer with the

activation function ϕ:

$$\mathbf{H}_t = \phi\left(\mathbf{X}_t\mathbf{W}_{xh} + \mathbf{H}_{t-1}\mathbf{W}_{hh} + \mathbf{b}_h\right) \tag{2.2.6}$$

where $\mathbf{W}_{xh} \in \mathbb{R}^{d\times h}$ and $\mathbf{W}_{hh} \in \mathbb{R}^{h\times h}$ are the weight parameters; $\mathbf{b}_h \in \mathbb{R}^{1\times h}$ is the bias parameter.

As for the output layer, it is similar to the MLP: A fully-connected layer where the hidden variable $\mathbf{H}_t$ will be fed to obtain the output $\mathbf{Y}_t$ at time step t:

$$\mathbf{Y}_t = \mathbf{H}_t\mathbf{W}_{hq} + \mathbf{b}_q \tag{2.2.7}$$

where $\mathbf{W}_{hq} \in \mathbb{R}^{h\times q}$ is the weight parameter and $\mathbf{b}_q \in \mathbb{R}^{1\times q}$ is the bias parameter. q is the number of neurons in the output layer.

Figure 2.2.3 shows the computational graph of the RNN that contains 3 adjacent time steps.

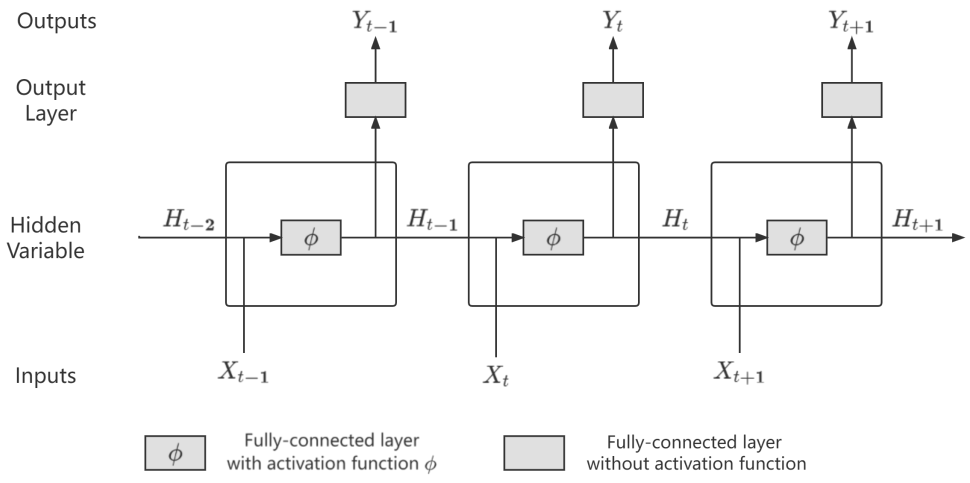


Figure 2.2.3: The computational graph of the RNN

2.2.3.2 LSTM

Models with hidden variables may encounter some situations: When the early inputs are highly significant, a mechanism is needed to preserve the long-term information; On the other hand, for some inputs that are irrelevant to prediction, the mechanism is able to ignore them even if they are short-term. One of the approaches to solve this problem is the LSTM [4].

LSTM introduces memory cell $\mathbf{C}_t$ at time step t, which is designed to record additional information. Memory cells are gating, meaning that LSTM has dedicated mechanisms - which is called output gate $\mathbf{O}_t$, input gate $\mathbf{I}_t$ and forget gate $\mathbf{F}_t$, to determine when the data in memory cell should be output, input or reset.

At the time step t, both the input at the current time step $\mathbf{X}_t$, and the hidden variable at the previous time step $\mathbf{H}_{t-1}$ are fed into 3 separate fully connected layers with sigmoid activation functions to compute the values of the three gates $\mathbf{O}_t, \mathbf{I}_t, \mathbf{F}_t \in \mathbb{R}^{n\times h}$. Hence, the results are ranged from 0 to 1. The mathematical expressions are [3]:

$$\mathbf{O}_t = \sigma\left(\mathbf{X}_t\mathbf{W}_{xo} + \mathbf{H}_{t-1}\mathbf{W}_{ho} + \mathbf{b}_o\right) \tag{2.2.8}$$

$$\mathbf{I}_t = \sigma\left(\mathbf{X}_t\mathbf{W}_{xi} + \mathbf{H}_{t-1}\mathbf{W}_{hi} + \mathbf{b}_i\right) \tag{2.2.9}$$

$$\mathbf{F}_t = \sigma\left(\mathbf{X}_t\mathbf{W}_{xf} + \mathbf{H}_{t-1}\mathbf{W}_{hf} + \mathbf{b}_f\right) \tag{2.2.10}$$

where $\mathbf{W}_{xo}, \mathbf{W}_{xi}, \mathbf{W}_{xf} \in \mathbb{R}^{d\times h}$ and $\mathbf{W}_{ho}, \mathbf{W}_{hi}, \mathbf{W}_{hf} \in \mathbb{R}^{h\times h}$ are the weight parameters; $\mathbf{b}_o, \mathbf{b}_i, \mathbf{b}_f \in \mathbb{R}^{1\times h}$ are the bias parameters.

The memory cell $\mathbf{C}_t$ is updated by forgetting parts of the previous memory $\mathbf{C}_{t-1}$ and remembering some of the contents from the candidate memory cell

$\tilde{\mathbf{C}}_t \in \mathbb{R}^{n\times h}$, which uses *tanh* as the activation function:

$$\tilde{\mathbf{C}}_t = \tanh\left(\mathbf{X}_t\mathbf{W}_{xc} + \mathbf{H}_{t-1}\mathbf{W}_{hc} + \mathbf{b}_c\right) \tag{2.2.11}$$

where $\mathbf{W}_{xc} \in \mathbb{R}^{d\times h}$ and $\mathbf{W}_{hc} \in \mathbb{R}^{h\times h}$ are the weight parameters; $\mathbf{b}_c \in \mathbb{R}^{1\times h}$ is the bias parameter.

And the update of $\mathbf{C}_t$ is controlled by the forget gate $\mathbf{F}_t$ and the input gate $\mathbf{I}_t$:

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \tilde{\mathbf{C}}_t \tag{2.2.12}$$

where $\odot$ means element-wise multiplication.

Next, the hidden variable $\mathbf{H}_t$ is influenced by the output gate $\mathbf{O}_t$ and *tanh* of the memory cell $\mathbf{C}_t$:

$$\mathbf{H}_t = \mathbf{O}_t \odot \tanh(\mathbf{C}_t) \tag{2.2.13}$$

Finally, like the RNN, the hidden variable $\mathbf{H}_t$ is fed into a fully-connected layer to obtain the output $\mathbf{Y}_t$, as described in Eqn. (2.2.7).

The computational graph of the LSTM at time step t is shown in Figure 2.2.4.

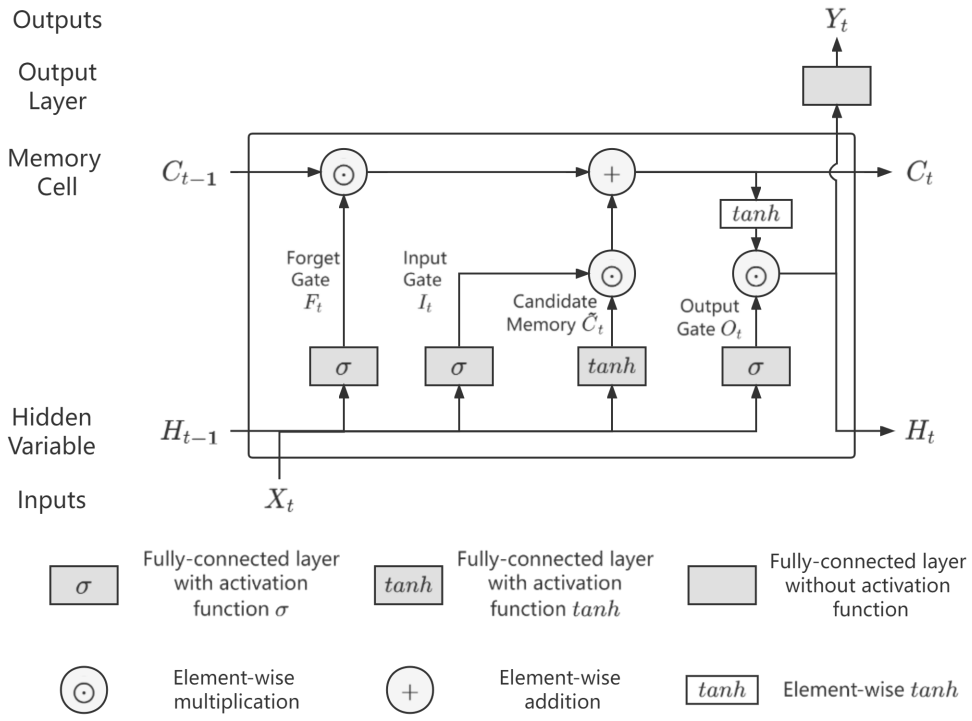


Figure 2.2.4: The computational graph of the LSTM

Chapter 3

Literature Review

This chapter introduces the past research on housing price prediction, which is divided into methods to predict the exact housing prices and methods to predict the future trend of housing price statistics, follow by the efforts on combining these two kinds of methods. At the end of this chapter, the performances of different methods are generalized and compared.

3.1 Housing Price Prediction

With the rapid development of artificial intelligence technology, related basic data analysis technology is widely used in all walks of life, and has achieved good results in many fields. House price is an index that dynamically describes the average change trend of various real estate prices in a region. It is not only an important indicator to grasp the prosperity of the real estate market from a macro perspective, but also a reference basis for real estate investment from a micro perspective. Based on the above analysis, the prediction of house prices is very important and has become a hot issue concerned by many scholars. There are generally many factors affecting house price prediction, and generally, only some aspects of data can be collected for processing, so it is often difficult to achieve the expected effect. However, many researchers continue to try and study in this field, and have achieved a series of excellent results. From traditional machine learning to deep learning developed in recent years, with the introduction of various technical means, the field of house price prediction has been continuously developed and broken through. There is evidence that house prices are predictable to some extent [5]. Therefore, it is very useful to determine which prediction model can best capture the future trend of house prices.

In the following parts, I mainly focus on the history and application of two kinds of algorithms in the field of house price prediction, and analyze the advantages and existing problems of various algorithms, which will be of great help to my experiments in the future.

3.1.1 Prediction of the Exact Housing Price

Before the rise of deep learning, some researches on house price prediction mainly used traditional machine learning methods, such as multiple linear regression, K-nearest Neighbors (KNN) regression, gradient boosting regression, and so on. These algorithms have simple models, few parameters, and low computational complexity. They often have good prediction results for projects with small data sets. Later, with the development of deep learning technology, more and more scholars adopted deep learning methods in the field of house price prediction, such as Artifical Neural Network (ANN) network and Convolutional Neural Network (CNN). These algorithms using deep learning have good feature extraction ability, can effectively count and distinguish small changes between data, and often have good results in the processing of massive data, so they are favored by more and more research scholars.

Gu Jirong, et al [6] combine the traditional genetic algorithm with the support vector machine, to propose a hybrid house price prediction method (G-SVM). Compared with the network algorithm, the genetic algorithm takes less time and has better performance. Therefore, the genetic algorithm is used to optimize the parameters of the support vector machine at the same time. Through an example, it is verified that the G-SVM method has better house price prediction ability. Peng et al. [7] used XGBoost algorithm to analyze the data of second-hand houses in Chengdu. Through experiments, they proved that XGBoost model is more robust than other machine learning methods in house price prediction. This is because XGBoost adds a regular term to the cost function to control the complexity of the model and prevent overfitting. Boek and Moller [8] chose to use the dynamic model to predict the real estate prices of all states in the United States, which makes his model parameters change and can change with time and place,

effectively improving the accuracy of the whole prediction. Phan [9] proposed a new model integration machine learning method, and used a stepwise method in the training process to make the final prediction result more accurate.

3.2 Prediction of the Future Trend of Housing Price Statistic

The traditional time series model and grey correlation model generally have the problem of low prediction accuracy. In contrast, MLP neural network model has higher prediction accuracy, However, the disadvantages are also obvious. The whole training process is relatively long, and the global optimal value cannot be reached if the parameters are not set well. Like the MLP model, the LSTM model is also a kind of neural network model. It is different from the MLP in that it is specially designed for time-series data, so it has advantages in dealing with time series data such as house price index. LSTM model has been successfully applied to wind speed forecasting, energy load forecasting, and stock market income forecasting. This indicates that it has unique advantages in processing time-series data. At present, many researchers have applied LSTM to house price prediction.

Autoregressive Integrated Moving Average (ARIMA) model is an analysis method for time series identification, estimation, test, and prediction. ARIMA model can be used to simulate stationary series. After changing the unstable house price index series into a stationary series through difference, a differential autoregressive moving average model is proposed. Temur et al. [10] combine the ARIMA model with the LSTM model to develop a new house price prediction model. At the same time, the model is adjusted and optimized in combination with the evaluation index, so that it has higher accuracy in the final prediction result.

3.3 Combination of models

Yu et al. [11] developed a house price prediction model based on deep learning, which combines the LSTM and the CNN networks, considering not only the structural characteristics of data, but also the temporal characteristics, compared with the performance of the traditional autoregressive moving average model, it has better prediction effect. You et al. [12] used the most advanced visual features, the recursive neural network is used to predict the real estate price. This method takes into account the customer's subjective visual feeling, directly seizes the factors affecting the customer's choice from the psychological level, and plays an important role in the house price prediction, so the model has also achieved good results.

Pourseed et al. [13] considered that the key factors affecting house prices are multifaceted, not only the superficial statistical data, but also some potential factors. Therefore, they use the convolution data of some potential features of the house, and then develop a new neural network to predict the internal features of the house, including the convolution data. They also developed a complete automatic evaluation framework to support the input of various photo data, and achieved good prediction results. Xiaochen Chen [14] also combined the autoregressive comprehensive moving average model and LSTM network to build a prediction model. The combined model combines the advantages of the two models. After parameter optimization, it not only has a better prediction effect than a single model, but also has good robustness. Compared with the traditional machine model, it has better prediction accuracy in time series.

3.4 Summary

Compared with traditional machine learning, deep learning has many parameters and does not need feature design. Before using machine learning classifiers, we often need to do some data preprocessing, or feature design and extraction, so that we can make better use of the characteristics of the collected data. Convolution filter in deep learning is a good feature lifter, It can extract high-level semantic features of data and find subtle differences between data. Therefore, in the era of big data, the prediction effect of deep learning is often better than traditional machine learning. House price prediction is an important social topic related to people's livelihood. After the research and comparison of historical literature, we'd like to explore a new approach for the prediction by making use of time-dependent and time-independent features separately in machine learning models.

Chapter 4

Empirical Study

This chapter delves into details on the overall procedures of data processing, analysis, model training and prediction. Starts with a workflow that describes the methodology to complete the prediction of housing price, the chapter then provides information about how the data is collected and preprocessed, followed by the implementation of hyper-parameters tuning, model training and evaluation.

4.1 Flow Diagram

Figure 4.1.1 shows the overall flow diagram in this project. Starting with data sets being downloaded from *data.gov.sg*, the two-phase method was then implemented: LSTM was first used to predict the Median Resale Price per Square Meter (MRP/m^2) in Singapore for a month in the future, based on the MRP/m^2 of the previous past 24 months. After that, the predicted MRP/m^2 of the month when a flat was resold, along with its intrinsic and external attributes, would be fed into the second phase model (LSTM, MLP, or CNN) to predict the resale prices of the flat. Also, in order to demonstrate the effectiveness of the first phase model (LSTM), the single-phase method was established by removing the MRP/m^2 from the input features and directly predicting the resale prices by the second phase models. Finally, the performances of the two methods are compared.

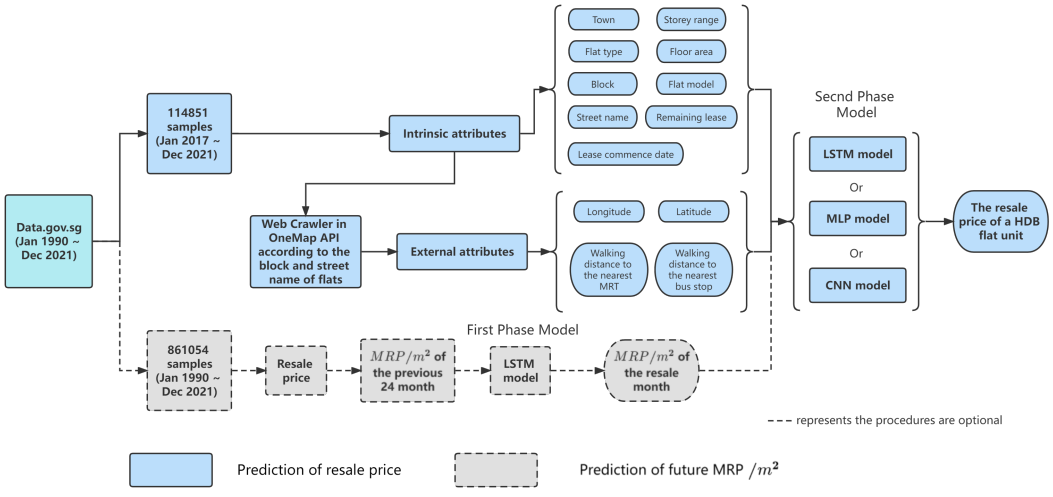


Figure 4.1.1: Flow diagram

4.2 Data Preparation

This section introduces the used data source in the dissertation, and how the data was prepared for predictions. Data was analyzed as well to roughly display the relevance between various attributes and the resale prices.

4.2.1 Data Source

The selection of dataset has a great impact on the prediction results of the model. The samples should reflect the changes in housing prices as objectively and truthfully as possible. *Data.gov.sg* is Singapore's open data portal that aims to provide one-stop access to the government's publicly-available data. It is a trusted source to obtain data.

This article used the dataset [15] published on *data.gov.sg* by the HDB. The dataset collected the resale prices of flats from January 1990 onwards, and provided specific information about each flat, whose details are shown in Table 4.2.1. In total, the dataset had 861,054 samples covering all 23 towns and 3 estates in Singapore. Figure 4.2.1 shows the data of three samples as an example.

```
1  month,town,flat_type,block,street_name,storey_range,floor_area_sqm,flat_model,lease_commence_date,remaining_lease,resale_price
2  2017-01,ANG MO KIO,2 ROOM,406,ANG MO KIO AVE 10,10 TO 12,44,Improved,1979,61 years 04 months,232000
3  2017-01,ANG MO KIO,3 ROOM,108,ANG MO KIO AVE 4,01 TO 03,67,New Generation,1978,60 years 07 months,250000
4  2017-01,ANG MO KIO,3 ROOM,602,ANG MO KIO AVE 5,01 TO 03,67,New Generation,1980,62 years 05 months,262000
```

Figure 4.2.1: Data of 3 Samples

Name	Data Type	Description
Month	String	The month that a flat was resold
Town	String	The town where a flat resided

Flat type	String	According to the number and structure of the rooms, flats were classified into 7 types: 1 room, 2 rooms, 3 rooms, 4 rooms, 5 rooms, executive and multi-generation. The specific differences between different types can be seen in Appendix A Table A-1.
Block	Number	The block number of the a flat
Street name	String	The street where an HDB flat was located
Storey range	String	The range of the floor where a flat is located.
Floor area in square meters	number	The area (measured as square meters) taken up by a flat.
Flat Model	String	Flats were categorized into 20 models, such as standard, improved and new generations, etc. This factor served as a supplement to the flat type. See Table A2 in Appendix A for specific explanations.
Lease Commence Data	Number	The starting year of the 99-year lease ownership.
Remaining Lease	String	The number of years and months left before the lease ends
Resale Price	number	The price of a flat

Table 4.2.1: Data contained in each sample

4.2.2 Data Preparation for Prediction of future $\mathrm{MRP}/\mathrm{m}^2$

To predict the future $\mathrm{MRP}/\mathrm{m}^2$, only the month, floor area, and the resale price were needed, other data in a sample was ignored. For each sample, divide its resale price by the floor area to determine its resale price per square meter. Sort the samples by month, then the $\mathrm{MRP}/\mathrm{m}^2$ of the flats monthly could be easily obtained. Figure 4.2.2 shows the change in $\mathrm{MRP}/\mathrm{m}^2$ in Singapore over the past 3 decades. It can be observed that during 1993-1996, 2008-2013, and 2021 onwards, there were large increases, while during 1997-1999 and 2013-2015, there were significant declines. And the values for other periods fluctuated.

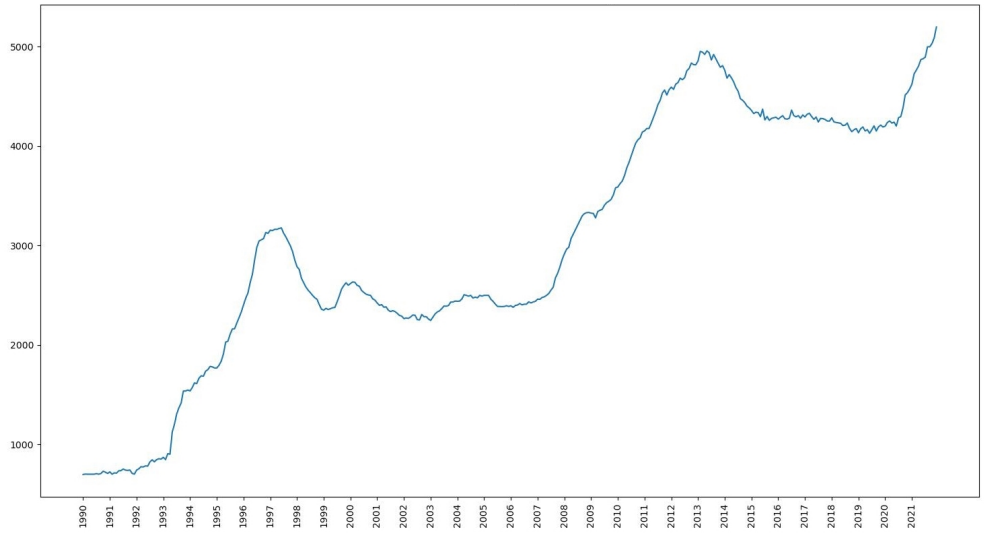


Figure 4.2.2: $\mathrm{MRP}/\mathrm{m}^2$ monthly 1990 onwards

$\mathrm{MRP}/\mathrm{m}^2$ could be viewed as a time series $\{x_t\}$, where $t = 1, 2, \cdots, 384$ represented the month order. Although the specific value of x_t might alter, the dynamics of the sequence itself did not change for a short period - as if it had inertia. Therefore, future values of x_t could be predicted from observations of its history. That's where a deep learning model came into play.

January 2017 was the division time to split the training and test dataset: This generated a training sequence of length 324 (324 time-steps) and a test sequence of length 60.

In this project, future data was predicted based on historical data from the past 24 months. So, the model ought to process a mini-batch sequence of length 24 at a time, which meant the training and test sequences shall both be divided into subsequences of length 24. Random sampling was adopted to guarantee each sample is an arbitrarily captured subsequence on the original long sequence.

The subdivision schematic is shown in Figure 4.2.3. A long sequence of length 30 could be divided into 6 subsequences. Thus, the training and test sequences would bring about 300 and 36 subsequences, respectively.

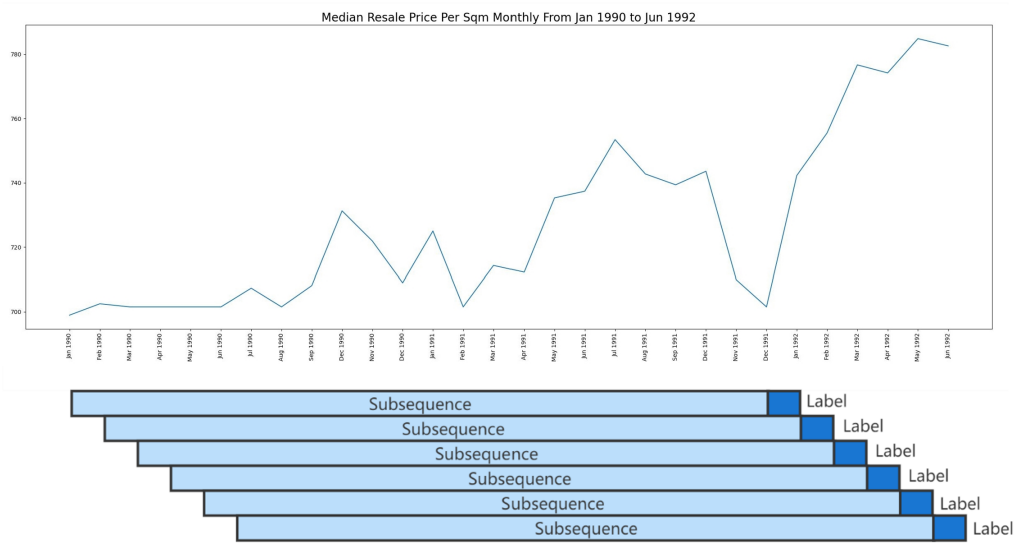


Figure 4.2.3: The subdivision schematic

4.2.3 Data Preparation for Prediction of Resale Price

Housing prices depend on many factors, including those that can be quantified and those that cannot. Unquantifiable factors such as national policies, cultural

background and personal preferences, are difficult to be collected and considered as features of the sample data. Therefore, this dissertation only focused on the quantitative factors that were provided by the raw dataset, as well as some external attributes that could be obtains using web crawler.

4.2.3.1 Data Acquisition through Web Crawler

It can be seen from Figure 4.2.1 that the address of a flat in the raw data set is shown by its block number and street name. In order to use such information as features, it needs to be quantified into latitude and longitude. By using python's Requests library [16] to send an HTTP request to catch the data from OneMap Application Programming Interface (API), the latitude and longitude of each flat could be easily obtained.

OneMap [17] is the authoritative national map of Singapore with detailed and timely updated information developed by the Singapore Land Authority. Its Web service API provides developers with HTTP interfaces through which developers can use various types of geographic data services such as search, route planning, and distance measurement. By searching a specific address in OneMap, the information of the nearest MRT/LRT and Bus Stop could be obtained. After that, the walking distances to the nearest MRT/LRT and Bus Stop could be gained by implementing the route planning.

However, when the distance between one flat and its nearest MRT/LRT or Bus Stop was too far, OneMap could not plan the walking route as walking to the destination was impractical. So, it was necessary to estimate the distance when route planning did not work. This was done by the Haversine formula that determines the distance d between two points on a sphere given their latitudes and

longitudes:

$$d = 2r \sin^{-1} \sqrt{\sin^2\left(\frac{\theta_2 - \theta_1}{2}\right) + \cos(\theta_1)\cos(\theta_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)} \tag{4.2.1}$$

where θ_1, θ_2 and λ_1, λ_2 are the latitudes and longitudes of two points in radians, respectively, and r is the radius of the earth.

All the data acquisition and calculations were done by python. Part of the output of the script can be seen in Figure 4.2.4 showing the month when the flats were resold, the address, retrieved latitudes and longitudes of the flats, the names of their nearest MRTs/LRTs and Bus Stops, as well as the corresponding walking distances. The data, which is described in Table 4.2.2, was treated as the external attributes of flats.

```
1  Month,Block,Street name,Latitude,Longitude,MRT name,Walking distance to MRT,Bus Stop name,Walking distance to Bus Stop
2  2017-01,406,ANG MO KIO AVE 10,1.36200453938712,103.853879910407,ANG MO KIO MRT STATION,1117,OPP CHRIST THE KING CH,92
3  2017-01,108,ANG MO KIO AVE 4,1.37096635222625,103.838201940326,MAYFLOWER MRT STATION,266,MAYFLOWER STN EXIT 5,166
4  2017-01,602,ANG MO KIO AVE 5,1.38070883044887,103.835368226602,LENTOR MRT STATION,726,AFT LENTOR STN EXIT 4,202
```

Figure 4.2.4: Data Acquisition Outputs Using the Python Script

Name	Data Type	Description
Latitude and Longitude	Number	The geographical coordinates of the flat.
Walking Distance to the Nearest MRT/LRT	Number	The length of the route (measured as meters) required to walk from the flat to the nearest MRT/LRT.
Walking Distance to the Nearest Bus Stop	Number	The length of the route (measured as meters) required to walk from the flat to the nearest Bus Stop.

Table 4.2.2: The external attributes of an HDB flat acquired by Web Crawler

4.2.3.2 Features Encoding

Some of the data in the raw dataset are words, such as town, flat type, flat model, etc. Some other factors contain numbers plus other characters, like the month, storey range, and remaining lease. In order for these factors to become machine-readable and operable, they need to be converted into numerical valued features.

For the **month** data, take January 1990 as the base month, and number it as 0. Then number the subsequent months accordingly in chronological order, that is, February 1990 is No.1, March 1990 is No.2, and until December 2021 is No.383. These numbers are called encoded months and they are used in the prediction of future MRP/m^2.

For the **town**, **flat type** and **flat model** data, each unique category has its own numerical index, as shown in Table B1, B2 and B3 in Appendix B. However, feeding these indices directly to a neural network might increase the difficulty of learning. To solve this problem, one-hot encoding is usually used to map different classes to mutually different unit vectors. For example, there are 26 towns in Singapore, so their corresponding numerical indices range from 0 to 25. Assuming that the index of a town is the integer i, then create an all-zeros vector of length 26 and set the i^{th} element to 1. This vector is the one-hot vector representing this town.

For the data of **storey range**, take the average storey as its feature. For instance, the storey range for the first sample in Figure 4.2.1 is "10 to 12", then take 11 as its storey.

For the data of **remaining lease**, convert it to a value that counts in months. Taking the sample mentioned above as an example, the remaining lease is "61 years 04 months", then convert it to $61 \times 12 + 4 = 736$.

After encoding, the feature dimension of each sample became 62 when including the MRP/m^2, and 61 when excluding the MRP/m^2.

4.2.3.3 Data Analysis

Before utilizing the data to train some models, some analysis was done to visualize the relevance between the various features and the resale prices in all the samples.

First, with the help of Mapbox [18], a powerful tool that can provide map and location data to developers, the geographic distribution of the 114,851 samples used in predicting resale prices is displayed in Figure 4.2.5. The values of the resale prices are represented by different colors, just as the bar shows on the right side of Figure 4.2.5.

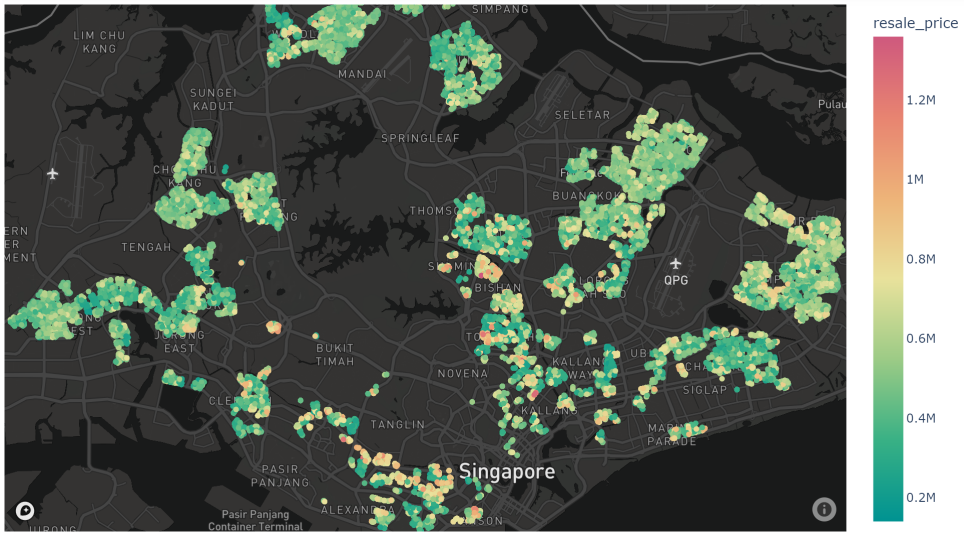


Figure 4.2.5: Distribution of resale prices on map.

It can be seen from Figure 4.2.5 that the sample data is distributed in blocks, spread across different towns/estates in Singapore, and the resale prices in different regions vary. For instance, resale prices in the Central and South are generally higher than that in the West and North. This may also be proved by the plot of

town v.s. mean value of resale prices in the specific town, as shown in Figure 4.2.6. The mean values of resale prices in Bukit Tmah, Bishan (Central), and Queens Town (South) are much higher compared to that in Jurong West (West) and Sembawang (North).

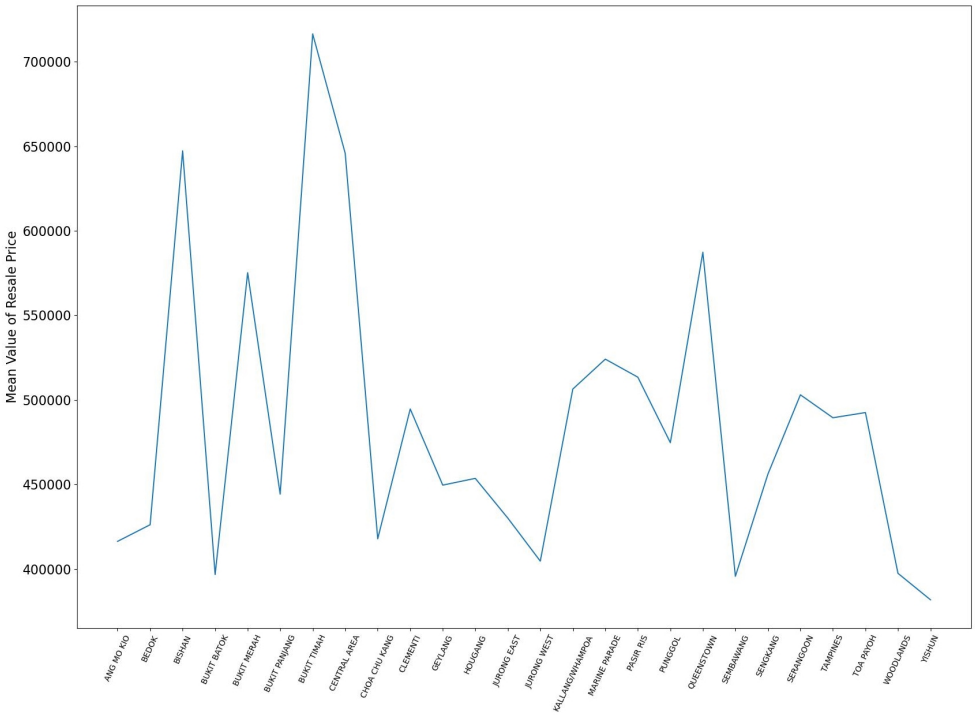


Figure 4.2.6: Town v.s. mean value of resale prices in the specific town.

Next, the relevance of the flat types and the resale prices are demonstrated by averaging the resale price in different categories, as shown in Figure 4.2.7. The results were much in line with expectations: units with more rooms were more expensive.

After that, The degree of correlation between an individual numerical feature and the resale price is measured by the Pearson Correlation Coefficient which

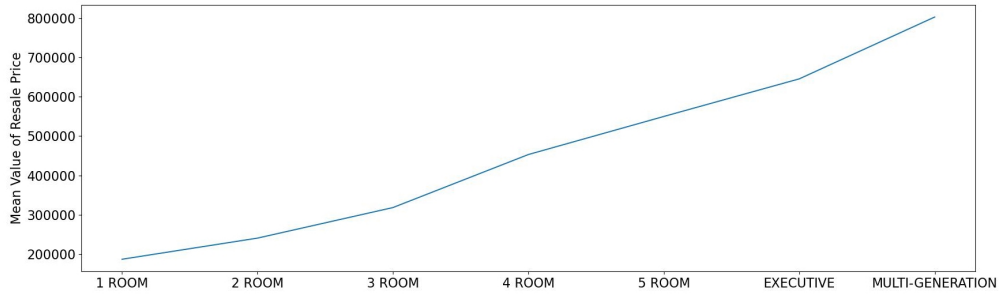


Figure 4.2.7: Flat type v.s. mean value of resale prices for the specific flat type.

ranges from -1 to 1:

$$Pearson = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{4.2.2}$$

where $\overline{x}$ and $\overline{y}$ are the average values of the variables x and y.

The results are shown in the descending order of the absolute values, as displayed in Figure 4.2.8. If the coefficient is positive, the feature and the resale price are positively correlated; if the coefficient is negative, the feature and the resale price are negatively correlated. Also, the absolute value of the coefficient represents the degree of correlation: the closer the absolute value is to 1, the stronger the linear correlation.

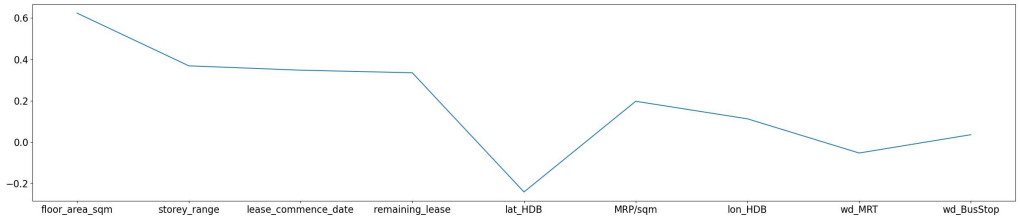


Figure 4.2.8: Pearson correlation of different features w.r.t the resale price.

Figure 4.2.8 shows that the resale price has the strongest correlation with its floor area, and the weakest correlation with the walking distance to the nearest bus stop. This is probably because the bus transportation in Singapore is very

well developed and any HDB flat is not far away from its nearest bus stop.

4.3 Model Training

This section introduces the first phase model used in the prediction of the future MRP/m^2, and the second phase models for predicting the resale prices. Also, how to tune hyper-parameters and how to train and evaluate the models are explained.

4.3.1 Data Normalization

Data normalization is important for deep learning. Since normalization is a linear transformation, it will not 'damage' the data by changing the original numerical order. On this premise, it can improve the convergence speed and avoid some numerical problems when training the model. In addition, the scales of different features might be different. For example, the storey range was always a single- or double-digit number, while the remaining lease was usually ten times more than that. When using a model to make predictions, the features with large scales would play a decisive role, while the roles of the small scale features might be ignored. To eliminate the influence of scale differences, the features needed to be normalized.

Several commonly used normalization methods are min-max, z-score and log. Min-max normalization was chosen in this dissertation. The formula is shown in Eqn. (4.3.1).

Transform the sequence $\{x_1, x_2, \cdots, x_n\}$:

$$y_i = \frac{x_i - \min\limits_{1 \le j \le n} \{x_j\}}{\max\limits_{1 \le j \le n} \{x_j\} - \min\limits_{1 \le j \le n} \{x_j\}} \tag{4.3.1}$$

then the new sequence $\{y_1, y_2, \cdots, y_n\} \in [0, 1]$.

When predicting the future MRP/m^2, min-max normalization was applied to each subsequence and the corresponding label. When predicting the resale prices, min-max normalization was applied, after features encoding, to the labels and the following features: MRP/m^2, storey range, remaining lease, latitude and longitude, walking distance to the nearest MRT/LRT, walking distance to the nearest bus stop.

4.3.2 Hyper-Parameter Tuning

Hyper-parameters are parameters that can be tuned but cannot be updated during training. If the hyper-parameters are not selected properly, under-fitting or over-fitting problems may occur. Hence, hyper-parameter tuning is crucial in model training.

In this dissertation, a grid search with cross-validation was adopted to find the optimal hyper-parameters. The workflow of parameters tuning and model evaluation is shown in Figure 4.3.1. In the beginning, the data set was partitioned into training and test sets. The training set was used in grid search with k-fold cross validation to select the best hyper-parameters of the model. Then the model with such optimal selection was retrained in the whole training set and evaluated in the test set.

Grid search is a common tuning method that performs an exhaustive search in the hyper-parameters list, which is generated by adjusting each hyper-parameter by a specified step size within a specified range. After that, train models with each parameters combination to find the optimal one.

K-fold cross validation is conducted to judge the performance of the hyper-

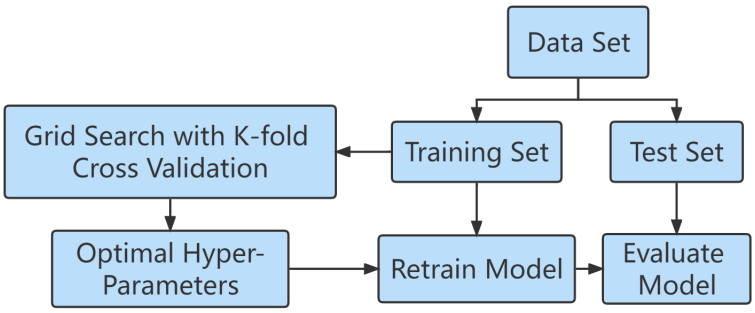


Figure 4.3.1: Work flow of parameter tuning and model evaluation.

parameters. The original training set is divided into K non-overlapping subsets. Then, model training and validation are performed K times. Each time train on $K-1$ subsets, and validate on the remaining subset that were not used for training in this time. Finally, training and validation errors are estimated by averaging the results of the K experiments.

4.3.3 Models

4.3.3.1 First Phase Model: to Predict the future $\mathrm{MRP}/\mathrm{m}^2$

The first phase model used to predict the future $\mathrm{MRP}/\mathrm{m}^2$ was organized by 1 or 2 LSTM layers plus a Fully-Connected (FC) layer with 1 neuron. Each time a mini-batch of subsequences with 24 time-steps was fed into the first LSTM layer, and then passed through the second LSTM layer or the FC layer to output the prediction results. *Adam* [19] was chosen to be the optimizer since it combines the pros of *AdaGrad* and *RMSProp* [20] - straightforward and efficient. The learning rate was set to be 0.0001. Apart from these, the hyper-parameters to be tuned were:

- The number of the LSTM layers: 1 or 2

- The neurons in each LSTM layer: 100, 200 or 300
- The batch size: 128 or 256

The performance metric used during training was Mean Square Error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.3.2}$$

where y_i and $\hat{y}_i$ are the true and predicted values for the i^{th} sample.

The tuning results can be seen in Table C-1 in Appendix C.

4.3.3.2 Second Phase Models: to Predict the Resale Prices

(i) LSTM

The LSTM model used to predict the resale prices of HDB flats was the same as the first phase LSTM model in structure, activation function, and learning rate. The essential difference was at the input: The inputs to the second phase LSTM model were the flats' intrinsic and extrinsic attributes, in addition to the MRP/m^2 that was optional for comparing the performances of the two-phase and single-phase methods. Besides, the tuning hyper-parameters were:

- The number of the LSTM layers: 1 or 2
- The neurons in each LSTM layer: 64, 128 or 256
- The batch size: 64, 128 or 256

The tuning results are displayed in Table C-2 in Appendix C.

(ii) MLP

The MLP model used to predict the resale prices of HDB flats consisted of 2 or 3 hidden layers and an output layer that was an FC network with 1 neuron.

To assess the utility of the feature - MRP/m^2, two kinds of MLP models needed to be built. For the first kind, only the intrinsic and external attributes of flats, as shown in Figure 4.1.1, were fed to the neural network as features. That is, the input dimension of the samples was 61. In comparison, the other model fed MRP/m^2 as the additional feature, which would change the input dimension to 62. The hyper-parameters of these two kinds of MLP models were both tuned to find the optimal combination.

As previously, Adam and MSE were chosen to be the optimizer and the performance metric in training, and the learning rate was set to be 0.001. ReLU is selected as the activation function. The hyper-parameters that needed to be tuned were:

- The number of the hidden layers: 2 or 3
- The neurons in each hidden layer: 64, 128 or 256
- The batch size: 128 or 256

The tuning results can be seen in Table C-3 in Appendix C.

(iii) CNN

The CNN model consisted of 2 or 3 convolutional layers, a max-pooling layer, and an FC layer with 1 neuron to output the resale prices of HDB flats.

Similarly, to evaluate the effectiveness of the MRP/m^2 as a feature, two kinds of CNN models were built: the first one only fed the intrinsic and external attributes to the CNN, while the other added MRP/m^2. Note that the CNN only

accepted a square tensor as input, so it was necessary to reshape the features to an 8 by 8 tensor by padding zeros at the end, changing the feature dimension to 64.

As always, the hyper-parameters of these two kinds of CNN models were tuned, searching for the best combination. Adam, MSE, and ReLU were used, and the learning rate was set to be 0.0001. The tuning hyper-parameters in grid search were:

- The number of the convolutional layers: 2 or 3
- The neurons in each convolutional layer: 64, 128 or 256
- The batch size: 128 or 256

The tuning results are shown in Table C-4 in Appendix C.

4.3.4 Prediction

The resale prices of flats are basically in accordance with the laws of the market that will not change much within a month. Thus, during one month, the price distribution must have some relationship with its median value. And it is reasonable to assume that MRP/m^2 can help the second phase models better predict the resale prices. However, when a flat is resold in the future, the MRP/m^2 in the future month is not known. That's why the first phase LSTM model is required to forecast the future MRP/m^2.

As shown in Figure 4.2.1, a sample contains information about the month when a flat was resold. According to such information, the MRP/m^2 of the previous 24 months could be found after data preparation, and they were used as the input of the first phase model to estimate the MRP/m^2 at the resale month.

The estimated result, as well as other features, would then feed into the second phase models to predict the future resale price of the flat.

As the first phase LSTM model is used to predict the future trend of housing prices, the accuracy is likely to decline over time. A simple solution is to update and retrain the model with new incoming data of $\mathrm{MRP}/\mathrm{m}^2$. On the contrary, the predicted accuracies of the second phase models are not so time-sensitive.

Chapter 5

Results and Analysis

This chapter encompasses the performance of models on the test set. According to the evaluation results, the analysis and comparison are made. In the end, the feasibility and superiority of the optimal model are proved.

5.1 Performance Metric

The resale prices of flats are like stock prices, in that the relative quantities, not absolute ones, are cared about. Therefore, the relative error should be paid more concerned. An efficient method is to measure the price discrepancies by their logarithm. The Log RMSE, which is calculated as the error rate of a model, can be expressed as Eqn. (5.1.1):

$$Log\ RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\ln y_i - \ln \hat{y}_i)^2} \tag{5.1.1}$$

where y_i is the true resale price, $\hat{y}_i$ is the predicted resale price, and $i = 1, 2, \cdots, n$ represents the sample number in the test dataset.

5.2 Results

5.2.1 Results for Prediction of future $\mathrm{MRP/m^2}$

Table 5.2.1 shows the performance of two first phase models on the test set.

Model	Description	Log RMSE
LSTM1	1 LSTM layer	0.014227
LSTM2	2 LSTM layers	**0.011541**

Table 5.2.1: The performance of the first phase models with optimal hyper-parameters

Based on the table above, it can be concluded that the first phase model with 2 LSTM layers outperforms the single LSTM layer model.

To more intuitively view the results of the first phase model with 2 LSTM

layers, the comparison of the true and predicted values of MRP/m^2 is shown in Figure 5.2.1. The solid line represents the true values while the dashed line represents the predicted results. The values of MRP/m^2 for the first 24 months are used to predict, so the predicted results begin from the 25^{th} month. Compared to the value of MRP/m^2 (4 to 5 thousand), the error is relatively very small, so it can be demonstrated that the model fits well.

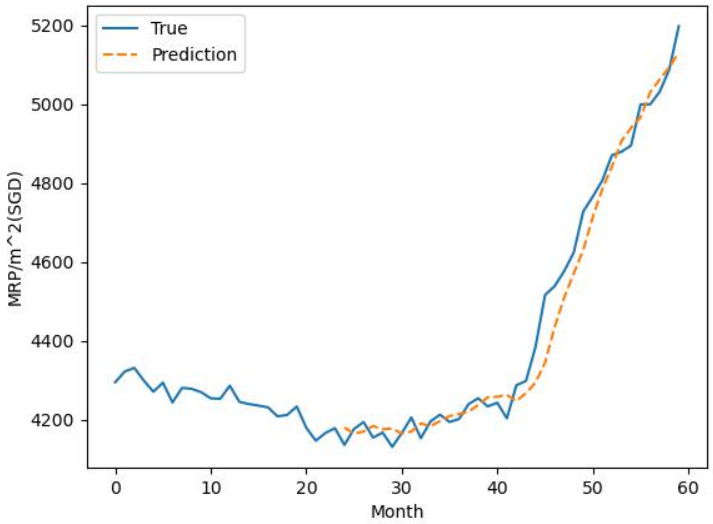


Figure 5.2.1: Comparison of the true and predicted values for the LSTM2 model.

5.2.2 Results for Prediction of Resale Prices

Table 5.2.2 shows the comparison of the performance of six second-phase models with and without the input feature MRP/m^2.

According to the table 5.2.2, it is clear that the second phase models with MRP/m^2 as the input features have a better performance, regardless of LSTM, MLP or CNN. These results prove the contribution of the first phase model to the prediction of the resale price. Also, the MLP models, no matter with or without

Log RMSE		Description	
		With MRP/m^2 as input feature	Without MRP/m^2 as input feature
Models	LSTM	0.075812	0.091896
	MLP	**0.066186**	0.068778
	CNN	0.076060	0.099221

Table 5.2.2: The performance of the second phase models with optimal hyper-parameters

the MRP/m^2 as input features, always outperform the LSTM and CNN models.

Parts of the predicted results for the three second phase models - LSTM, MLP OR CNN, when two-phase method is applied, are displayed in Figure X. The blue solid line represents the true value; the orange dashed line represents the predicted results of the LSTM; the green dotted line represents the predicted results of the MLP; and the red dashdotted line represents the predicted results of the CNN. All three models fit the test samples to some extent, but the MLP ranks first in accuracy, followed by the LSTM and the CNN at the bottom.

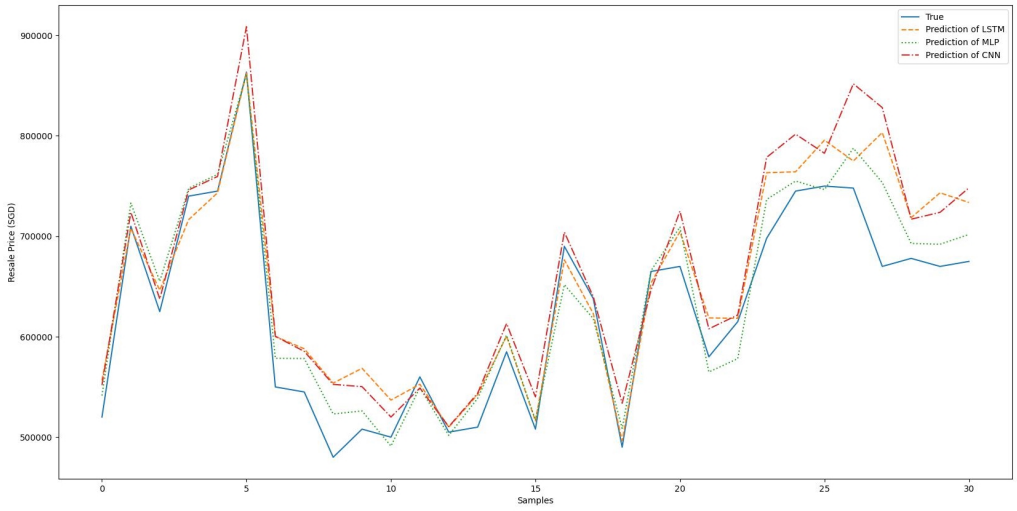


Figure 5.2.2: Comparison of the true and predicted values for the second phase models - LSTM, MLP and CNN.

CNN's worse performance is perhaps because it is more suitable for graph-

structured data analysis, yet the sample features used in this dissertation is a kind of tabular data without structural information.

LSTM can do well in predicting the future trend of housing price, but has unsatisfactory performance when predicting the resale price of flat units. This indicates that LSTM is is probably better to fit samples with time-dependent features, and poorer to fit those with time-independent features.

Figure 5.2.3 shows the true and the predicted results of the MLP models with and without the MRP/m^2 as inputs for 31 samples in the test set. The blue solid line represents the true resale prices; the orange dashed line represents the predicted results of the single-phase method using MLP model; and the green dotted line represents the predicted values of the two-phase method using MLP model. Obviously, the dotted line fits the solid line better, compared to the dashed line. So, the two-phase method with MLP model provides more accurate predictions in these samples than the single-phase method.

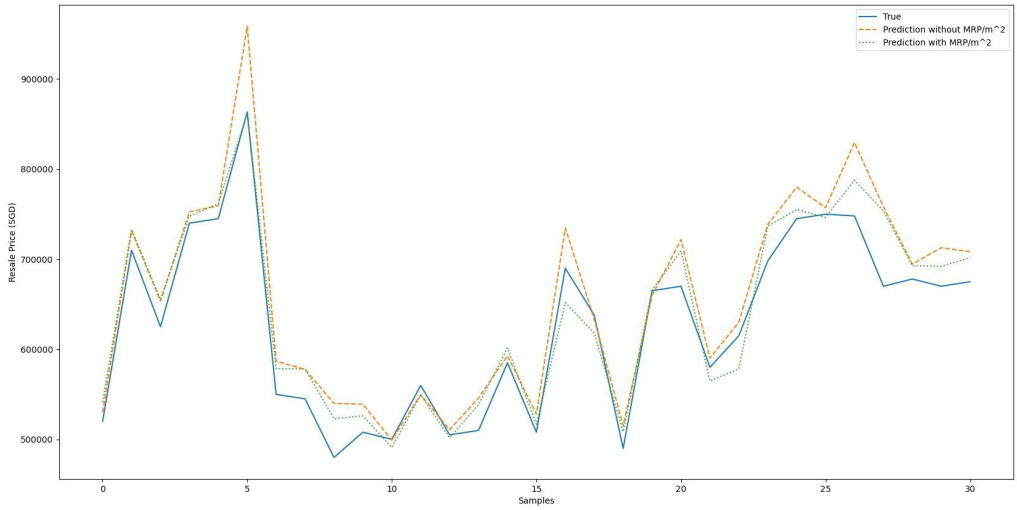


Figure 5.2.3: Comparison of the true and predicted values for the two MLP models: two-phase method v.s. single-phase method.

In conclusion, the optimal solution of resale prices prediction is the two-

phase method that first uses the LSTM model to predict the MRP/m^2, and then combine the MLP model to achieve the final results. In other words, among the six choices of single-phase methods: LSTM, MLP and CNN, and two-phase methods: LSTM+LSTM, LSTM+MLP, and LSTM+CNN, the collaboration of the LSTM and the MLP models outperforms others.

Chapter 6

Conclusions and Future Works

This chapter summarizes the whole work that has been done. The conclusion of the experiment results is drawn, along with some recommendations for future work to suggest possible ways to further optimize the model.

6.1 Conclusion

In this project, I first used the first phase LSTM model to predict the MRP/m^2 of HDB flats in Singapore for the next month based on data from the previous 24 months. The resale price of a certain flat unit was then predicted using the second phase models: LSTM, MLP, or CNN. In order to observe whether the two-phase methods are superior to that of the single-phase, six kinds of models were set up, three of which only inputted the flats' intrinsic attributes from the dataset, and the external attributes acquired through a web crawler, into the second phase models; The other three added MRP/m^2 for the resale month, which was predicted by the first phase model, as input feature.

The experiment results showed that the first phase LSTM model could fit well and accurately predict the MRP/m^2 of the flats in a certain month. Both the LSTM, MLP and CNN models could achieve a relatively small error in predicting the resale price of the flats, but LSTM and CNN performed slightly worse than MLP. Besides, adding the MRP/m^2 as an input to the second phase models moderately improved the prediction accuracy, which indicates that the two-phase methods appear to be better than the single-phase ones. Accordingly, combining the **LSTM** and **MLP** model achieved the best prediction results with error **0.066186** on the resale price of the HDB flats.

6.2 Recommendations for Further Research

Future work can import more information that may affect the resale price of a flat. In terms of the attributes of a flat unit, the number of schools, banks, or supermarkets nearby could become an influence on the buyers' intentions. In addition, the exterior of a flat, as well as how well the furniture is maintained, are

actually decisive factors in resale price, despite the difficulty to collect such data.

Moreover, the macroeconomic or policy-related factors, like Return on Investment (ROI) of rental, GDP growth, changes in the resident population, social employment rate, or the bank loan rate, should also be taken into consideration, since they will more or less have an effect upon the housing price.

References

[1] (2020, Jul) Public housing – a singapore icon. [Online]. Available: https://www.hdb.gov.sg/cs/infoweb/about-us

[2] (2021, Apr) Hdb towns, your home. [Online]. Available: https://www.hdb.gov.sg/cs/infoweb/about-us/history/hdb-towns-your-home

[3] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021.

[4] . S. J. Hochreiter, S., "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] K. E. Case and R. J. Shiller, "The efficiency of the market for single-family homes," *The American Economic Review*, vol. 79, no. 1, pp. 125–137, 1989.

[6] L. J. J. Gu, M. Zhu, "Housing price forecasting based on genetic algorithm and support vector machine," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3383–3386, 2011.

[7] Q. H. Z. Peng and Y. Han, "Model research on forecast of secondhand house price in chengdu based on xgboost algorithm," in *Proc. IEEE 11th Int. Conf. Adv. Infocomm Technol. (ICAIT)*, Oct 2019, pp. 168–172.

[8] L. Bork and S. V. Moller, "Forecasting house prices in the 50 states using dynamic model averaging and dynamic model selection," *Int. J. Forecasting*, vol. 31, no. 1, pp. 63–78, Jan 2015.

[9] T. D. Phan, "Housing price prediction using machine learning algorithms: The case of melbourne city, australia," in *Proc. Int. Conf. Mach. Learn. Data Eng. (iCMLDE)*, Dec 2018, pp. 35–42.

[10] M. A. A. S. Temür and G. Temür, "Predicting housing sales in turkey using arima, lstm and hybrid models," *J. Bus. Econ. Manage.*, vol. 20, no. 5, pp. 920–938, Jul 2019.

[11] H. X. Y. W. L. Yu, C. Jiao and K. Wang, "Prediction on housing price based on deep learning," *Int. J. Comput. Inf. Eng.*, vol. 12, no. 2, pp. 90–99, 2018.

[12] L. C. Q. You, R. Pang and J. Luo, "Image-based appraisal of real estate properties," *IEEE Trans. Multimedia*, vol. 19, no. 12, pp. 2751–2759, Dec 2017.

[13] T. M. O. Poursaeed and S. Belongie, "Vision-based real estate price estimation," *Mach. Vis. Appl.*, vol. 29, no. 4, pp. 667–676, May 2018.

[14] X. Chen, L. Wei, and J. Xu, "House price prediction using lstm," 2017. [Online]. Available: https://arxiv.org/abs/1709.08432

[15] L. P. Wen and T. X. Qin. (2021, Dec) Resale flat prices based on registration date. [Online]. Available: https://data.gov.sg/dataset/resale-flat-prices

[16] (2021, May) Python 3.8.10 documentation. [Online]. Available: https://docs.python.org/release/3.8.10/

[17] (2018) Onemap: The most detailed and comprehensive map of singapore. [Online]. Available: https://www.onemap.gov.sg/home/

[18] Mapbox docs: Examples, tutorials, and api references to help you start building with mapbox. [Online]. Available: https://docs.mapbox.com/

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[20] S. Ruder, "An overview of gradient descent optimization algorithms," 2016. [Online]. Available: https://arxiv.org/abs/1609.04747

Appendix A: Explanation about Flat Type and Flat Model

Table A-1: Types of Flats

HDB Flat Types	Flat Features
1-Room	- Kitchen, living room, and bedroom are combined into one larger space
2-Room	- 1 bedroom - 1 bathroom - Living/Dining - Kitchen - Household shelter
3-Room	- 2 bedrooms, 1 of which is a master bedroom with attached bathroom - Common bathroom - Living/dining - Kitchen/utility - Household shelter
4-Room	- 3 bedrooms, 1 of which is a master bedroom with attached bathroom - Common bathroom - Living/dining - Kitchen/utility - Service yard - Household shelter

Continued on next page

Table A-1 – continued from previous page

HDB Flat Types	Flat Features
5-Room	- 3 bedrooms, 1 of which is a master bedroom with attached bathroom - Common bathroom - Living/dining - Kitchen/utility - Service yard - Household shelter - Suggested study
Executive	- 1 master bedroom with attached bathroom - 2 additional bedrooms - Common bathroom - Living/dining - Kitchen
Multi-generation	- 4 bedrooms, 2 of which have attached bathrooms - Common bathroom - Living/dining - Kitchen - Service yard - Household shelter

Table A-2: Models of Flat

Instead of the floor plan layout, the naming of flat type and model represent approximate size in square meters and number of rooms. The naming standards of flats change over time

Years	**Model**	**Type**	**Usual area in square meters**	**Difference in layout**
1950s-1960s	Terrace	3-Room		Built in the 1950s by the Singapore Improvement Trust (SIT). When HDB came into existence, they took over the HDB terrace houses from SIT in the late 1960s and early 1970s.
		4-Room		
1960s-1970s	Standard	2-Room	41	One WC/Shower
		3-Room	50-55	
		4-Room	70-75	
		5-Room	117-123	Two toilets
	Improved	1-Room	33	One WC/Shower
		2-Room	44	Separate WC and shower
		3-Room	60	
		4-Room	82-84	
1970s-1980s	New Generation	3-Room	67	Two toilets
		4-Room	92	
	Improved	5-Room	121-123	
	Simplified	3-Room	64	Two full-size toilets
		4-Room	84	

Continued on next page

Table A2 - continued from previous page

Years	Model	Type	Usual area in square meters	Difference in layout
1970s-1980s	Model A	3-Room	73-75	Two full-size toilets
		4-Room	105	
		5-Room	135	Two full-size toilets. Replaced by Executive Apartment after 1984.
	Model A-Maisonette	5-Room	Most 140, some up to 155	Two storeys. Replaced by Executive Maisonette after 1984.
	Apartment	Executive	142-146	One storey.
	Maisonette	Executive	147, some up to 160	Two storeys.
	Multi Generation	Multi - Generation	132 or 151-171, most 166	One storey, two entrances
	Adjoined flat	4-Room	90-175	Adjoined by owners. 2 entrances, 2 kitchens, up to 4 toilets.
		5-Room		
		Executive		
1990s-present	Model A	2-Room	45	
		3-Room	60 (2002-2008) or 65 (2008-present)	
		4-Room	100 (1998-2000) or 90 (2000-present)	
		5-Room	133-137	
	Model A2	4-Room	90 (1998-2000), 85, some 80 (2000-present)	Only in SERS blocks
	Apartment	Executive	140 (1998-2000), 125 or 130 (2000-2005)	

Continued on next page

Table A2 - continued from previous page

Years	Model	Type	Usual area in square meters	Difference in layout
1990s-present	Maisonette	Executive	140	
	Multi Generation	Multi - Generation	115	
	Premium Apartments	2-Room		Featuring better quality finishes, you get them in ready-to-move condition, with flooring, kitchen cabinets, built-in wardrobes.
		3-Room		
		4-Room		
		5-Room		
		Executive		
	Premium Maisonette	Executive		
	Premium Apartment Loft	4-Room	95-97	Two storeys and a high ceiling.
		5-Room	147-149	
	Design, Build and Sell Scheme (DBSS)	3-Room		DBSS is public housing built by private developers and sold by agents, their architectural design is similar with condos, but lack condo facilities such as guards, gym, pool, tennis courts, etc.
		4-Room		
		5-Room		
	Type S1	4-Room	90-93	For HDB anniversary of 50 years, the first 50-storey public housing complex was launched, including 2 flat types: Type S1 and Type S2.
	Type S2	5-Room	103-106	

Appendix B: Supplement to Feature Encoding

Table B-1: Feature Encoding for Town

Town	Numeric Index
ANG MO KIO	0
BEDOK	1
BISHAN	2
BUKIT BATOK	3
BUKIT MERAH	4
BUKIT PANJANG	5
BUKIT TIMAH	6
CENTRAL AREA	7
CHOA CHU KANG	8
CLEMENTI	9
GEYLANG	10
HOUGANG	11
JURONG EAST	12
JURONG WEST	13
KALLANG/WHAMPOA	14
MARINE PARADE	15
PASIR RIS	16
PUNGGOL	17
QUEENSTOWN	18
SEMBAWANG	19
SENGKANG	20
SERANGOON	21
TAMPINES	22

Continued on next page

Table B1 - continued from previous page

Town	Numeric Index
TOA PAYOH	23
WOODLANDS	24
YISHUN	25

Table B-2: Feature Encoding for Flat Type

Flat Type	Numeric Index
1 ROOM	0
2 ROOM	1
3 ROOM	2
4 ROOM	3
5 ROOM	4
EXECUTIVE	5
MULTI-GENERATION	6

Table B-3: Feature Encoding for Flat Model

Flat Model	Numeric Index
Terrace	0
2-room	1
Standard	2
Improved	3
New Generation	4
Simplified	5
Model A	6
Model A-Maisonette	7
Apartment	8
Maisonette	9
Improved-Maisonette	10
Multi Generation	11
Adjoined flat	12
Model A2	13
Premium Apartment	14
Premium Maisonette	15
Premium Apartment Loft	16
DBSS	17
Type S1	18
Type S2	19

Appendix C: Hyper-parameters Tuning Results

Table C-1: Hyper-parametes tuning results for the first phase LSTM model

Batch Size	Number of LSTM Layers	Number of Neurons	Avg TrainMSE	Avg ValMSE	Standard Deviation
128	1	[100]	0.000234	0.000483	0.000585
128	1	[200]	0.000195	0.000360	0.000364
128	1	[300]	0.000165	0.000359	0.000378
128	2	[100, 100]	0.000184	0.000420	0.000518
128	2	[100, 200]	0.000188	0.000373	0.000417
128	2	[100, 300]	0.000175	0.000417	0.000502
128	2	[200, 100]	0.000164	0.000378	0.000413
128	2	[200, 200]	0.000150	0.000324	0.000344
128	2	[200, 300]	0.000149	0.000340	0.000392
128	2	[300, 100]	0.000138	0.000359	0.000412
128	2	[300, 200]	0.000154	0.000369	0.000429
128	**2**	**[300, 300]**	**0.000127**	**0.000275**	**0.000310**
256	1	[100]	0.000549	0.000791	0.000759
256	1	[200]	0.000361	0.000443	0.000348
256	1	[300]	0.000320	0.000453	0.000402
256	2	[100, 100]	0.000494	0.000627	0.000625
256	2	[100, 200]	0.000322	0.000514	0.000534
256	2	[100, 300]	0.000253	0.000445	0.000431
256	2	[200, 100]	0.000294	0.000463	0.000444
256	2	[200, 200]	0.000243	0.000405	0.000397
256	2	[200, 300]	0.000261	0.000396	0.000366
256	2	[300, 100]	0.000247	0.000414	0.000388
256	2	[300, 200]	0.000208	0.000379	0.000369
256	2	[300, 300]	0.000216	0.000376	0.000364

Table C-2: Hyper-parameters tuning results for the second phase LSTM models

Batch Size	No. of LSTM Layers	No. of LSTM Neurons	Avg ValMSE	
			LSTM1(with MRP/m^2)	LSTM2(without MRP/m^2)
64	1	[64]	0.0009786	0.0010126
64	1	[128]	0.0009676	0.0010007
64	1	[256]	0.0009479	0.0009843
64	2	[64, 64]	0.0009348	0.0009588
64	2	[64, 128]	0.0009221	0.0009466
64	2	[64, 256]	0.0009218	0.0009508
64	2	[128, 64]	0.0008977	0.0009378
64	2	[128, 128]	0.0008919	0.0009339
64	2	[128, 256]	0.0008854	0.0009201
64	2	[256, 64]	0.0008355	0.0008878
64	**2**	**[256, 128]**	0.0008309	**0.0008667**
64	**2**	**[256, 256]**	**0.0008267**	0.0008773
128	1	[64]	0.0010386	0.0010852
128	1	[128]	0.0010265	0.0010658
128	1	[256]	0.0010107	0.0010552
128	2	[64, 64]	0.0009981	0.0010373

Continued on next page

Table C-2 – continued from previous page

Batch Size	No. of LSTM Layers	No. of LSTM Neurons	Avg ValMSE	
			LSTM1(with MRP/m^2)	LSTM2(without MRP/m^2)
128	2	[64, 128]	0.0009919	0.0010283
128	2	[64, 256]	0.0009837	0.0010270
128	2	[128, 64]	0.0009670	0.0010151
128	2	[128, 128]	0.0009572	0.0009974
128	2	[128, 256]	0.0009554	0.0009931
128	2	[256, 64]	0.0009421	0.0009837
128	2	[256, 128]	0.0009343	0.0009775
128	2	[256, 256]	0.0009311	0.0009708
256	1	[64]	0.0010920	0.0011330
256	1	[128]	0.0010853	0.0011288
256	1	[256]	0.0010774	0.0011206
256	2	[64, 64]	0.0010643	0.0011073
256	2	[64, 128]	0.0010552	0.0010987
256	2	[64, 256]	0.0010524	0.0010927
256	2	[128, 64]	0.0010492	0.0010881
256	2	[128, 128]	0.0010433	0.0010799
256	2	[128, 256]	0.0010310	0.0010808

Continued on next page

Table C-2 – continued from previous page

Batch Size	No. of LSTM Layers	No. of LSTM Neurons	Avg ValMSE	
			LSTM1(with MRP/m^2)	LSTM2(without MRP/m^2)
256	2	[256, 64]	0.0010313	0.0010702
256	2	[256, 128]	0.0010172	0.0010613
256	2	[256, 256]	0.0010147	0.0010516

Table C-3: Hyper-parameters tuning results for the second phase MLP models

Batch Size	No. of Hidden Layers	No. of Hidden Neurons	Avg ValMSE	
			MLP1(with MRP/m^2)	MLP2(without MRP/m^2)
128	2	[64, 64]	0.0004567	0.0005448
128	2	[64, 128]	0.0004313	0.0005238
128	2	[64, 256]	0.0004212	0.0005171
128	2	[128, 64]	0.0004112	0.0005177
128	2	[128, 128]	0.0003963	0.0005048
128	2	[128, 256]	0.0003762	0.0005032
128	2	[256, 64]	0.0003982	0.0005050
128	2	[256, 128]	0.0003837	0.0004923
128	2	[256, 256]	0.0003623	0.0005003
128	3	[64, 64, 64]	0.0004087	0.0005107
128	3	[64, 64, 128]	0.0004139	0.0005184
128	3	[64, 64, 256]	0.0004028	0.0005170
128	3	[64, 128, 64]	0.0003902	0.0005070
128	3	[64, 128, 128]	0.0003768	0.0005040
128	3	[64, 128, 256]	0.0003735	0.0005026
128	3	[64, 256, 64]	0.0003747	0.0005020

Continued on next page

Table C-3 – continued from previous page

Batch Size	No. of Hidden Layers	No. of Hidden Neurons	Avg ValMSE	
			MLP1(with MRP/m^2)	MLP2(without MRP/m^2)
128	3	[64, 256, 128]	0.0003619	0.0005039
128	3	[64, 256, 256]	0.0003520	0.0005032
128	3	[128, 64, 64]	0.0004023	0.0004998
128	3	[128, 64, 128]	0.0003908	0.0005044
128	3	[128, 64, 256]	0.0003924	0.0004959
128	3	[128, 128, 64]	0.0003641	0.0005010
128	3	[128, 128, 128]	0.0003642	0.0004932
128	3	[128, 128, 256]	0.0003556	0.0004989
128	3	[128, 256, 64]	0.0003465	0.0004964
128	3	[128, 256, 128]	0.0003437	0.0004972
128	3	[128, 256, 256]	0.0003325	0.0005038
128	3	[256, 64, 64]	0.0003873	0.0005002
128	3	[256, 64, 128]	0.0003740	0.0005025
128	3	[256, 64, 256]	0.0003786	0.0005063
128	3	[256, 128, 64]	0.0003615	0.0005008
128	3	[256, 128, 128]	0.0003441	0.0004990
128	3	[256, 128, 256]	0.0003619	0.0004980

Continued on next page

Table C-3 – continued from previous page

Batch Size	No. of Hidden Layers	No. of Hidden Neurons	Avg ValMSE	
			MLP1(with MRP/m^2)	MLP2(without MRP/m^2)
128	3	[256, 256, 64]	0.0003413	0.0004966
128	3	[256, 256, 128]	0.0003328	0.0005032
128	3	[256, 256, 256]	0.0003320	0.0005037
256	2	[64, 64]	0.0004780	0.0005620
256	2	[64, 128]	0.0004507	0.0005404
256	2	[64, 256]	0.0004344	0.0005238
256	2	[128, 64]	0.0004308	0.0005285
256	2	[128, 128]	0.0004177	0.0005112
256	2	[128, 256]	0.0003873	0.0004934
256	2	[256, 64]	0.0004011	0.0005119
256	2	[256, 128]	0.0003848	0.0004987
256	2	[256, 256]	0.0003586	0.0004934
256	3	[64, 64, 64]	0.0004213	0.0005259
256	3	[64, 64, 128]	0.0004344	0.0005236
256	3	[64, 64, 256]	0.0004174	0.0005176
256	3	[64, 128, 64]	0.0004000	0.0005161
256	3	[64, 128, 128]	0.0003910	0.0004994

Continued on next page

Table C-3 – continued from previous page

Batch Size	No. of Hidden Layers	No. of Hidden Neurons	Avg ValMSE	
			MLP1(with MRP/m^2)	MLP2(without MRP/m^2)
256	3	[64, 128, 256]	0.0003724	0.0005084
256	3	[64, 256, 64]	0.0003832	0.0005021
256	3	[64, 256, 128]	0.0003718	0.0005019
256	3	[64, 256, 256]	0.0003516	0.0004965
256	3	[128, 64, 64]	0.0004093	0.0005145
256	3	[128, 64, 128]	0.0003979	0.0004992
256	3	[128, 64, 256]	0.0003841	0.0004960
256	3	[128, 128, 64]	0.0003862	0.0004935
256	3	[128, 128, 128]	0.0003722	0.0005010
256	3	[128, 128, 256]	0.0003623	0.0004932
256	3	[128, 256, 64]	0.0003502	0.0004856
256	3	[128, 256, 128]	0.0003391	0.0004977
256	3	[128, 256, 256]	0.0003215	0.0004923
256	3	[256, 64, 64]	0.0003754	0.0005013
256	3	[256, 64, 128]	0.0003818	0.0004965
256	3	[256, 64, 256]	0.0003655	0.0004915
256	3	[256, 128, 64]	0.0003547	0.0004912

Continued on next page

Table C-3 – continued from previous page

Batch Size	No. of Hidden Layers	No. of Hidden Neurons	Avg ValMSE	
			MLP1(with MRP/m^2)	MLP2(without MRP/m^2)
256	**3**	**[256, 128, 128]**	0.0003505	**0.0004854**
256	3	[256, 128, 256]	0.0003482	0.0004917
256	3	[256, 256, 64]	0.0003327	0.0004890
256	3	[256, 256, 128]	0.0003232	0.0004927
256	**3**	**[256, 256, 256]**	**0.0003189**	0.0004915

Table C-4: Hyper-parameters tuning results for the second phase CNN models

Batch Size	No. of Conv. Layers	No. of Conv. Neurons	Avg ValMSE	
			CNN1(with MRP/m^2)	CNN2(without MRP/m^2)
128	2	[64, 64]	0.0009031	0.0009495
128	2	[64, 128]	0.0008561	0.0009048
128	2	[64, 256]	0.0008160	0.0008673
128	2	[128, 64]	0.0008780	0.0009043
128	2	[128, 128]	0.0008171	0.0008760
128	2	[128, 256]	0.0007850	0.0008360
128	2	[256, 64]	0.0008368	0.0008818
128	2	[256, 128]	0.0008091	0.0008346
128	2	[256, 256]	0.0007645	0.0008066
128	3	[64, 64, 64]	0.0007998	0.0008432
128	3	[64, 64, 128]	0.0007659	0.0008247
128	3	[64, 64, 256]	0.0007343	0.0007828
128	3	[64, 128, 64]	0.0007603	0.0008044
128	3	[64, 128, 128]	0.0007303	0.0007812
128	3	[64, 128, 256]	0.0007024	0.0007467
128	3	[64, 256, 64]	0.0007194	0.0007824

Continued on next page

Table C-4 – continued from previous page

Batch Size	No. of Conv. Layers	No. of Conv. Neurons	Avg ValMSE	
			CNN1(with MRP/m^2)	CNN2(without MRP/m^2)
128	3	[64, 256, 128]	0.0006926	0.0007473
128	3	[64, 256, 256]	0.0006732	0.0007212
128	3	[128, 64, 64]	0.0007767	0.0008326
128	3	[128, 64, 128]	0.0007530	0.0008002
128	3	[128, 64, 256]	0.0007254	0.0007700
128	3	[128, 128, 64]	0.0007389	0.0007916
128	3	[128, 128, 128]	0.0007129	0.0007647
128	3	[128, 128, 256]	0.0006963	0.0007385
128	3	[128, 256, 64]	0.0007141	0.0007615
128	3	[128, 256, 128]	0.0006727	0.0007304
128	3	[128, 256, 256]	0.0006562	0.0007065
128	3	[256, 64, 64]	0.0007637	0.0008060
128	3	[256, 64, 128]	0.0007325	0.0007875
128	3	[256, 64, 256]	0.0007157	0.0007564
128	3	[256, 128, 64]	0.0007267	0.0007714
128	3	[256, 128, 128]	0.0006965	0.0007465
128	3	[256, 128, 256]	0.0006724	0.0007215

Continued on next page

Table C-4 – continued from previous page

Batch Size	No. of Conv. Layers	No. of Conv. Neurons	Avg ValMSE	
			CNN1(with MRP/m^2)	CNN2(without MRP/m^2)
128	3	[256, 256, 64]	0.0006883	0.0007365
128	3	[256, 256, 128]	0.0006603	0.0007130
128	**3**	**[256, 256, 256]**	**0.0006270**	**0.0006872**
256	2	[64, 64]	0.0009643	0.0009971
256	2	[64, 128]	0.0009167	0.0009519
256	2	[64, 256]	0.0008592	0.0009131
256	2	[128, 64]	0.0009243	0.0009587
256	2	[128, 128]	0.0008804	0.0009351
256	2	[128, 256]	0.0008352	0.0008851
256	2	[256, 64]	0.0008883	0.0009343
256	2	[256, 128]	0.0008541	0.0009046
256	2	[256, 256]	0.0008133	0.0008591
256	3	[64, 64, 64]	0.0008479	0.0008933
256	3	[64, 64, 128]	0.0008155	0.0008725
256	3	[64, 64, 256]	0.0007827	0.0008419
256	3	[64, 128, 64]	0.0008104	0.0008570
256	3	[64, 128, 128]	0.0007807	0.0008323

Continued on next page

Table C-4 – continued from previous page

Batch Size	No. of Conv. Layers	No. of Conv. Neurons	Avg ValMSE	
			CNN1(with MRP/m^2)	CNN2(without MRP/m^2)
256	3	[64, 128, 256]	0.0007546	0.0008054
256	3	[64, 256, 64]	0.0007739	0.0008284
256	3	[64, 256, 128]	0.0007599	0.0008067
256	3	[64, 256, 256]	0.0007282	0.0007786
256	3	[128, 64, 64]	0.0008326	0.0008868
256	3	[128, 64, 128]	0.0008054	0.0008579
256	3	[128, 64, 256]	0.0007869	0.0008301
256	3	[128, 128, 64]	0.0007936	0.0008454
256	3	[128, 128, 128]	0.0007670	0.0008183
256	3	[128, 128, 256]	0.0007512	0.0007983
256	3	[128, 256, 64]	0.0007645	0.0008194
256	3	[128, 256, 128]	0.0007427	0.0007957
256	3	[128, 256, 256]	0.0007218	0.0007683
256	3	[256, 64, 64]	0.0008032	0.0008644
256	3	[256, 64, 128]	0.0007980	0.0008308
256	3	[256, 64, 256]	0.0007685	0.0008163
256	3	[256, 128, 64]	0.0007773	0.0008279

Continued on next page

Table C-4 – continued from previous page

Batch Size	No. of Conv. Layers	No. of Conv. Neurons	Avg ValMSE	
			CNN1(with MRP/m^2)	CNN2(without MRP/m^2)
256	3	[256, 128, 128]	0.0007537	0.0008012
256	3	[256, 128, 256]	0.0007247	0.0007801
256	3	[256, 256, 64]	0.0007452	0.0008010
256	3	[256, 256, 128]	0.0007186	0.0007718
256	3	[256, 256, 256]	0.0006978	0.0007469