Predicting Housing Market Trend by NLP models

by

Chun-Wah Luk, David
Student ID: 2222991

Supervised by

Dr Rob Procter
Professor, Department of Computer Science

for the degree of

MSc Data Analytics

Department of Computer Science
University of Warwick

September 2024

Abstract

This dissertation explores the application of Natural Language Processing (NLP) techniques in predicting housing market trends for Hong Kong and the United Kingdom. This study develops and evaluates models including SARIMAX, LSTM, BERT and Lag-Llama, to capture market sentiments and forecast housing price movements by integrating quantitative economic indicators with qualitative textual data. It demonstrates the potential of NLP in enhancing traditional forecasting methods, and the results indicate that the effectiveness of NLP models varies with market conditions, showing the difference in the more dynamic Hong Kong housing market compared to the stable United Kingdom housing market. The study also highlights the challenges of balancing model complexity with adaptability across different market environments. This work tries to contribute to the field by providing a comprehensive framework for applying NLP to real estate market analysis and offering insights into the strengths and limitations of different modeling approaches.

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Rob Procter, for his support and guidance throughout this project. I also wish to thank JLL and Knight Frank for granting me permission to use their reports as the textual data source for the model training.

Declarations

I hereby declare that all work contained in this project is my own. All sentences or passages quoted in this report from other people's work have been specially acknowledged by clear cross-referencing to author, work and page(s). Some parts of the report are from my research proposal, presentation and interim report.

Contents

1 Introduction **6**
1.1 Hong Kong Housing Market 6
1.2 The United Kingdom Housing Market 7
1.3 Natural Language Processing for Analysing 8
1.4 Objectives 8

2 Literature Review **10**
2.1 Time Series Analysis and Financial Forecasting Methodologies 10
2.2 NLP Model Selection 12
2.2.1 LSTM Model 12
2.2.2 BERT Model 13
2.2.3 Hybrid Models 13
2.2.4 Evaluation Metrics for Model Selection 13
2.2.5 Large Language Models 14

3 Methodology **15**
3.1 Data and Preparation 15
3.2 Data Sources and Collection 15
3.2.1 Quantitative Data Sources 15
3.2.2 Qualitative Data Sources 17
3.3 Data Collection Process 18
3.3.1 Quantitative Data Processing 18
3.3.2 Qualitative Data Processing 19
3.4 Ensuring Data Quality 21

4 Model Specification **23**
4.1 Quantitative Model Implementation 23
4.1.1 Baseline Quantitative model - SARIMAX 23
4.1.2 Parameter Optimisation 23
4.1.3 Performance Evaluation 24
4.1.4 Model Interpretation 25
4.2 LSTM Model Implementation 25
4.2.1 Model Architecture 25
4.3 FinBERT Model Implementation 28
4.3.1 Transfer Learning and Model Architecture 28
4.4 Lag-Llama Model Implementation 31

4.4.1 Dataset Creation and Model Configuration 31

5 Discussion of Results **34**
5.1 Quantitative Model 34
5.2 LSTM Model 35
5.3 FinBERT Model 36
5.4 Lag-Llama Model 39
5.5 Result Evaluation 41
5.5.1 Hybrid Model Approaches 41
5.5.2 Model Robustness and Market Conditions 41
5.5.3 Probabilistic Forecasting and Uncertainty 42
5.5.4 Model Interpretability 42

6 Conclusion **43**
6.1 Achievements 43
6.2 Limitations 44
6.3 Future Work 45

7 Project Management and Tools **47**
7.1 Phase 1: Foundational Research and Data Preparation (February - April 2024) 47
7.2 Phase 2: Model Building and Evaluation (April - July 2024) . 48
7.3 Phase 3: Report Development and Review (July - August 2024) 48
7.4 Tools and Technologies 48

A Grid search result of HK SARIMAX Quantitative model **55**

B Grid search result of UK SARIMAX Quantitative model **57**

C Grid search result of HK LSTM SARIMAX hybrid model **59**

D Grid search result of UK LSTM SARIMAX hybrid model **61**

E Grid search result of HK BERT SARIMAX hybrid model **63**

F Grid search result of UK BERT SARIMAX hybrid model **65**

G Data Access Authorisation **67**

1 Introduction

The real estate market is a cornerstone of the economy. When it comes to buying and selling properties, these kind of activities play a key role in both building individual wealth and keeping the national economy healthy. That's the reason why it's important not just for investors, but also for policymakers and everyday people.

On the other hand, like other markets, the housing market goes through its own cycles. The changes from growing, stable and dips are driven by a mix of factors, like interest rates, situation of the economy, population growth, government policies, and even global events.

1.1 Hong Kong Housing Market

Hong Kong has long been known as the world's most expensive housing market [1]. Although property prices decreased during the 1997 Asian Financial Crisis, the market bounced back quickly in the early 2000s and continued to climb steadily. This growth was accelerated by a few key factors such as limited land, high demand, and low interest rates.

Despite the attempts from government to cool the market, such as introducing measures to curb prices and increasing the housing supply [2], property prices stayed high and kept increasing. However in the past few years, a mix of events including the COVID-19 pandemic and the pass of national security law that came into effect in 2020 turned things around: While housing in Hong Kong is still far from affordable, we can see a prices dropping about 20-30% from their peak over the last two years, indicating not just a small volatility, but a sign of long term market correction.

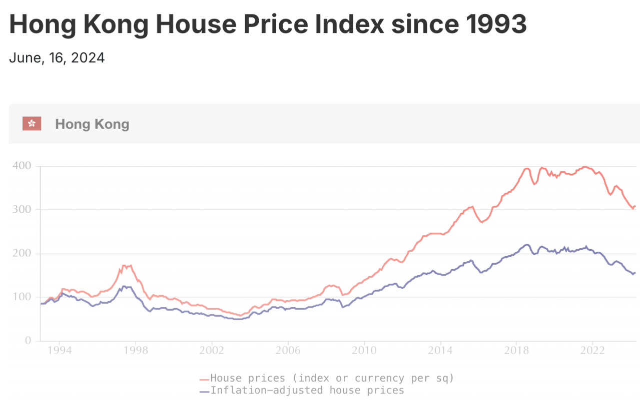


Figure 1: Hong Kong House Price Index since 1993 [3]

1.2 The United Kingdom Housing Market

As one of the world's largest economies, the United Kingdom's housing market is continuing to climb over time. Even though there were downturns during the 2008 global financial crisis and the COVID-19 pandemic in 2019, the market bounced back quickly and surged from 2000 to 2024 overall. Same as the Hong Kong housing market, this growth was driven by a few key factors, including economic expansion and low interest rates. The trend is especially noticeable in London, where demand has consistently outstripped supply, pushing prices even higher.

Figure 2: United Kingdom House Price Growth since 1994 [4]

1.3 Natural Language Processing for Analysing

The theory of "informational efficiency" suggests that in a perfectly efficient market, all transactions would reflect fair value, leaving no room for anyone to gain excess returns [5]. But in reality, especially in financial markets like the housing market, there's often an imbalance in the information that participants have access to. This gap creates an opportunity to use NLP techniques to analyse the market through different textual data, such as reports and news available today, and analyse the market sentiments and hence to predict the future trends. With the explosion of information in digital age, the qualitative data that influences housing markets has become more transparent and easier to access. The development of transformer-based LLMs and the impressive capabilities of NLP have opened up new research possibilities in this area.

The housing market is special because it has fewer transactions and lower liquidity compared to other financial markets. As a result, the information in reports and news articles becomes even more critical for both buyers and sellers. In this context, using NLP-based sentiment analysis can provide valuable insights by analysing the unstructured textual data. This makes NLP techniques especially appealing for improving decision-making processes in the real estate sector.

1.4 Objectives

The primary goal of this dissertation is to assess the effectiveness of NLP models in predicting housing market trends. This involves three key stages:

1. Collecting different property-related textual data.

2. Quantifying and extracting features from the collected data for sentiment analysis.

3. Integrating this data into different models for time series analysis and forecasting.

This approach aims to evaluate the capabilities of NLP models in the context of housing market forecasting.

The second objective is to compare the performance of different NLP models, especially the latest machine learning models such as LLMs, against traditional NLP techniques.

Last but not least, this project focuses on the housing markets of Hong Kong and the United Kingdom. This cross-market analysis helps to examine the adaptability and robustness of these NLP models under different market conditions.

In line with these objectives, the following research questions are posed and to be answered for this dissertation:

1. How does the accuracy of NLP models' forecasting compare with traditional forecasting methods when measured against historical data?

2. What are the performances of different NLP models, such as LSTM, BERT, or LLMs, in the context of housing market forecasting?

3. Does the efficacy of NLP models in forecasting housing market trends vary across markets with differing conditions, such as the declining trend in Hong Kong's market and the stability of the UK's market?

4. Has the optimal performance of these models been achieved?

2 Literature Review

To prepare this dissertation, we have conducted a thorough literature review that focused on three main areas. First of all, we studied the background of NLP in financial forecasting, as well as the methodologies used in time series analysis. This gave us a solid foundation and taught the core concepts of our dissertation.

Next, we have looked into different quantitative and qualitative models, considering both the numerical and textual data we had collected for this project. This part of the review helped us to weigh the pros and cons of different approaches when it comes to analysing the housing market trends.

Finally, we have explored how Large Language Models (LLMs) are being applied to time series analysis. This area of research was crucial in guiding our choice of LLM for the study.

This approach of literature review can help us to cover things from well-established methods to the latest techniques of NLP and time series forecasting, which can be used in housing market analysis and prediction.

2.1 Time Series Analysis and Financial Forecasting Methodologies

To shape our approach to housing market forecasting, we have reviewed a range of literature on using NLP and sentiment analysis for financial and stock market prediction. These studies offered valuable insights into methodologies that could be adapted for analysing the housing market.

'Natural language based financial forecasting: a survey' [6] provides a thorough overview of Natural Language Based Financial Forecasting (NLFF). This paper underscores the growing importance of NLP in financial forecasting, because of the increased availability of data and advancements in NLP techniques, both are the key to our study. Although their focus was on stock market and Foreign Exchange Rate (FOREX) prediction through market sentiment, the insights they offered significantly influenced the direction of our research.

Their study also examines how machine learning techniques like Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks can be combined with traditional time series models, such as Autoregressive Integrated Moving Average (ARIMA). This approach of combination was used in our dissertation, which as a result of this paper guiding our choice of the sentiment analysis model.

‘Text mining for market prediction: A systematic review’ explores various text mining approaches for market prediction [7], offering key reflections on forecasting methodologies that could be applied to the housing market. Their review of system architecture, data sources, pre-processing techniques, feature selection methods, and machine learning algorithms provided a fundamental understanding of the factors influencing market prediction in NLFF.

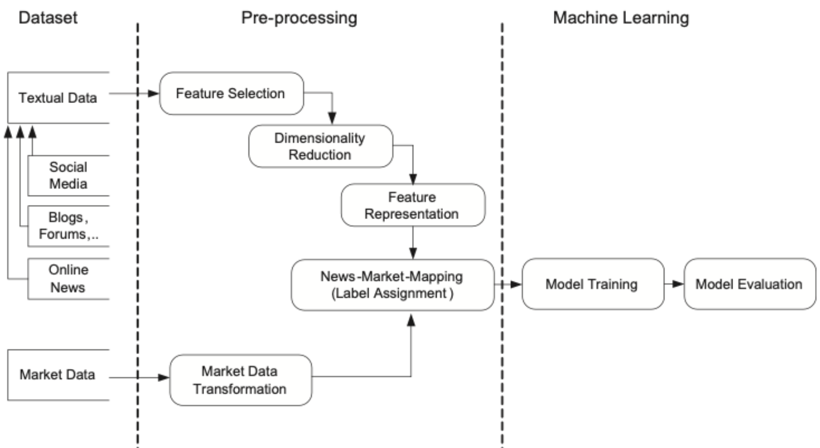


Figure 3: Generic system diagram of NLFF [7]

Khalil and Pipa’s ‘Is Deep-Learning and Natural Language Processing Transcending the Financial Forecasting? Investigation Through Lens of News Analytic Process’ [8] also studies the use of deep learning and sentiment analysis in financial forecasting. While their focus was on stock market prediction, their innovative approach, which incorporates news analytics processed through LSTM models, offered useful insights for our dissertation. It highlighted the effectiveness of combining news sentiment with LSTM for predicting price movements.

Although these studies are primarily focused on financial and stock markets, many of their findings can be applied to housing market analysis. Sentiment analysis plays an important role in housing markets, which the sentiment expressed in reports heavily influences the market trends, more than the other markets. We adapted the methodologies for processing textual data and extracting sentiment discussed in these papers to suit our study.

Furthermore, the NLP techniques and machine learning models that were highlighted in these studies can be applied to housing market textual data, including real estate reports and social media discussions. This allows us to extract relevant information and predict trends, leveraging the power of these advanced techniques specifically for housing market forecasting.

2.2 NLP Model Selection

The next phase of our study involved examining different NLP models to select suitable one for our research. Our approach focused on identifying models capable of performing sentiment analysis on textual datasets and how to incorporating them into predictive models.

Although there is a lack of literature directly discussing NLP models for housing market prediction, those insights from paper of related domains provided valuable guidance, particularly of the factors to consider in the selection of model.

Key considerations included:

1. **Data Type**: Our study requires a model capable of effectively handling both numerical time series data and textual data, as we are working on housing market forecasting using NLP.

2. **Data Dependencies**: The housing market exhibits trends that necessitate a model capable of capturing data dependencies, potentially including seasonal patterns. This factor is important in our consideration of model selection.

3. **Model Complexity**: While more complex models may perform better in capturing hidden sentiment in textual data, we needed to balance this with the complexity of our dataset. We hoped to find an appropriate equilibrium, recognising that overly complex models might not offer higher accuracy or could even underperform with relatively simple datasets.

Based on these factors and our literature review, we considered the following models and approaches:

2.2.1 LSTM Model

As a type of recurrent neural network, Long Short-Term Memory (LSTM) model is well-suited for time series data due to its ability to learn long-term dependencies. 'Forecasting Economics and Financial Time Series: ARIMA vs. LSTM' [9] demonstrates LSTM's capacity to outperform traditional

ARIMA models in certain economic and financial time series predictions. Additionally, Xu and Murata's 'Stock Market Trend Prediction with Sentiment Analysis based on LSTM Neural Network' [10] illustrates LSTM's potential to capture trends and seasonality by learning from historical data, which is relevant to housing market analysis.

2.2.2 BERT Model

FinBERT, a model developed from the architecture of BERT, is specifically designed for financial sentiment analysis by leveraging the strengths of pre-trained language models and tailoring them to the financial domain [11].

Financial language is highly specialised, often using terminology and expressions that are not commonly found in general language corpora. As a result, general-purpose sentiment analysis models may have struggle on capturing sentiment within financial contexts.

FinBERT addresses this challenge by undergoing further pre-training on a large financial corpus. This additional training enables the model to grasp the subtleties of financial language, including domain-specific vocabulary and semantic relationships, thereby enhancing its ability to accurately assess sentiment in different financial texts. It demonstrates that this domain-specific adaptation significantly improves FinBERT's performance in financial sentiment analysis compared to other methods, including traditional machine learning approaches and even other pre-trained language models like ELMo and ULMFit [12].

2.2.3 Hybrid Models

The approach of combining models to leverage the strengths was explored in 'Sentiment analysis from textual data using multiple channels deep learning models' [13]. Their idea of combining CNN and LSTM provides insight into the potential for integrating numerical and textual data in housing market forecasting. We extended this concept to application, which combined and transformed the sentiment analysis model into a quantitative model.

2.2.4 Evaluation Metrics for Model Selection

For accuracy metrics, we followed the approach recommended in 'An Introductory Study on Time Series Modeling and Forecasting' by Adhikari and

Agrawal [14], and 'Combination of time series analysis and sentiment analysis for stock market forecasting' [15]. These studies agree on the use of Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) as common metrics to evaluate the accuracy of forecasting models. We adopted these metrics to quantify the difference between predicted and actual values during back-testing, with lower values indicating better accuracy.

2.2.5 Large Language Models

In studying the potential of LLMs for this project, we focused on two main directions, the first one is sentiment analysis. 'Large Language Models for Aspect-Based Sentiment Analysis' introduces the performance of LLMs, as demonstrated by a fine-tuned GPT-3.5 model, in the area of Aspect-Based Sentiment Analysis (ABSA) [16]. According to this paper, ABSA provides a more detailed analysis of opinions expressed in text than simply classifying overall sentiment. This paper offers insights into the latest sentiment analysis abilities of LLMs.

The second direction of our study is time series prediction. LLMs are traditionally used for NLP tasks, but some papers demonstrate their ability to perform time series analysis [17]. For example, the paper 'Time-LLM: Time Series Forecasting by Reprogramming Large Language Models' demonstrates that the TIME-LLM framework can reprogram time series data and uses a technique called Prompt-as-Prefix to enhance the LLM's reasoning abilities with this data [18].

In addition to TIME-LLM, we also studied TimeGPT, another framework that reprograms LLMs originally intended for text [19]. However, we finally decided to use Lag-Llama [20], a probabilistic forecasting model trained to output a probability distribution based on LLaMA by Meta AI [21], rather than the other two models. This decision was made because Lag-Llama's architecture inherently understands temporal patterns through its unique tokenization scheme, which uses lags as covariates, making it better suited for our studies of housing market time series prediction [20].

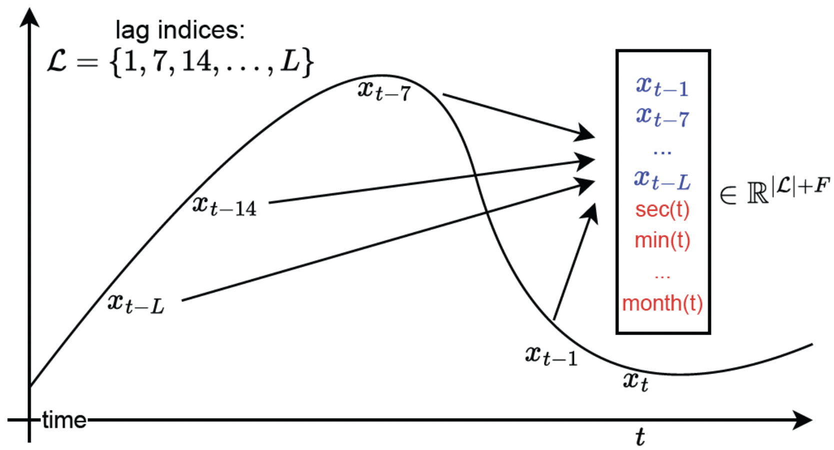


Figure 4: Lag features constructed of Lag-Llama [20]

3 Methodology

3.1 Data and Preparation

When it comes to predicting housing market trends, the quality and preparation of data are important factors on building successful models. For this dissertation, we combined quantitative economic indicators with various qualitative textual data to create a comprehensive dataset for the housing markets in Hong Kong and the United Kingdom. Proper data processing help us to transform raw data into a coherent format which is ready for the models to train and analysis.

In this section, we will explain the steps to clean, standardise, and align the datasets, along with the reasoning behind these decisions. The goal here is to build a strong foundation for the NLP and time series analysis models that we will be developing next.

3.2 Data Sources and Collection

3.2.1 Quantitative Data Sources

First of all, we gathered a set of important quantitative data that including important economic indicators which influencing the housing market. For both Hong Kong and the United Kingdom, we collected data on:

- Gross Domestic Product (GDP)
- Housing Price Index
- Interest Rates
- Unemployment Rates

We made sure to source all these datasets from government departments in Hong Kong and the United Kingdom to ensure they were reliable and authentic. For instance, we obtained GDP data for Hong Kong from the Hong Kong Census and Statistics Department [22], while the United Kingdom's GDP data came from the Office for National Statistics [23].

When it came to the housing price index, which was crucial for training and labelling our models, we used different sources tailored to each market. For Hong Kong, we relied on the Centa-City Leading Index [24]. Though this index was not published from government department, it's highly regarded because it provides comprehensive, regularly updated data that reflects overall market trends, regional variations, and different unit sizes, which makes it a reliable tool for analysing the housing price dynamics of Hong Kong.

For the United Kingdom, we used the United Kingdom Government Housing Index [25]. This index, which updated monthly, offers authoritative data on house prices across the entire United Kingdom, capturing regional differences and trends. It's a dependable and thorough resource for effectively analysing the United Kingdom housing market.

The figures below illustrate the correlation of the housing index by emphasising the strong inverse relationship between the housing index and the unemployment rate and interest rate, with lagging effects.

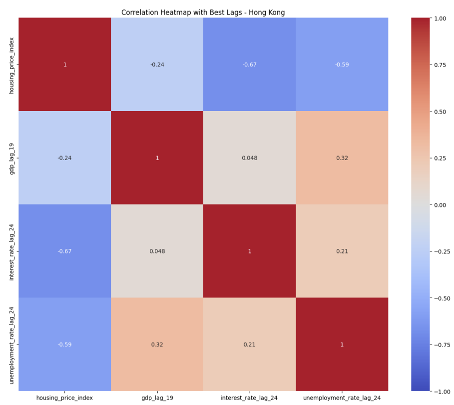


Figure 5: Correlation Heatmap of HK Qualitative Data

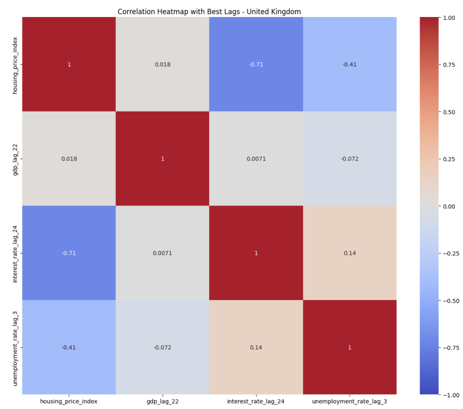


Figure 6: Correlation Heatmap of the UK Qualitative Data

3.2.2 Qualitative Data Sources

For the qualitative data, we gathered textual information from a variety of sources.

In the case of Hong Kong, we used annual Hong Kong Property Review reports [26], along with monthly market reports from Knight Frank [27] and JLL [28]. Both monthly reports regarding the Hong Kong housing market are provided by experienced professionals with in-depth analysis, making them reliable. We also included online forum discussions from platforms like geoexpat.com(http://geoexpat.com/), as well as relevant social media posts, to capture a broader range of perspectives.

For the United Kingdom, we relied on monthly government housing reports [29] and Propertymark's monthly market reports [30]. To add more depth, we also included discussions from the House Price Crash forum [31] and other relevant social media posts as part of our dataset.

3.3 Data Collection Process

We can organise the above quantitative and qualitative data into the following table:

Data Source	Location	Data Type	Frequency	Time Range	Importance
housing price index	HK/UK	quantitative	Weekly	1999 - 2024	N/A
GDP	HK/UK	quantitative	Quarterly	1990 - 2024	N/A
interest rates	HK/UK	quantitative	Weekly	1998 - 2024	N/A
unemployment rates	HK/UK	quantitative	Yearly	1960 - 2024	N/A
Hong Kong Property Review	HK	qualitative	Yearly	1980 - 2024	High
Knight Frank	HK	qualitative	Monthly	2012 - 2024	Medium
JIL	HK	qualitative	Monthly	2018 - 2024	Medium
geoexpat.com	HK	qualitative	N/A	2003 - 2024	Low
UK gov	UK	qualitative	Monthly	2016 - 2024	High
propertymark	UK	qualitative	Monthly	2017 - 2024	Medium
House Price Crash Forum	UK	qualitative	N/A	2010 - 2024	Low

All the above data was collected by downloading from official websites and extracting plain text from PDF files. For the quantitative data, we gathered information spanning from 2000 to 2024. However, the qualitative data covers a shorter time range, such as from 2018 to 2024, due to various restrictions and limitations from the sources.

3.3.1 Quantitative Data Processing

Initial Data Loading First, we loaded the quantitative data, mostly in CSV format, using the Python pandas library. Each dataset was loaded into separate DataFrame objects:

```
hk_gdp =
pd.read_csv('./Data/Quantitative/gdp_hk.csv', delimiter=',')
```

Date Conversion and Indexing Next, we processed the DataFrame to ensure the data was in the correct format and set the date as the index. For example:

```
hk_gdp['DateTime'] = pd.to_datetime(hk_gdp['DateTime'])
hk_gdp.set_index('DateTime', inplace=True)
```

This step was to ensure the datasets were properly aligned with the correct time periods, especially for later resampling operations.

Data Resampling One of the main challenges with this dataset was the varying frequencies at which indicators were reported, while some data came in weekly and others monthly. To create a unified dataset, we resampled all data to a monthly frequency. This involved aggregating higher frequency data to monthly values and using a forward-fill method for lower frequency data to generate monthly data points.

```
hk_monthly_housing_price_index =
hk_housing_price_index['CCL'].resample('M').mean()
hk_monthly_gdp_change = hk_gdp['Value'].resample('M').ffill()
```

We chose these resampling methods(mean for higher frequency data and forward-fill for lower frequency data) to preserve the data's characteristics while creating a consistent monthly time series.

Handling Missing Values Even though we sourced the data from authoritative sources, there were still some missing values. We addressed this by using the forward-fill method, which assumes the last known value is valid until a new one is available. This is a reasonable approach for various types of economic data.

```
hk_monthly_interest_rate =
hk_interest_rate['Value'].resample('M').ffill()
```

Data Normalisation Finally, for the quantitative data, normalisation is important to ensure that all data features are on a comparable scale, which helps prevent the models from being biased toward features with larger magnitudes. While this step wasn't handled immediately, we planned to use different techniques like min-max scaling or standardisation later, depending on the type of quantitative data and the model being trained.

3.3.2 Qualitative Data Processing

Text Data Loading Qualitative data is very important for our models, as it significantly determines the quality of the training process. To ensure high-quality input, we processed the textual data through several steps. First, we loaded text from individual files, which each representing a specific report, article, or post from social media or forums. All the files stored in the folder were systematically loaded as follows:

```
for filename in os.listdir(folder_path_hk):
    if filename.endswith('.txt'):
        text_file_set_hk.append(filename)
```

Text Cleaning After different studies and experiments conducted above, we decided on a specific sequence of regular expressions for text cleaning:

```
patterns = [
    re.compile(r'\\[page \\d+\\]'),  # Remove page numbers
    re.compile(r'^.*?\\n\\n', flags=re.DOTALL),  # Remove headers
    re.compile(r'\\n\\n.*?$', flags=re.DOTALL),  # Remove footers
    re.compile(r'\\s+'),  # Replace multiple spaces with a single space
    re.compile(r'\\s+([.,!?;:])'),  # Remove spaces before punctuation
    re.compile(r'[^a-zA-Z0-9.,!?;:\\s]'),  # Remove special characters except spe
    re.compile(r'\\b\\w{1}\\b'),  # Remove single-character words
    re.compile(r'\\b\\d+\\b'),  # Remove standalone numbers
]
```

Regular expressions are powerful tools in NLP data processing because they allow for precise and flexible pattern matching, validation, and modification, which is the key objective at this stage. We began by removing page numbers, headers, and footers that appear during text extraction from PDF files. We then reduced multiple spaces to a single space and removed spaces before punctuation to ensure consistency.

Special characters, except for essential punctuation, were removed, along with single-character words and standalone numbers. These steps help to minimise noise, resulting in cleaner text for NLP processing.

This order was carefully designed to standardise the text data systematically, ensuring that it remains logical and coherent after cleaning, making it ready for further analysis.

Tokenisation After cleaning, the text data was tokenised, breaking it down into fundamental units or tokens. We used the word_tokenize function from the NLTK library for this process. NLTK's word_tokenize function is highly effective for this task because it handles complex tokenisation scenarios, making it one of the most reliable tools for precise text tokenisation in the domain of NLP [32].

Stop Word Removal To improve the quality of the textual data, we removed common words that usually do not contribute significantly to the text's meaning. This step reduced noise and helped to focus on the most meaningful words [33].

We also used NLTK's English stop words corpus for this step:

```
stop_words = set(stopwords.words('english'))
tokens = [token for token in tokens if token.isalpha()
and token not in stop_words]
```

Lemmatisation After removing stop words, we reduced words to their base or dictionary form through lemmatisation. To make it more accurate, we considered each word's part of speech using NLTK's part-of-speech tags. The get_wordnet_pos function maps NLTK's part-of-speech tags to WordNet's format.

Here is an example of report text after the above processing:

```
cleaned_dataset_hk['KF202211.txt']
```

```
'buyer cautious amid interest rate hike weaken economy residential market sentiment sluggish amid continued rise interest rate worsen l
ocal economy fluctuate stock market potential buyer hesitant enter property market lead poor performance transaction volume price late
data land registry show total residential sale record october drop mom primary sale plunge drastically mom home overall residential hom
e price decrease mom september accord rating valuation department reach low level since january accumulate drop since january record un
it sell market value mass market give uncertain market condition developer also reduce ask price new project attract buyer positive not
e luxury market relatively stable one notable transaction record month sq ft house mont rouge beacon hill sell hk million hk per sq tra
nsaction price high year first hand villa kowloon lease front local move remain core driver overall home rental index september rise fi
fth straight month significant leasing transaction include sq ft unit ultima ho man tin lease hk per month tenant take opportunity down
ward market upgrade apartment amount homeowner willing offer discount tenant potential interest rate hike couple weak economic growth e
xpect put pressure overall home price near term dampen market sentiment particular mass market forecast drop home price mass residentia
l market flat performance luxury home price full year'
```

Figure 7: Text from Knight Frank Report of Nov 2022 after processing

3.4 Ensuring Data Quality

We encountered several challenges during data processing and developed few strategies to overcome them. First of all, we used resampling techniques to address the different data frequencies in both the quantitative and qualitative datasets. To handle inconsistent data formats, we implemented a standardised parsing process across all datasets.

We also established a comprehensive data processing approach to ensure that raw textual data from reports and discussions were thoroughly prepared for model training. This process included multiple experiments with cleaning and lemmatisation techniques to determine the most effective methods for text processing.

After completing these data processing steps, we ended up with a clean, well-aligned dataset that integrates both quantitative economic indicators and qualitative data for Hong Kong and the United Kingdom housing markets. This dataset now served as the foundation for market sentiment analysis, ready for the application of NLP and time series prediction models.

4 Model Specification

4.1 Quantitative Model Implementation

To establish a robust baseline for our NLP prediction models, we carefully selected a quantitative model: the Seasonal Autoregressive Integrated Moving Average with exogenous variables (SARIMAX) [34]. We chose SARIMAX because it can capture both seasonal patterns and the impact of external factors [35], which in this case, are the economic indicators we have integrated into our datasets.

4.1.1 Baseline Quantitative model - SARIMAX

The SARIMAX model is specified as follows: [36]

$$\Theta(L)^p \theta(L^s)^P \Delta^d \Delta_s^D y_t = \Phi(L)^q \phi(L^s)^Q \Delta^d \Delta_s^D \epsilon_t + \sum_{i=1}^{n} \beta_i x_t^i$$

Where:

- (p,d,q) are the non-seasonal parameters
- (P,D,Q) are the seasonal parameters
- s is the number of periods per season

These parameters are not set in advance, but are determined through a grid search process to optimise the model's performance for each specific market context.

As mentioned earlier, the quantitative dataset includes GDP figures, interest rates, and unemployment rates for both Hong Kong and the United Kingdom from 2000 to 2024, with housing price indices as the target variable. This preprocessed data, prepared during the data preparation stage, will be used to train the model and predict housing market dynamics.

4.1.2 Parameter Optimisation

To identify the optimal SARIMAX parameters, we used a grid search function to explore different combinations of p, d, and q values. Specifically, the values for p and q ranged from 0 to 2, while d values ranged from 0 to 1. We also tested various trend specifications, including no trend, constant, linear, and quadratic trends. The grid search function allowed us to find the best set of parameters for the SARIMAX model by fitting the training data and minimising the Mean Absolute Error (MAE), which we used as the primary metric to evaluate performance.

4.1.3 Performance Evaluation

To evaluate the performance of the SARIMAX models, we used three key error assessment methods to gain a comprehensive view of the results:

- Mean Absolute Error (MAE): This measures the average magnitude of errors in the forecast, without considering whether the errors are positive or negative.
- Mean Squared Error (MSE): This measures the average squared difference between the predicted values and the actual values, placing more emphasis on larger errors.
- Root Mean Squared Error (RMSE): This is the square root of the MSE, measures the standard deviation of the residuals and translating the error back into the original units of measurement.

While MAE treats all errors equally, MSE places greater weight on larger errors, and RMSE translates this into a more interpretable unit of measurement. Together, these three error metrics provide different perspectives on the accuracy and reliability of the model's predictions.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

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

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

4.1.4 Model Interpretation

This implementation of quantitative SARIMAX model establishes an important benchmark, helping to compare the improvements achieved by the using of NLP models later on. This baseline serves a statistical purpose, allowing us to make meaningful comparisons between traditional quantitative methods for predicting market trends that incorporate sentiment analysis from contemporary texts in the context of housing market trend forecasting.

4.2 LSTM Model Implementation

After establishing the quantitative baseline model, we moved on to develop our first NLP model: a Long short-term memory neural network. LSTM model, a type of recurrent neural network (RNN), are particularly suitable for capturing long-term dependencies in sequential data, making them ideal for analysing textual data related to housing markets.

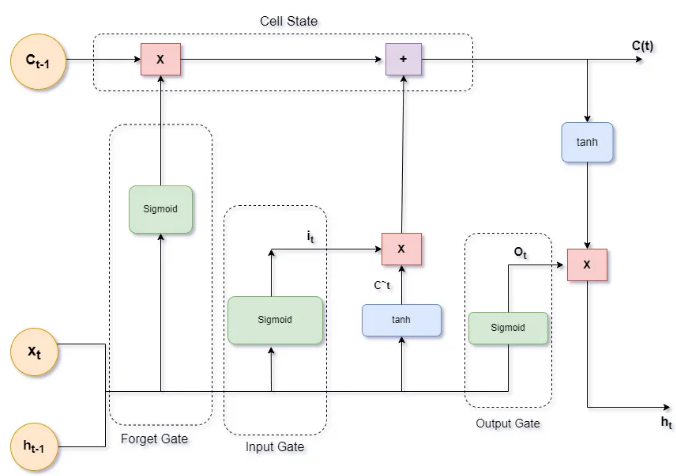


Figure 8: Long short-term memory neural network Architecture [37]

4.2.1 Model Architecture

To effectively interpret textual data in the housing market and prepare it for sentiment analysis, each layer of the LSTM model architecture serves a specific function within the deep learning pipeline.

Data Splitting For consistency across all models, we split the dataset into training and validation sets using an 80-20 ratio, applying the 'random_split' function. We used 'DataLoader' to batch the data, setting a batch size of 8 and enabling shuffling for the training set.

Embedding Layer The embedding layer, which serves as the initial layer of the neural network, converts tokenised sequences into dense vector representations. This layer learns continuous vector representations for the words in the vocabulary, allowing the model to capture semantic relationships between them. In our case, the embedding dimension was set to 100.

LSTM Layer The core of the model consists of two stacked LSTM layers, each with 50 units. These layers are critical for processing the sequential nature of the text data. The first LSTM layer processes the embedded sequences from the embedding layer, refining temporal features, and passes them on to the second layer, which extracts sequence information into a fixed-size representation for the final output. This stacked design enables the model to capture both short-term dependencies (in the first layer) and long-term dependencies (in the second layer) within the textual data.

Dropout To prevent overfitting, we included a dropout layer after each LSTM layer, with a dropout rate of 0.2. This means that during training, 20% of the input units are randomly set to 0 at each update. This technique reduces interdependent learning among neurons, enhancing the model's ability to generalise and learn more robust features.

Dense Output Layer The final layer of the neural network is a fully connected layer with a single neuron using a hyperbolic tangent (tanh) activation function. This neuron produces a single continuous output in the range of [-1, 1], representing the normalized sentiment scores for the corresponding period. This range represents the result captured for both positive and negative sentiments in the housing market, while -1 means the most negative sentiment value and 1 means the most positive sentiment value.

Early Stopping To further prevent overfitting, we implemented an early stopping mechanism with a patience of 2 epochs. This means that training would stop once if the validation loss did not improve for two consecutive epochs, which is a reasonable approach for a model of this small size.

Hyperparameters The LSTM model was implemented using TensorFlow's Sequential architecture. Through forward and backward propagation during training, the model learns to capture patterns in the text that relate to the target variable.

After empirical testing, hyperparameters such as the number of units (50) and the dropout rate (0.2) were carefully selected to strike a balance between the LSTM model's capacity and its generalization ability. These parameters help the model capture the dynamics of the housing market while avoiding overfitting to market noise. We applied the same architecture to both Hong Kong and the United Kingdom housing market predictions to ensure that the results were comparable. This consistency also allowed us to explore and compare the differences between these two geographical markets.

```python
def create_model():
    model = Sequential()
    model.add(Embedding(input_dim=len(tokenizer.word_index) + 1,
    output_dim=100))
    model.add(LSTM(50, return_sequences=True))
    model.add(Dropout(0.2))
    model.add(LSTM(50, return_sequences=False))
    model.add(Dropout(0.2))
    model.add(Dense(1, activation='tanh'))
    # Tanh activation for output in range [-1, 1]
    model.compile(optimizer='adam', loss='mean_squared_error')
    return model
```

```
23/23 ━━━━━━━━━━━━━━━━━━━━ 4s 126ms/step - loss: 0.3191 - val_loss: 0.1547
Epoch 2/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 115ms/step - loss: 0.1354 - val_loss: 0.0710
Epoch 3/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 115ms/step - loss: 0.0493 - val_loss: 0.0491
Epoch 4/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 116ms/step - loss: 0.0379 - val_loss: 0.0483
Epoch 5/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 117ms/step - loss: 0.0288 - val_loss: 0.0443
Epoch 6/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 116ms/step - loss: 0.0232 - val_loss: 0.0413
Epoch 7/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 116ms/step - loss: 0.0287 - val_loss: 0.0621
Epoch 8/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 115ms/step - loss: 0.0318 - val_loss: 0.0509
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 124ms/step
MAE for current fold: 0.1675530672583305
Epoch 1/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 4s 127ms/step - loss: 0.3134 - val_loss: 0.0787
Epoch 2/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 117ms/step - loss: 0.0706 - val_loss: 0.0645
Epoch 3/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 116ms/step - loss: 0.0495 - val_loss: 0.0461
Epoch 4/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 117ms/step - loss: 0.0322 - val_loss: 0.0665
Epoch 5/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 116ms/step - loss: 0.0480 - val_loss: 0.0531
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 129ms/step
MAE for current fold: 0.17997879020058372
Epoch 1/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 4s 128ms/step - loss: 0.3894 - val_loss: 0.2033
Epoch 2/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 117ms/step - loss: 0.0735 - val_loss: 0.1039
Epoch 3/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 115ms/step - loss: 0.0467 - val_loss: 0.1335
Epoch 4/10
23/23 ━━━━━━━━━━━━━━━━━━━━ 3s 116ms/step - loss: 0.0522 - val_loss: 0.1428
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 123ms/step
MAE for current fold: 0.20292967158919908
```

Figure 9: LSTM Model Cross-Validation Training Result

4.3 FinBERT Model Implementation

The second sentiment analysis model we implemented is the Bidirectional Encoder Representations from Transformers (BERT) model, which is a relatively new approach in the NLP field and has shown exceptional performance in various text analysis tasks, including sentiment analysis.

4.3.1 Transfer Learning and Model Architecture

The foundation of our BERT model is FinBERT, a pre-trained BERT model developed by "ProsusAI/finbert," [20] which is specifically designed for financial sentiment analysis.

To prepare our datasets for the FinBERT model, we applied a tokeniser,

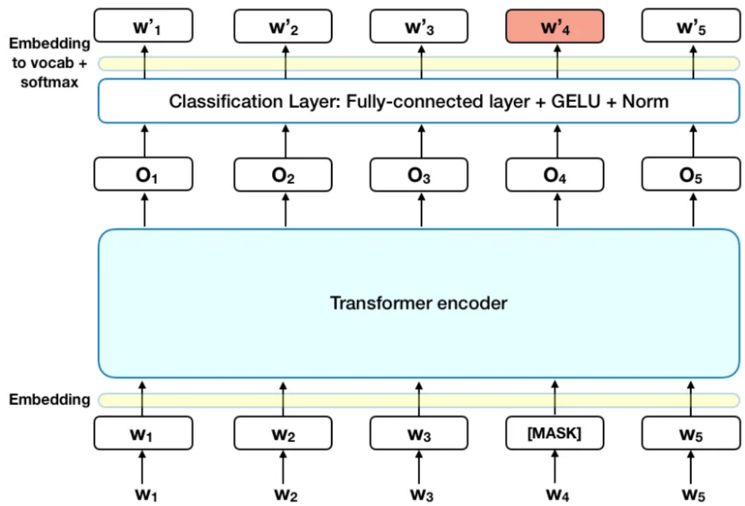


Figure 10: Bidirectional Encoder Representations from Transformers model Architecture [38]

setting parameters for truncation (max_length=512) and padding to ensure a consistent input size. This step converts our raw text data into a format suitable for processing by the BERT model.

To tailor FinBERT to our specific task of sentiment analysis, which ranges from [-1, 1] rather than the original classification task, we replaced the final classification layer with a linear regression layer. Such modification allows the model to output a continuous sentiment score. Therefore, the input data is processed through the BERT layers, and the regression layer is applied to the pooled output, producing a single sentiment score during training and evaluation.

We used an Adam optimizer with a learning rate of 1e-5, a relatively low value chosen to prevent overshooting during the fine-tuning process of the pre-trained BERT model. This learning rate was determined to be effective through various experiments.

To prevent exploding gradients, a common issue in training deep neural networks, we implemented gradient clipping with parameter maximum norm of 1.0.

The data splitting and early stopping techniques used in the LSTM model were also applied here. For the loss function, we chose Mean Squared Error (MSE), which is well-suited for regression tasks.

```python
# Define the model
class FinBERTForRegression(nn.Module):
    def __init__(self):
        super(FinBERTForRegression, self).__init__()
        self.bert =
        AutoModelForSequenceClassification.from_pretrained("ProsusAI/finbert")

        # Replace the classifier layer with a regression layer
        self.bert.classifier = nn.Linear(self.bert.classifier.in_features, 1)

    def forward(self, input_ids, attention_mask):
        outputs = self.bert.bert(input_ids=input_ids,
        attention_mask=attention_mask)
        pooled_output = outputs[1]
        # Get the pooled output from the BERT model
        logits = self.bert.classifier(pooled_output)
        # Apply the regressor
        return logits

model = FinBERTForRegression()

# Define the loss function
criterion = nn.MSELoss()

# Define a smaller learning rate and use gradient clipping
optimizer = torch.optim.Adam(model.parameters(), lr=1e-5)
# Reduced learning rate
```

```
Training Epoch 1/20: 100%|██████████████████████| 13/13 [01:10<00:00,  5.42s/it]
Epoch 1/20, Average Training Loss: 0.326114956002969
Epoch 1/20, Average Validation Loss: 0.18194952979683876
Training Epoch 2/20: 100%|██████████████████████| 13/13 [01:07<00:00,  5.23s/it]
Epoch 2/20, Average Training Loss: 0.13750716040913874
Epoch 2/20, Average Validation Loss: 0.13535334169864655
Training Epoch 3/20: 100%|██████████████████████| 13/13 [01:11<00:00,  5.52s/it]
Epoch 3/20, Average Training Loss: 0.10816901564024962
Epoch 3/20, Average Validation Loss: 0.1286834478378296
Training Epoch 4/20: 100%|██████████████████████| 13/13 [01:05<00:00,  5.06s/it]
Epoch 4/20, Average Training Loss: 0.06904707662761211
Epoch 4/20, Average Validation Loss: 0.08720746822655201
Training Epoch 5/20: 100%|██████████████████████| 13/13 [01:00<00:00,  4.66s/it]
Epoch 5/20, Average Training Loss: 0.04629139315623503
Epoch 5/20, Average Validation Loss: 0.061210665735416114
Training Epoch 6/20: 100%|██████████████████████| 13/13 [01:01<00:00,  4.73s/it]
Epoch 6/20, Average Training Loss: 0.038119776389346674
Epoch 6/20, Average Validation Loss: 0.07971485238522291
Training Epoch 7/20: 100%|██████████████████████| 13/13 [01:00<00:00,  4.66s/it]
Epoch 7/20, Average Training Loss: 0.038677300254885964
Epoch 7/20, Average Validation Loss: 0.053660700970795006
Training Epoch 8/20: 100%|██████████████████████| 13/13 [01:05<00:00,  5.04s/it]
Epoch 8/20, Average Training Loss: 0.03610716946423054
Epoch 8/20, Average Validation Loss: 0.13707260973751545
Training Epoch 9/20: 100%|██████████████████████| 13/13 [01:05<00:00,  5.03s/it]
Epoch 9/20, Average Training Loss: 0.02989088134983411
Epoch 9/20, Average Validation Loss: 0.058924928307533264
Early stopping applied after epoch 9
Training stopped early due to lack of improvement.
```

Figure 11: BERT Model Cross-Validation Training Result

4.4 Lag-Llama Model Implementation

The third NLP model we explored is the implementation of LLMs, which we used Lag-Llama model, a specialized variant of LLMs designed for time series analysis within the LLM framework.

We began by loading the model from a pre-trained checkpoint, and same as our BERT model approach, we leveraged transfer learning to fine-tune the model tailor on our dataset. This approach allows the model to adapt its sequential capability to process our housing market data from Hong Kong and the United Kingdom, capturing patterns in both the time series data and the sentiment scores derived from textual data.

4.4.1 Dataset Creation and Model Configuration

To ensure that our housing market sentiment dataset was compatible with the Lag-Llama model, we created a dataset using the GluonTS framework. This involved setting a 'PeriodIndex' with monthly frequency during the dataset creation process, which was crucial for maintaining the alignment and readability of the dataset for the Lag-Llama model.

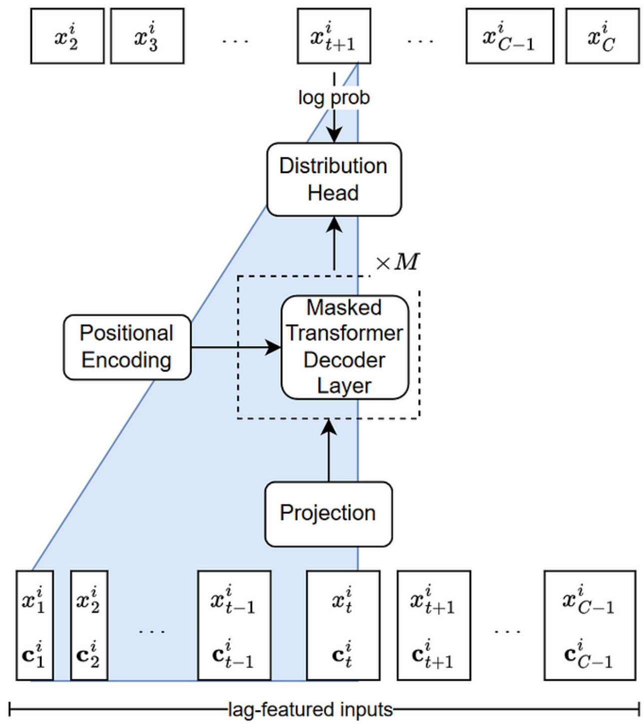


Figure 12: Architecture of Lag-Llama. Image from Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting [20]

We configured our Lag-Llama estimator with different hyperparameter values, including the prediction length, context length, and specific model architecture details such as the number of layers and embedding dimensions.

Unlike the LSTM and BERT models, the Lag-Llama model is a probabilistic forecasting model. It predicts and generates forecasts for a specified number of future time steps using a multi-sample approach. This means that the model generates different prediction scenarios to capture the uncertainty inherent in forecasting, which can be a useful feature in the context of housing markets, where future trends can be influenced by a wide range of factors.

For visualisation, rather than relying solely on metrics for comparison as with the LSTM and BERT models, we used a plotting function to display both the actual historical data and the model's forecasts during the backtest.

This visualisation includes confidence intervals, providing a clear view of the model's certainty and robustness across different time points. This approach not only aids in analysis but also serves as an effective way to present the results.

The implementation of the Lag-Llama model for housing market prediction showcases the application of state-of-the-art machine learning techniques for time series forecasting. This methodology successfully combines advanced time series modeling with the capabilities of LLMs.

```
estimator = LagLlamaEstimator(
    ckpt_path="lag-llama.ckpt",
    prediction_length=prediction_length,
    # pretrained length
    context_length=128,
    # estimator arguments
    input_size=estimator_args["input_size"],
    n_layer=estimator_args["n_layer"],
    n_embd_per_head=estimator_args["n_embd_per_head"],
    n_head=estimator_args["n_head"],
    scaling=estimator_args["scaling"],
    time_feat=estimator_args["time_feat"],
    batch_size=1,
    num_parallel_samples=100,
)
```

5 Discussion of Results

5.1 Quantitative Model

HK	Metric	Value
	Average MAE	0.2979
	Average MSE	0.3878
	Average RMSE	0.3613
UK	**Metric**	**Value**
	Average MAE	0.1876
	Average MSE	0.2856
	Average RMSE	0.2222

Table 1: Performance metrics of the HK and UK quantitative models (refer to Appendices A and B for full details)

The results demonstrate the basic ability of the SARIMAX quantitative time series prediction model in forecasting housing market trends in Hong Kong and the United Kingdom from 2014 to 2024. The SARIMAX model integrates exogenous variables: GDP, interest rates, and unemployment rates, to capture the housing price dynamics.

In terms of performance, these relatively low values indicate that the model's ability to predict the housing price index, with the backtest predictions aligning with actual market trends. This also suggests that the chosen exogenous variables were valuable in enabling the quantitative model to predict trends accurately.

The different optimal parameters selected through the grid search method demonstrate the robustness of the cross-validation approach, validating the use of SARIMAX as a baseline model for quantitative forecasting in the housing market.

5.2 LSTM Model

		Quantitative model	LSTM model	Improvement (%)
HK	**Average MAE**	0.2979	0.1675	-43.77%
	Average MSE	0.3878	0.1379	-64.44%
	Average RMSE	0.3613	0.1675	-53.64%
UK	**Average MAE**	0.1876	0.1772	-5.54%
	Average MSE	0.2856	0.1593	-44.22%
	Average RMSE	0.2222	0.1772	-20.25%

Table 2: Performance metrics and decrease in percentage of error of the HK and UK LSTM hybrid models (refer to Appendices C and D for full details)

The results indicate that integrating sentiment analysis generally improved the overall model performance in capturing market trends for both the Hong Kong and the United Kingdom housing markets. However, the differing model performance between Hong Kong and the United Kingdom demonstrates the challenges NLP models face during periods of different market trends and changes. The lower error rates in the Hong Kong market compared to the United Kingdom suggest that the model's effectiveness can vary significantly depending on market conditions and the specific nature of the housing market.

These findings highlight the importance of incorporating both quantitative and qualitative data in housing market predictions, particularly in volatile markets like Hong Kong. They also underscore the potential of hybrid models in enhancing predictive accuracy during periods of high market uncertainty.

The density plots of errors below also show that the error values become concentrated in a lower value near zero in range, which shows the improvement of hybrid model over the quantitative model.

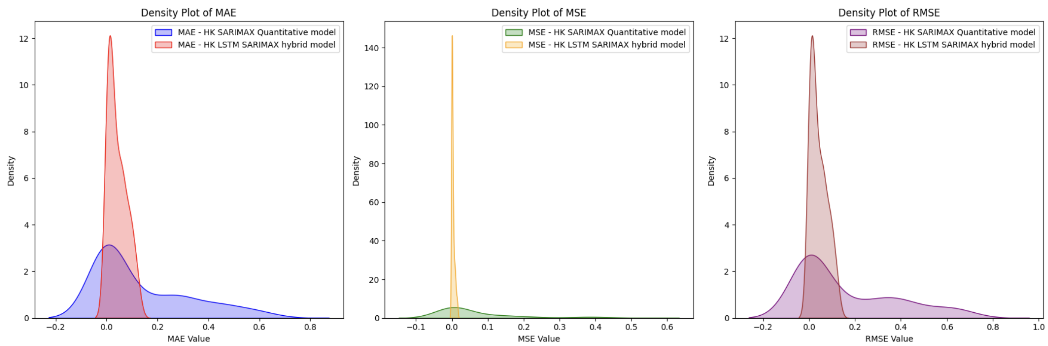


Figure 13: Density plots of MAE, MSE, and RMSE comparing the HK SARIMAX Quantitative model and HK LSTM SARIMAX hybrid model

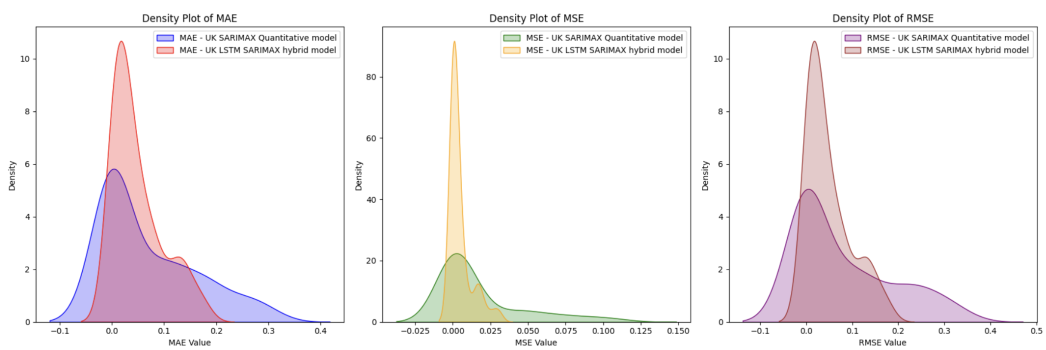


Figure 14: Density plots of MAE, MSE, and RMSE comparing the UK SARIMAX Quantitative model and UK LSTM SARIMAX hybrid model

5.3 FinBERT Model

Refer to table 3, the results show that integrating the FinBERT model for sentiment analysis with the SARIMAX quantitative model for housing market forecasting have similar outcomes compared with LSTM model. In the more volatile Hong Kong market, the BERT model demonstrated strengths in handling sentiment-driven textual data, indicating improved performance in scenarios involving extreme sentiment shifts.

In the United Kingdom market, where volatility is lower, BERT-based models faced challenges, while including a tendency to classify text with

		Quantitative model	BERT model	Improvement (%)
HK	**Average MAE**	0.2979	0.1621	-45.59%
	Average MSE	0.3878	0.1172	-69.78%
	Average RMSE	0.3613	0.1621	-55.13%
UK	**Average MAE**	0.1876	0.1559	-16.90%
	Average MSE	0.2856	0.1555	-45.55%
	Average RMSE	0.2222	0.1559	-29.84%

Table 3: Performance metrics and decrease in percentage of error of the HK and UK BERT hybrid models (refer to Appendices E and F for full details)

neutral sentiment scores, which limiting their impact on forecasting accuracy. Though the United Kingdom market's error values showed less improvement compared with the Hong Kong market, they still outperformed a bit compared with LSTM model, referring to table 4.

The density plots of errors below also show a similar result with LSTM model, which show the improvement of error value distribution compared with the quantitative model.

		LSTM model	BERT model
HK	**Average MAE**	-43.77%	-45.59%
	Average MSE	-64.44%	-69.78%
	Average RMSE	-53.64%	-55.13%
UK	**Average MAE**	-5.54%	-16.90%
	Average MSE	-44.22%	-45.55%
	Average RMSE	-20.25%	-29.84%

Table 4: Decrease in percentage of error of LSTM model and BERT model compared to the quantitative model for HK and UK data

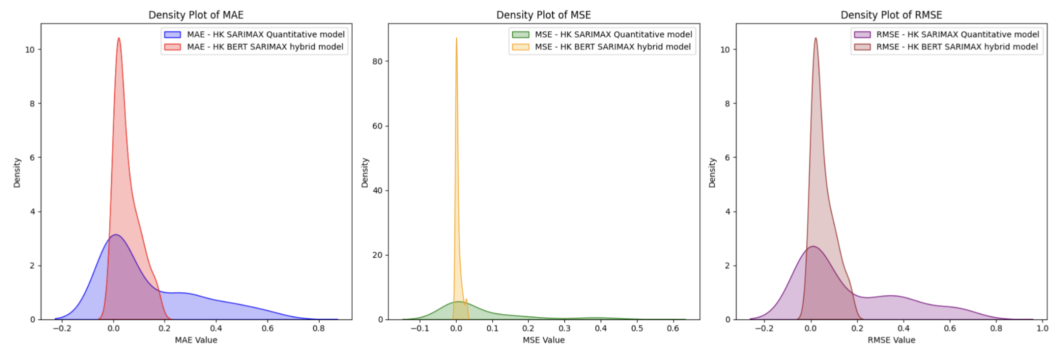


Figure 15: Density plots of MAE, MSE, and RMSE comparing the HK SARIMAX Quantitative model and HK BERT SARIMAX hybrid model

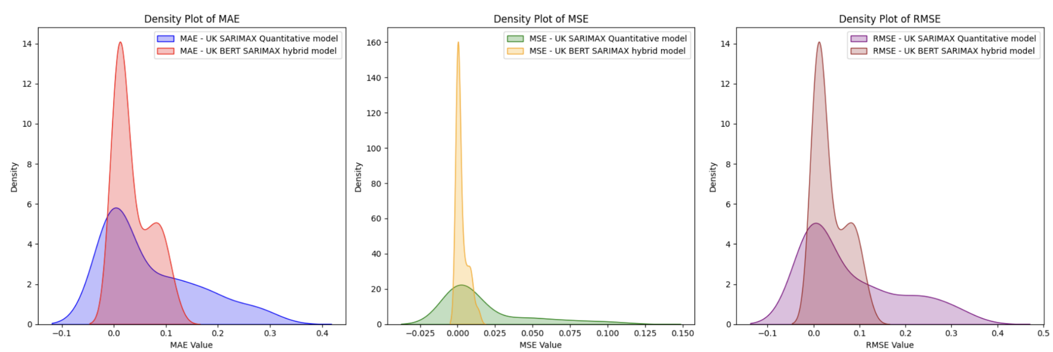


Figure 16: Density plots of MAE, MSE, and RMSE comparing the UK SARIMAX Quantitative model and UK BERT SARIMAX hybrid model

5.4 Lag-Llama Model

The final model we studied was the application of LLMs, specifically the Lag-Llama model. This model demonstrated potential in time series forecasting across the Hong Kong and the United Kingdom datasets, after fine-tuning the pre-trained model using checkpoints and our specific time series data and sentiment scores.

Additionally, the use of 100 parallel samples during the prediction phase improved the model's ability to generate probabilistic forecasts. The multiple layers of the model also helped capture complex patterns and dependencies in the data.

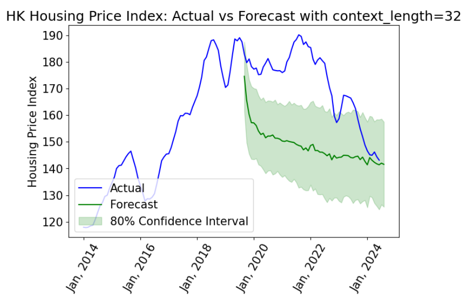


Figure 17: HK Lag-Llama Housing Price Index: Actual vs Forecast with context_length=32

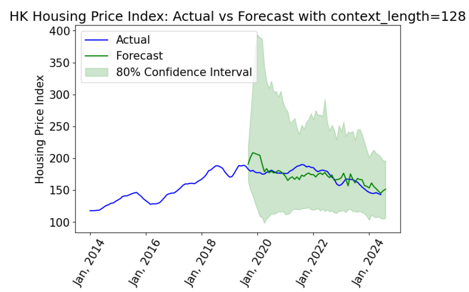


Figure 18: HK Lag-Llama Housing Price Index: Actual vs Forecast with context_length=128

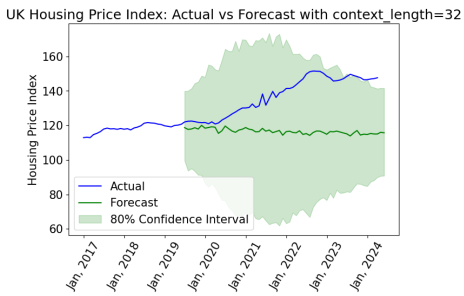


Figure 19: UK Lag-Llama Housing Price Index: Actual vs Forecast with context_length=32

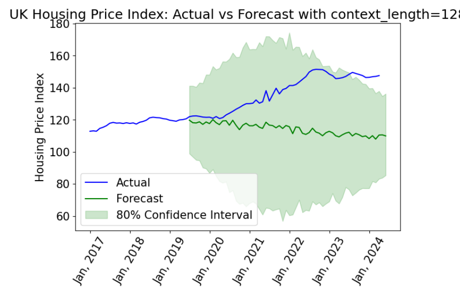


Figure 20: UK Lag-Llama Housing Price Index: Actual vs Forecast with context_length=128

The results above show the actual versus Lag-Llama forecasted housing price indices for Hong Kong and the United Kingdom of different context lengths (32 and 128). It indicates that for Hong Kong market, shorter context lengths (32) result in narrower confidence intervals. For longer context lengths it aligns with the actual data more, but with wider confidence intervals meaning a broader range of potential outcomes. On the other hand, for the United Kingdom market, the Lag-Llama does not perform relatively well with potentially less adaptive forecasts.

5.5 Result Evaluation

The results reveal that different models provide unique insights into both strengths and limitations when forecasting housing market trends in Hong Kong and the United Kingdom. The use of various models, including SARIMAX, LSTM, BERT, and Lag-Llama, offered a diverse range of perspectives on market behaviour and the capacity to handle complex sentiment in textual data.

5.5.1 Hybrid Model Approaches

We employed a hybrid model approach for the LSTM and BERT models with SARIMAX model, integrating quantitative and qualitative data to highlight the potential for enhancing forecasting accuracy by using sentiment analysis on different sources of textual information. There are two goals for such approach: firstly, to directly compare the performance of the NLP technique by observing any decrease in error after applying sentiment scores; and secondly, to provide a more comprehensive view of market trends by capturing both numerical and sentiment factors driving housing prices. This approach helps with incorporating numerical data while focusing on text, combining sentiment scores with actual numerical trends for a more comprehensive analysis.

However, the evaluation also showed that the effectiveness of these hybrid models is context-dependent. In Hong Kong, where textual data contained higher sentiment levels, the integration of sentiment analysis was beneficial. In more stable markets like the United Kingdom, the added complexity of these hybrid models did not always lead to large improvements compared with Hong Kong, due to the generally neutral sentiment in the text due to the market trend.

5.5.2 Model Robustness and Market Conditions

One key observation is the varying performance of models depending on housing market conditions. The Hong Kong housing market, characterised by higher volatility and significant market shift from increasing trend to decreasing trend in recent years, provided richer textual data for the models compared to the more stable United Kingdom market. Textual data with stronger sentiment scores yielded better sentiment predictions, allowing LSTM and BERT models to demonstrate greater effectiveness in Hong Kong due to their ability to capture the market's dynamic nature.

In contrast, the United Kingdom market, with its more gradual trends and lower volatility, benefited less from sentiment analysis models. This suggests that while complex models may perform well in volatile environments, simpler models may suffice in more stable markets. This variation underscores the importance of aligning model complexity with the nature of the data and specific forecasting goals.

5.5.3 Probabilistic Forecasting and Uncertainty

The Lag-Llama model's ability to provide probabilistic forecasts is particularly valuable in predicting markets with high uncertainty, as it allows for a range of potential outcomes rather than a single-point estimate. However, the model also has limitations, such as its inability to be directly compared with the LSTM and BERT models.

Moreover, the accuracy of probabilistic forecasting depends heavily on the model's ability to capture the true distribution of outcomes. In this study, the Lag-Llama model's probabilistic forecasts did not fully capture the extremes of the market in Hong Kong, suggesting room for improvement. This could potentially be addressed through more extensive fine-tuning or by incorporating additional data sources to better model the distribution of outcomes.

5.5.4 Model Interpretability

Another important factor in evaluating these models is interpretability. While advanced models like Lag-Llama may offer strong performance in capturing complex patterns, they often do so at the cost of interpretability.

The SARIMAX model, while simpler, provides a more transparent framework, clearly showing the influence of exogenous variables like GDP, interest rates, and unemployment rates. For the LSTM and BERT models, the sentiment scores are normalised from -1 to 1, making it easier to interpret how these scores are fed back into the quantitative model.

In contrast, the Lag-Llama model functions more as a "black box," making it challenging to understand the exact mechanisms behind its predictions or to seek targeted improvements.

6 Conclusion

This dissertation explored the application of NLP techniques in predicting housing market trends, in the context of Hong Kong and the United Kingdom. By combining quantitative economic indicators with qualitative textual data, we developed and evaluated three NLP models to capture market sentiments and forecast housing price movements. This conclusion part summarises our key achievements, acknowledges the limitations of our approach, and suggests some directions for future work.

6.1 Achievements

Our study has made some significant result in the field of NLP housing market prediction, responding to the research questions listed:

1. **How does the accuracy of NLP models' forecasting compare with traditional forecasting methods when measured against historical data?**

   We have successfully integrated quantitative and qualitative data, creating a comprehensive dataset that merges traditional economic indicators with textual data. Following this integration, we developed NLP models to capture the market sentiments of the textual data. Our results demonstrated that the hybrid models of LSTM and BERT, outperformed traditional forecasting methods with decreased value of error, particularly in volatile market conditions.

2. **What are the performances of different NLP models, such as LSTM, BERT, or LLMs, in the context of housing market forecasting?**

   We successfully implemented these models, including SARIMAX as a quantitative baseline, LSTM and BERT for sentiment analysis, and Lag-Llama as an advanced time series forecasting model. While relatively simple NLP hybrid models like LSTM and BERT demonstrated strong capabilities in capturing complex market dynamics and sentiment patterns, improving the performance of the quantitative baseline model, their effectiveness varied depending on the market conditions. On the other hand, Lag-Llama, representing the LLMs, showed quite promising predictive results, but it is challenging to directly compare with the LSTM and BERT models due to the different behaviors and approaches of these models.

3. **Does the efficacy of NLP models in forecasting housing market trends vary across markets with differing conditions, such as the declining trend in Hong Kong's market and the stability of the UK's market?**

   Our cross-market analysis between Hong Kong and the United Kingdom evaluated the adaptability and robustness of NLP models under different market conditions. The results show that NLP models performed differently across these markets, with greater effectiveness observed in the more volatile Hong Kong market compared to the relatively stable United Kingdom market. This finding suggests that the efficacy of NLP models in forecasting housing market trends does indeed vary across markets with differing conditions.

4. **Has the optimal performance of these models been achieved?**

   While our study has made significant progress, it is difficult to assert whether the best possible performance has been achieved. Especially for the Lag-Llama model, which introduced probabilistic forecasting and offered a range of potential outcomes rather than single-point estimates, indicating that there is still room for improvement. However, for the LSTM and BERT models, we have made our best efforts in fine-tuning, and the performance metrics, which reflected in the decrease in error valuesm, show promising results in this dissertation.

6.2 Limitations

Other than these achievements, our study faced some limitations.

1. **Data Availability and Quality**: The availability of textual data varied between markets and over different period of time. This inconsistency have affected the performance of our NLP models, as in earlier periods there were less digital content available.

Another limitation is the data quality across different sources. Official reports and market analyses generally provided more reliable information, on the other hand we saw social media content introduced noise into the sentiment analysis. To mitigate this, weighted values were assigned to reports based on the authority of the source. However, this approach might have reduced the model's sensitivity to market sentiment.

2. **Model Complexity vs. Market Stability**: We found that more complex NLP models, such as LLMs, did not always outperform simpler models such as LSTM and BERT, especially in stable market conditions like those in the United Kingdom. This highlights the challenge of balancing model complexity with market characteristics.

3. **Interpretability Challenges**: While our LSTM and BERT models provided valuable insights, the Lag-Llama model, despite its strong performance, operated more like a "black box," making it difficult to interpret the exact mechanisms behind its predictions, and compare with LSTM and BERT model.

4. **External Factor Consideration**: While we incorporated several economic indicators, our models may not have captured all external factors influencing housing markets, such as global economic events or demographic shifts. These factors can be considered in future work to increase the accuracy of the model.

6.3 Future Work

Based on our findings and the limitations identified, we propose several directions for future research:

1. **Expand the Data Sources**: Future studies could incorporate a wider range of data sources, to provide a more comprehensive view of market sentiment and trends.

   Another direction of future research is to integrate different data sources, especially social media content, without decreasing the model's accuracy. Additionally, exploring the weighting systems that adjust based on the temporal relevance of different sources is also important, as such approach could potentially leverage the social media sentiment while maintaining the reliability of official reports.

2. **Incorporation of Additional Factors**: Future models could integrate more diverse factors, such as demographic data, environmental indicators, or geopolitical events, to capture a broader range of influences on housing markets.

3. **Adaptive Model Selection**: Creating a framework that dynamically selects or weights different models based on current market conditions could optimize prediction accuracy across various market states. This opens a new direction of future work, based on the prove that our study showed the promising result of improvement of NLP models.

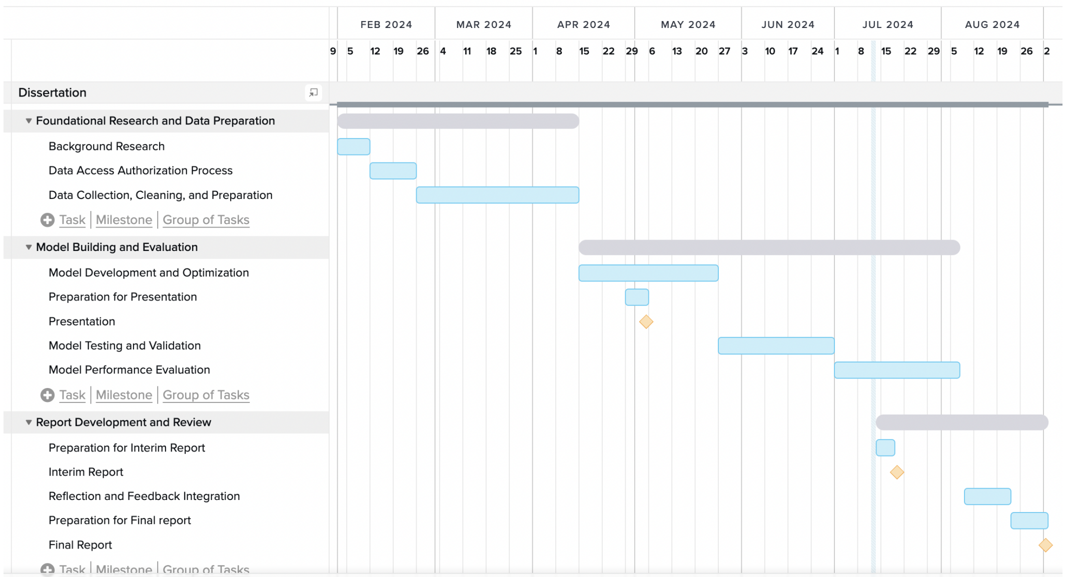
4. **Hybrid Model Development**: Building on the success of combining LSTM and BERT models with traditional forecasting methods, future work could explore more advanced hybrid models that leverage the strengths of such approaches. Moreover, more detailed fine-tuning could be done if more time would be allowed, and this could involve more in-depth study of different types of neural networks.

5. **Comparative Analysis with Economic Models**: Especially for Lag-Llama, which is a probabilistic time series forecasting model, a comprehensive comparison between these NLP-based models and traditional economic forecasting models could be conducted, and the economic models can be act as a baseline to better fine-tune the models or understand the strengths and weaknesses of such approach in different market conditions.

In conclusion, this study has demonstrated the potential of NLP techniques to enhance housing market trend predictions. By combining traditional economic indicators with sentiment analysis derived from textual data, we have shown that it is possible to capture the market dynamics that may not be evident from quantitative data alone. While challenges remain, particularly in model adaptability to different market conditions, the performance of the models can be improved further if have more time. Continued research in this area, addressing the limitations identified and exploring the suggested future directions, has the potential to significantly improve the prediction of housing market trends. This, in turn, can provide valuable insights for different parties to study the housing market underlying sentiment and hence to predict the trend.

7 Project Management and Tools

The Gantt chart below illustrates the project management for this dissertation. The project started from February 2024 and ended August 2024, which includes three main phases: Foundational Research and Data Preparation, Model Building and Evaluation, and Report Development and Review.



7.1 Phase 1: Foundational Research and Data Preparation (February - April 2024)

This initial phase began with background research to understand the time series prediction and the application of NLP in finance.

An important part of this phase was obtaining data access authorisation, ensuring ethical and legal compliance for using proprietary financial data and reports. After that, the process of data collection, cleaning, and preparation was done. This stage heavily relied on the python, jupyter notebook and pandas library, a powerful tool in Python for data manipulation.

7.2 Phase 2: Model Building and Evaluation (April - July 2024)

The core of the project lies in this phase, where different NLP models were developed and evaluated. The process began with model development and optimisation, utilising a combination of tools:

PyTorch This deep learning framework was essential for building and training the LSTM and BERT models.

TensorFlow Used with PyTorch, TensorFlow provided additional capabilities for deep learning model construction.

scikit-learn This machine learning library was important for implementing the SARIMAX model, which served as our quantitative baseline. It also provided essential tools for model evaluation and cross-validation.

Following this, model testing and validation began, where the models were rigorously tested on held-out datasets.

The model performance evaluation stage, from mid-May to early July, included comparing the effectiveness of different NLP approaches by performance metrics.

7.3 Phase 3: Report Development and Review (July - August 2024)

The final phase focused on documenting the research findings and preparing the dissertation report. The interim and final report preparations began in late July, with the project concluding with the final report submission in September.

7.4 Tools and Technologies

Throughout the project, several key tools and technologies were employed:

Adobe Acrobat Since most of the textual data was in PDF format, Adobe Acrobat was used to convert these files into .txt format. This also involves the using of Adobe OCR, to change the images to text.

Python The primary programming language used across all phases of the project, chosen for its rich ecosystem of data science and machine learning libraries.

Jupyter Notebook An interactive development environment that facilitated exploratory data analysis and result visualisation. Its ability to combine code, output, and documentation in a single document was helpful for our dissertation.

PyTorch and TensorFlow These deep learning frameworks were central to implementing the advanced NLP models.

pandas This data manipulation library was essential in handling the diverse datasets, from CSV files of indicators to textual data from various sources. Its data structures (DataFrame and Series) provided the foundation for data preprocessing.

scikit-learn Beyond its use in implementing the SARIMAX model, scikit-learn provided essential tools for data preprocessing, model evaluation, and cross-validation across all models.

matplotlib This visualization libraries were used for creating informative plots and graphs to illustrate model performance and market trends.

References

[1] C. Lai and N. Tam, "Cbre comments on hong kong sar government's second relief package," www.cbre.com.hk, 06 2020. [Online]. Available: https://www.cbre.com.hk/press-releases/hong-kong-holds-spot-as-worlds-priciest-residential-property-market

[2] Y. WANG, "Cooling measures in hong kong and its residential property market: ," Lingnan Scholars, 2022. [Online]. Available: https://scholars.ln.edu.hk/en/publications/cooling-measures-in-hong-kong-and-its-residential-property-market

[3] G. P. Guide, "Hong kong home prices, graphed as time-series," Global Property Guide, 06 2018. [Online]. Available: https://www.globalpropertyguide.com/Asia/Hong-Kong/Home-Price-Trends

[4] "Local house price indices," Savills.co.uk, 2017. [Online]. Available: https://www.savills.co.uk/research_articles/229130/187095-0

[5] A. W. Lo, "Efficient markets hypothesis," *The New Palgrave Dictionary of Economics*, pp. 1–17, 2008. [Online]. Available: https://link.springer.com/referenceworkentry/10.1057/978-1-349-95121-5_42-2

[6] F. Z. Xing, E. Cambria, and R. E. Welsch, "Natural language based financial forecasting: a survey," *Artificial Intelligence Review*, vol. 50, pp. 49–73, 10 2017.

[7] A. Khadjeh Nassirtoussi, S. Aghabozorgi, T. Ying Wah, and D. C. L. Ngo, "Text mining for market prediction: A systematic review," *Expert Systems with Applications*, vol. 41, pp. 7653–7670, 11 2014.

[8] F. Khalil and G. Pipa, "Is deep-learning and natural language processing transcending the financial forecasting? investigation through lens of news analytic process," *Computational Economics*, 07 2021.

[9] S. Siami-Namini and A. S. Namin, "Forecasting economics and financial time series: Arima vs. lstm," arXiv.org, 2018. [Online]. Available: https://arxiv.org/abs/1803.06386

[10] X. Jiawei and T. Murata, "Stock market trend prediction with sentiment analysis based on lstm neural network," Semantic Scholar,

2019. [Online]. Available: https://www.semanticscholar.org/paper/Stock-market-trend-prediction-with-sentiment-based-Jiawei-Murata/54827c06228b557ef64f7a14dfc092a3b31e2cf9

[11] D. Araci, "Finbert: Financial sentiment analysis with pre-trained language models," *arXiv:1908.10063 [cs]*, 08 2019. [Online]. Available: https://arxiv.org/abs/1908.10063

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv.org, 10 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[13] A. Rajesh and T. Hiwarkar, "Sentiment analysis from textual data using multiple channels deep learning models," *Journal of Electrical Systems and Information Technology*, vol. 10, 11 2023.

[14] R. Adhikari and A. R. K, "An introductory study on time series modeling and forecasting," arXiv.org, 2013. [Online]. Available: https://arxiv.org/abs/1302.6613

[15] H.-C. Chou, "Combination of time series analysis and sentiment analysis for stock market forecasting," Digital Commons @ University of South Florida, 2021. [Online]. Available: https://digitalcommons.usf.edu/etd/8749/

[16] P. F. Simmering and P. Huoviala, "Large language models for aspect-based sentiment analysis," arXiv.org, 2023. [Online]. Available: https://arxiv.org/abs/2310.18025

[17] X. Zhang, R. R. Chowdhury, R. K. Gupta, and J. Shang, "Large language models for time series: A survey," arXiv.org, 2024. [Online]. Available: https://arxiv.org/abs/2402.01801

[18] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y. Li, S. Pan, and Q. Wen, "Time-llm: Time series forecasting by reprogramming large language models," *arXiv (Cornell University)*, 10 2023.

[19] S. Garza and M. Mergenthaler-Canseco, "Timegpt-1," *arXiv (Cornell University)*, 10 2023.

[20] K. Rasul, A. Ashok, A. R. Williams, H. Ghonia, R. Bhagwatkar, A. Khorasani, M. J. D. Bayazi, G. Adamopoulos, R. Riachi, N. Hassen, M. Biloš, S. Garg, A. Schneider, N. Chapados, A. Drouin, V. Zantedeschi, Y. Nevmyvaka, and I. Rish, "Lag-llama: Towards foundation models for probabilistic time series forecasting," arXiv.org, 02 2024. [Online]. Available: https://arxiv.org/abs/2310.08278

[21] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," *arXiv:2302.13971 [cs]*, 02 2023. [Online]. Available: https://arxiv.org/abs/2302.13971

[22] "National accounts," www.censtatd.gov.hk. [Online]. Available: https://www.censtatd.gov.hk/en/scode250.html

[23] ONS, "Gross domestic product (gdp) - office for national statistics," Ons.gov.uk, 2023. [Online]. Available: https://www.ons.gov.uk/economy/grossdomesticproductgdp

[24] "Centadata — centaline property," hk.centanet.com. [Online]. Available: https://hk.centanet.com/findproperty/en/centadata

[25] H. L. Registry, "Uk house price index: reports," GOV.UK, 06 2016. [Online]. Available: https://www.gov.uk/government/collections/uk-house-price-index-reports

[26] "Rating and valuation department - hong kong property review 2019," Rvd.gov.hk, 2019. [Online]. Available: https://www.rvd.gov.hk/en/publications/hkpr.html

[27] "Hong kong monthly," Knightfrank.com.hk, 2024. [Online]. Available: https://www.knightfrank.com.hk/research/hong-kong-monthly-june-2024-11302.aspx

[28] "Hong kong property market monitor," www.jll.com.hk. [Online]. Available: https://www.jll.com.hk/en/campaign/hong-kong-property-market-monitor

[29] H. England, "Housing statistics," GOV.UK, 06 2014. [Online]. Available: https://www.gov.uk/government/collections/housing-statistics

[30] “Housing insight reports,” www.propertymark.co.uk. [Online]. Available: https://www.propertymark.co.uk/news-reports/housing-insight-report.html

[31] “House price crash forum,” House Price Crash Forum, 07 2011. [Online]. Available: https://www.housepricecrash.co.uk/forum/index.php?/forum/22-house-prices-and-the-economy/

[32] K. Lane, “Tokenization with nltk,” Medium, 10 2021. [Online]. Available: https://medium.com/@kelsklane/tokenization-with-nltk-52cd7b88c7d

[33] A. Jain, “All about tokenization, stop words, stemming and lemmatization in nlp,” Medium, 02 2024. [Online]. Available: https://medium.com/@abhishekjainindore24/all-about-tokenization-stop-words-stemming-and-lemmatization-in-nlp-1620ffaf0f87#:~:text=Definition%3A

[34] “Sarimax: Introduction - statsmodels 0.14.1,” www.statsmodels.org. [Online]. Available: https://www.statsmodels.org/stable/examples/notebooks/generated/statespace_sarimax_stata.html

[35] B. Artley, “Time series forecasting with arima , sarima and sarimax,” Medium, 05 2022. [Online]. Available: https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6

[36] *The Analysis of Time Series An Introduction, Sixth Edition*. Chapman And Hall/Crc, 2016.

[37] ListenToTheUniverse, “Understanding lstm: An in-depth look at its architecture, functioning, and pros cons,” Medium, 07 2023. [Online]. Available: https://medium.com/@ebinbabuthomas_21082/understanding-lstm-an-in-depth-look-at-its-architecture-functioning-and-pros-cons-5424dd72 ~:text=The

[38] R. Horev, “Bert explained: State of the art language model for nlp,” Medium, 11 2018. [Online]. Available: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

Appendix

A Grid search result of HK SARIMAX Quantitative model

```
Month 2015-01: Order=(2, 1, 1), Trend=t, MAE=0.8711, MSE=0.9166, RMSE=0.9574
Month 2015-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2015-03: Order=(0, 1, 0), Trend=n, MAE=0.5460, MSE=0.5962, RMSE=0.7722
Month 2015-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2015-05: Order=(1, 1, 2), Trend=n, MAE=0.5578, MSE=0.4862, RMSE=0.6972
Month 2015-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2015-07: Order=(1, 1, 1), Trend=n, MAE=0.4806, MSE=0.2380, RMSE=0.4878
Month 2015-08: Order=(1, 1, 1), Trend=n, MAE=0.3056, MSE=0.1496, RMSE=0.3868
Month 2015-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2015-10: Order=(1, 0, 2), Trend=n, MAE=0.6125, MSE=0.7248, RMSE=0.8513
Month 2015-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2015-12: Order=(1, 0, 0), Trend=t, MAE=0.3339, MSE=0.1248, RMSE=0.3532
Month 2016-01: Order=(0, 0, 1), Trend=c, MAE=0.2593, MSE=0.1338, RMSE=0.3658
Month 2016-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2016-03: Order=(1, 0, 0), Trend=n, MAE=0.2678, MSE=0.0751, RMSE=0.2741
Month 2016-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2016-05: Order=(2, 0, 1), Trend=t, MAE=0.2552, MSE=0.1230, RMSE=0.3507
Month 2016-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2016-07: Order=(0, 0, 1), Trend=n, MAE=0.9405, MSE=1.0959, RMSE=1.0468
Month 2016-08: Order=(0, 0, 2), Trend=ct, MAE=0.9060, MSE=1.5562, RMSE=1.2475
Month 2016-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2016-10: Order=(1, 0, 0), Trend=n, MAE=0.3909, MSE=0.2232, RMSE=0.4724
Month 2016-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2016-12: Order=(2, 0, 0), Trend=t, MAE=0.1690, MSE=0.0456, RMSE=0.2135
Month 2017-01: Order=(1, 0, 2), Trend=ct, MAE=0.2894, MSE=0.1642, RMSE=0.4053
Month 2017-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2017-03: Order=(0, 1, 0), Trend=ct, MAE=0.7706, MSE=1.1462, RMSE=1.0706
Month 2017-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2017-05: Order=(1, 0, 0), Trend=ct, MAE=0.0093, MSE=0.0001, RMSE=0.0094
Month 2017-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2017-07: Order=(0, 1, 0), Trend=n, MAE=0.0039, MSE=0.0000, RMSE=0.0055
Month 2017-08: Order=(2, 0, 0), Trend=c, MAE=0.1417, MSE=0.0203, RMSE=0.1426
Month 2017-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2017-10: Order=(2, 0, 1), Trend=t, MAE=0.4489, MSE=0.3357, RMSE=0.5794
Month 2017-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2017-12: Order=(2, 1, 2), Trend=c, MAE=0.6449, MSE=0.7374, RMSE=0.8587
Month 2018-01: Order=(0, 1, 0), Trend=t, MAE=0.8666, MSE=1.2920, RMSE=1.1367
Month 2018-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-03: Order=(1, 1, 0), Trend=t, MAE=1.6638, MSE=5.2947, RMSE=2.3010
Month 2018-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-05: Order=(2, 1, 1), Trend=ct, MAE=0.4730, MSE=0.2324, RMSE=0.4821
Month 2018-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-07: Order=(0, 1, 0), Trend=n, MAE=0.1600, MSE=0.0512, RMSE=0.2263
Month 2018-08: Order=(0, 1, 2), Trend=n, MAE=0.7423, MSE=0.5897, RMSE=0.7680
Month 2018-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-10: Order=(0, 0, 2), Trend=t, MAE=0.7687, MSE=0.6313, RMSE=0.7946
Month 2018-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-12: Order=(1, 0, 1), Trend=c, MAE=0.9111, MSE=0.9150, RMSE=0.9566
Month 2019-01: Order=(0, 1, 2), Trend=t, MAE=0.3127, MSE=0.1302, RMSE=0.3608
Month 2019-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-03: Order=(2, 0, 0), Trend=t, MAE=1.7553, MSE=5.5368, RMSE=2.3530
Month 2019-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-05: Order=(0, 1, 0), Trend=n, MAE=0.2513, MSE=0.1263, RMSE=0.3554
Month 2019-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-07: Order=(1, 0, 0), Trend=ct, MAE=0.1276, MSE=0.0241, RMSE=0.1552
Month 2019-08: Order=(0, 0, 2), Trend=n, MAE=0.5761, MSE=0.3996, RMSE=0.6322
Month 2019-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-10: Order=(0, 1, 1), Trend=t, MAE=0.2748, MSE=0.0927, RMSE=0.3045
Month 2019-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-12: Order=(1, 0, 1), Trend=ct, MAE=0.2306, MSE=0.0882, RMSE=0.2969
Month 2020-01: Order=(2, 0, 1), Trend=t, MAE=0.2923, MSE=0.1222, RMSE=0.3495
Month 2020-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2020-03: Order=(1, 0, 2), Trend=n, MAE=0.1381, MSE=0.0257, RMSE=0.1603
Month 2020-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2020-05: Order=(1, 0, 0), Trend=n, MAE=0.5620, MSE=0.4083, RMSE=0.6390
Month 2020-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2020-07: Order=(1, 0, 1), Trend=t, MAE=0.3686, MSE=0.1369, RMSE=0.3700
Month 2020-08: Order=(1, 0, 1), Trend=ct, MAE=0.6473, MSE=0.4193, RMSE=0.6476
Month 2020-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2020-10: Order=(1, 1, 2), Trend=n, MAE=0.0372, MSE=0.0024, RMSE=0.0490
Month 2020-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
```

```
Month 2020-12: Order=(1, 0, 2), Trend=c, MAE=0.2553, MSE=0.0814, RMSE=0.2854
Month 2021-01: Order=(2, 1, 2), Trend=c, MAE=0.0529, MSE=0.0038, RMSE=0.0618
Month 2021-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-03: Order=(2, 0, 0), Trend=c, MAE=0.3117, MSE=0.1939, RMSE=0.4404
Month 2021-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-05: Order=(2, 0, 1), Trend=ct, MAE=0.4230, MSE=0.1793, RMSE=0.4234
Month 2021-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-07: Order=(0, 1, 2), Trend=t, MAE=0.2666, MSE=0.0825, RMSE=0.2872
Month 2021-08: Order=(1, 0, 2), Trend=c, MAE=0.4282, MSE=0.1943, RMSE=0.4408
Month 2021-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-10: Order=(1, 0, 0), Trend=c, MAE=0.1883, MSE=0.0656, RMSE=0.2562
Month 2021-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-12: Order=(2, 1, 1), Trend=t, MAE=0.0901, MSE=0.0104, RMSE=0.1022
Month 2022-01: Order=(2, 0, 2), Trend=c, MAE=1.0193, MSE=1.0705, RMSE=1.0346
Month 2022-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-03: Order=(0, 1, 1), Trend=c, MAE=0.0525, MSE=0.0029, RMSE=0.0537
Month 2022-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-05: Order=(2, 0, 1), Trend=n, MAE=0.1185, MSE=0.0142, RMSE=0.1193
Month 2022-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-07: Order=(1, 0, 2), Trend=t, MAE=1.9317, MSE=6.5596, RMSE=2.5612
Month 2022-08: Order=(1, 0, 2), Trend=n, MAE=1.2654, MSE=1.8325, RMSE=1.3537
Month 2022-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-10: Order=(2, 0, 2), Trend=n, MAE=0.8238, MSE=0.7130, RMSE=0.8444
Month 2022-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-12: Order=(1, 0, 1), Trend=ct, MAE=0.4369, MSE=0.3709, RMSE=0.6090
Month 2023-01: Order=(2, 1, 2), Trend=c, MAE=0.9147, MSE=1.4236, RMSE=1.1931
Month 2023-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2023-03: Order=(0, 1, 0), Trend=n, MAE=0.1138, MSE=0.0259, RMSE=0.1609
Month 2023-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2023-05: Order=(0, 1, 0), Trend=n, MAE=0.1013, MSE=0.0205, RMSE=0.1432
Month 2023-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2023-07: Order=(1, 1, 0), Trend=ct, MAE=0.6454, MSE=0.7256, RMSE=0.8518
Month 2023-08: Order=(2, 0, 1), Trend=c, MAE=1.0647, MSE=1.5734, RMSE=1.2543
Month 2023-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2023-10: Order=(2, 0, 1), Trend=n, MAE=0.8633, MSE=1.3593, RMSE=1.1659
Month 2023-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2023-12: Order=(0, 1, 0), Trend=ct, MAE=0.5810, MSE=0.3815, RMSE=0.6177
Month 2024-01: Order=(0, 1, 2), Trend=t, MAE=0.4594, MSE=0.3593, RMSE=0.5994
Month 2024-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
```

B Grid search result of UK SARIMAX Quantitative model

```
Month 2018-01: Order=(1, 1, 2), Trend=t, MAE=0.0901, MSE=0.0087, RMSE=0.0933
Month 2018-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-03: Order=(0, 0, 2), Trend=c, MAE=0.2624, MSE=0.0937, RMSE=0.3061
Month 2018-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-05: Order=(0, 1, 1), Trend=c, MAE=0.1767, MSE=0.0555, RMSE=0.2355
Month 2018-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-07: Order=(1, 0, 0), Trend=c, MAE=0.1829, MSE=0.0436, RMSE=0.2088
Month 2018-08: Order=(2, 0, 2), Trend=c, MAE=0.2605, MSE=0.0696, RMSE=0.2639
Month 2018-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-10: Order=(2, 0, 1), Trend=n, MAE=0.0954, MSE=0.0110, RMSE=0.1051
Month 2018-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2018-12: Order=(2, 0, 2), Trend=n, MAE=0.1783, MSE=0.0450, RMSE=0.2122
Month 2019-01: Order=(1, 1, 2), Trend=t, MAE=0.0945, MSE=0.0101, RMSE=0.1007
Month 2019-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-03: Order=(0, 1, 0), Trend=c, MAE=0.2869, MSE=0.0823, RMSE=0.2869
Month 2019-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-05: Order=(1, 0, 0), Trend=c, MAE=0.1013, MSE=0.0104, RMSE=0.1019
Month 2019-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-07: Order=(0, 1, 0), Trend=n, MAE=0.0986, MSE=0.0194, RMSE=0.1394
Month 2019-08: Order=(0, 1, 1), Trend=n, MAE=0.0084, MSE=0.0001, RMSE=0.0110
Month 2019-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-10: Order=(2, 0, 2), Trend=ct, MAE=0.1102, MSE=0.0122, RMSE=0.1102
Month 2019-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2019-12: Order=(0, 1, 2), Trend=n, MAE=0.0070, MSE=0.0001, RMSE=0.0088
Month 2020-01: Order=(2, 0, 0), Trend=t, MAE=0.2096, MSE=0.0662, RMSE=0.2573
Month 2020-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2020-03: Order=(2, 0, 1), Trend=ct, MAE=0.4416, MSE=0.1998, RMSE=0.4470
Month 2020-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2020-05: Order=(2, 1, 1), Trend=c, MAE=0.1624, MSE=0.0272, RMSE=0.1650
Month 2020-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2020-07: Order=(2, 0, 0), Trend=c, MAE=0.1935, MSE=0.0472, RMSE=0.2172
Month 2020-08: Order=(1, 0, 1), Trend=ct, MAE=0.7775, MSE=0.6064, RMSE=0.7787
Month 2020-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2020-10: Order=(2, 0, 2), Trend=c, MAE=0.2087, MSE=0.0714, RMSE=0.2672
Month 2020-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2020-12: Order=(0, 1, 0), Trend=ct, MAE=0.0976, MSE=0.0144, RMSE=0.1199
Month 2021-01: Order=(2, 0, 0), Trend=t, MAE=0.0745, MSE=0.0109, RMSE=0.1044
Month 2021-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-03: Order=(0, 0, 2), Trend=n, MAE=0.3572, MSE=0.1314, RMSE=0.3625
Month 2021-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-05: Order=(1, 0, 2), Trend=t, MAE=2.9435, MSE=12.8029, RMSE=3.5781
Month 2021-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-07: Order=(2, 1, 1), Trend=c, MAE=0.9671, MSE=1.5023, RMSE=1.2257
Month 2021-08: Order=(2, 0, 2), Trend=t, MAE=1.3616, MSE=2.3794, RMSE=1.5425
Month 2021-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-10: Order=(2, 0, 2), Trend=c, MAE=0.4767, MSE=0.3597, RMSE=0.5997
Month 2021-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2021-12: Order=(2, 1, 1), Trend=t, MAE=0.2992, MSE=0.0974, RMSE=0.3121
Month 2022-01: Order=(2, 0, 2), Trend=c, MAE=0.0062, MSE=0.0000, RMSE=0.0063
Month 2022-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-03: Order=(1, 0, 0), Trend=t, MAE=0.4148, MSE=0.2707, RMSE=0.5203
Month 2022-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-05: Order=(1, 0, 2), Trend=t, MAE=0.1546, MSE=0.0285, RMSE=0.1687
Month 2022-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-07: Order=(0, 1, 1), Trend=t, MAE=0.0571, MSE=0.0042, RMSE=0.0645
Month 2022-08: Order=(0, 1, 0), Trend=n, MAE=0.0233, MSE=0.0011, RMSE=0.0329
Month 2022-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-10: Order=(2, 1, 2), Trend=c, MAE=0.1112, MSE=0.0140, RMSE=0.1184
Month 2022-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2022-12: Order=(0, 1, 0), Trend=n, MAE=0.8003, MSE=1.2809, RMSE=1.1318
Month 2023-01: Order=(1, 1, 2), Trend=n, MAE=0.3657, MSE=0.2088, RMSE=0.4569
Month 2023-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2023-03: Order=(0, 1, 0), Trend=n, MAE=0.1503, MSE=0.0452, RMSE=0.2125
Month 2023-04: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2023-05: Order=(2, 1, 2), Trend=n, MAE=0.3115, MSE=0.1676, RMSE=0.4093
Month 2023-06: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2023-07: Order=(2, 1, 1), Trend=ct, MAE=0.2523, MSE=0.1112, RMSE=0.3335
Month 2023-08: Order=(2, 1, 2), Trend=n, MAE=0.3470, MSE=0.1527, RMSE=0.3908
Month 2023-09: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
Month 2023-10: Order=(0, 1, 2), Trend=ct, MAE=0.1234, MSE=0.0152, RMSE=0.1234
Month 2023-11: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
```

```
Month 2023-12: Order=(0, 1, 0), Trend=n, MAE=0.1526, MSE=0.0466, RMSE=0.2158
Month 2024-01: Order=(2, 1, 2), Trend=t, MAE=0.0905, MSE=0.0088, RMSE=0.0937
Month 2024-02: Order=(0, 1, 0), Trend=n, MAE=0.0000, MSE=0.0000, RMSE=0.0000
```

C Grid search result of HK LSTM SARIMAX hybrid model

```
Month 2015-01: Order=(0, 1, 2), Trend=t, MAE=0.1015, MSE=0.0103, RMSE=0.1015
Month 2015-02: Order=(0, 1, 1), Trend=t, MAE=0.5413, MSE=0.2930, RMSE=0.5413
Month 2015-03: Order=(0, 0, 2), Trend=t, MAE=0.1073, MSE=0.0115, RMSE=0.1073
Month 2015-04: Order=(1, 1, 2), Trend=n, MAE=0.0072, MSE=0.0001, RMSE=0.0072
Month 2015-05: Order=(0, 1, 2), Trend=t, MAE=0.0470, MSE=0.0022, RMSE=0.0470
Month 2015-06: Order=(2, 0, 0), Trend=n, MAE=0.1529, MSE=0.0234, RMSE=0.1529
Month 2015-07: Order=(0, 1, 2), Trend=n, MAE=0.1177, MSE=0.0139, RMSE=0.1177
Month 2015-08: Order=(0, 1, 1), Trend=n, MAE=0.0337, MSE=0.0011, RMSE=0.0337
Month 2015-09: Order=(2, 0, 2), Trend=n, MAE=0.4166, MSE=0.1735, RMSE=0.4166
Month 2015-10: Order=(1, 1, 1), Trend=t, MAE=0.0235, MSE=0.0006, RMSE=0.0235
Month 2015-11: Order=(1, 1, 0), Trend=t, MAE=0.0511, MSE=0.0026, RMSE=0.0511
Month 2015-12: Order=(2, 1, 1), Trend=t, MAE=0.0230, MSE=0.0005, RMSE=0.0230
Month 2016-01: Order=(2, 0, 2), Trend=ct, MAE=0.0078, MSE=0.0001, RMSE=0.0078
Month 2016-02: Order=(2, 1, 1), Trend=t, MAE=0.0668, MSE=0.0045, RMSE=0.0668
Month 2016-03: Order=(2, 0, 0), Trend=ct, MAE=0.0150, MSE=0.0002, RMSE=0.0150
Month 2016-04: Order=(1, 1, 0), Trend=ct, MAE=0.0023, MSE=0.0000, RMSE=0.0023
Month 2016-05: Order=(1, 0, 0), Trend=ct, MAE=0.0559, MSE=0.0031, RMSE=0.0559
Month 2016-06: Order=(1, 1, 0), Trend=t, MAE=0.0299, MSE=0.0009, RMSE=0.0299
Month 2016-07: Order=(0, 0, 0), Trend=ct, MAE=0.7802, MSE=0.6087, RMSE=0.7802
Month 2016-08: Order=(2, 0, 2), Trend=ct, MAE=0.0628, MSE=0.0039, RMSE=0.0628
Month 2016-09: Order=(0, 1, 1), Trend=t, MAE=0.0031, MSE=0.0000, RMSE=0.0031
Month 2016-10: Order=(2, 0, 0), Trend=ct, MAE=0.0583, MSE=0.0034, RMSE=0.0583
Month 2016-11: Order=(2, 0, 0), Trend=c, MAE=0.0086, MSE=0.0001, RMSE=0.0086
Month 2016-12: Order=(1, 1, 2), Trend=n, MAE=0.0269, MSE=0.0007, RMSE=0.0269
Month 2017-01: Order=(1, 0, 1), Trend=t, MAE=0.0746, MSE=0.0056, RMSE=0.0746
Month 2017-02: Order=(2, 0, 1), Trend=n, MAE=0.0873, MSE=0.0076, RMSE=0.0873
Month 2017-03: Order=(2, 1, 0), Trend=ct, MAE=0.0116, MSE=0.0001, RMSE=0.0116
Month 2017-04: Order=(0, 0, 1), Trend=n, MAE=0.1055, MSE=0.0111, RMSE=0.1055
Month 2017-05: Order=(0, 0, 1), Trend=t, MAE=0.0981, MSE=0.0096, RMSE=0.0981
Month 2017-06: Order=(1, 0, 2), Trend=c, MAE=0.0041, MSE=0.0000, RMSE=0.0041
Month 2017-07: Order=(1, 1, 0), Trend=n, MAE=0.0331, MSE=0.0011, RMSE=0.0331
Month 2017-08: Order=(2, 1, 2), Trend=n, MAE=0.0080, MSE=0.0001, RMSE=0.0080
Month 2017-09: Order=(2, 0, 0), Trend=ct, MAE=0.0648, MSE=0.0042, RMSE=0.0648
Month 2017-10: Order=(0, 0, 0), Trend=ct, MAE=0.2658, MSE=0.0707, RMSE=0.2658
Month 2017-11: Order=(0, 1, 2), Trend=t, MAE=0.0079, MSE=0.0001, RMSE=0.0079
Month 2017-12: Order=(1, 1, 1), Trend=t, MAE=0.1268, MSE=0.0161, RMSE=0.1268
Month 2018-01: Order=(1, 0, 1), Trend=n, MAE=0.0201, MSE=0.0004, RMSE=0.0201
Month 2018-02: Order=(0, 0, 0), Trend=ct, MAE=0.0273, MSE=0.0007, RMSE=0.0273
Month 2018-03: Order=(2, 0, 0), Trend=t, MAE=0.3388, MSE=0.1148, RMSE=0.3388
Month 2018-04: Order=(0, 1, 0), Trend=ct, MAE=0.0484, MSE=0.0023, RMSE=0.0484
Month 2018-05: Order=(1, 1, 1), Trend=ct, MAE=0.0349, MSE=0.0012, RMSE=0.0349
Month 2018-06: Order=(2, 1, 2), Trend=ct, MAE=0.0564, MSE=0.0032, RMSE=0.0564
Month 2018-07: Order=(0, 1, 1), Trend=n, MAE=0.0599, MSE=0.0036, RMSE=0.0599
Month 2018-08: Order=(2, 0, 0), Trend=t, MAE=0.0158, MSE=0.0003, RMSE=0.0158
Month 2018-09: Order=(0, 0, 1), Trend=c, MAE=0.0058, MSE=0.0000, RMSE=0.0058
Month 2018-10: Order=(2, 0, 1), Trend=t, MAE=0.0216, MSE=0.0005, RMSE=0.0216
Month 2018-11: Order=(0, 1, 1), Trend=n, MAE=0.0639, MSE=0.0041, RMSE=0.0639
Month 2018-12: Order=(2, 1, 1), Trend=n, MAE=0.0558, MSE=0.0031, RMSE=0.0558
Month 2019-01: Order=(1, 0, 0), Trend=n, MAE=0.0011, MSE=0.0000, RMSE=0.0011
Month 2019-02: Order=(1, 0, 1), Trend=t, MAE=0.3917, MSE=0.1534, RMSE=0.3917
Month 2019-03: Order=(2, 1, 1), Trend=t, MAE=0.0308, MSE=0.0009, RMSE=0.0308
Month 2019-04: Order=(1, 1, 2), Trend=t, MAE=0.0275, MSE=0.0008, RMSE=0.0275
Month 2019-05: Order=(0, 1, 0), Trend=n, MAE=0.1392, MSE=0.0194, RMSE=0.1392
Month 2019-06: Order=(0, 0, 2), Trend=n, MAE=0.0893, MSE=0.0080, RMSE=0.0893
Month 2019-07: Order=(0, 0, 0), Trend=n, MAE=0.0401, MSE=0.0016, RMSE=0.0401
Month 2019-08: Order=(2, 0, 1), Trend=t, MAE=0.3544, MSE=0.1256, RMSE=0.3544
Month 2019-09: Order=(2, 0, 1), Trend=n, MAE=0.0001, MSE=0.0000, RMSE=0.0001
Month 2019-10: Order=(2, 0, 2), Trend=ct, MAE=0.0494, MSE=0.0024, RMSE=0.0494
Month 2019-11: Order=(2, 1, 2), Trend=n, MAE=1.9554, MSE=3.8237, RMSE=1.9554
Month 2019-12: Order=(2, 0, 0), Trend=c, MAE=0.2073, MSE=0.0430, RMSE=0.2073
Month 2020-01: Order=(1, 0, 2), Trend=ct, MAE=0.0105, MSE=0.0001, RMSE=0.0105
Month 2020-02: Order=(0, 1, 1), Trend=t, MAE=0.0402, MSE=0.0016, RMSE=0.0402
Month 2020-03: Order=(0, 1, 1), Trend=ct, MAE=0.0177, MSE=0.0003, RMSE=0.0177
Month 2020-04: Order=(2, 0, 2), Trend=ct, MAE=0.1779, MSE=0.0316, RMSE=0.1779
Month 2020-05: Order=(1, 1, 0), Trend=ct, MAE=0.0146, MSE=0.0002, RMSE=0.0146
Month 2020-06: Order=(2, 1, 0), Trend=t, MAE=0.0731, MSE=0.0053, RMSE=0.0731
Month 2020-07: Order=(0, 0, 0), Trend=ct, MAE=1.3485, MSE=1.8185, RMSE=1.3485
Month 2020-08: Order=(1, 0, 1), Trend=c, MAE=0.1931, MSE=0.0373, RMSE=0.1931
Month 2020-09: Order=(1, 0, 1), Trend=n, MAE=0.0826, MSE=0.0068, RMSE=0.0826
Month 2020-10: Order=(1, 1, 2), Trend=n, MAE=0.0011, MSE=0.0000, RMSE=0.0011
Month 2020-11: Order=(0, 1, 1), Trend=ct, MAE=0.0086, MSE=0.0001, RMSE=0.0086
```

```
Month 2020-12: Order=(2, 0, 0), Trend=c, MAE=0.0388, MSE=0.0015, RMSE=0.0388
Month 2021-01: Order=(0, 1, 2), Trend=c, MAE=0.0112, MSE=0.0001, RMSE=0.0112
Month 2021-02: Order=(0, 0, 0), Trend=c, MAE=0.5030, MSE=0.2530, RMSE=0.5030
Month 2021-03: Order=(0, 1, 0), Trend=t, MAE=0.0622, MSE=0.0039, RMSE=0.0622
Month 2021-04: Order=(2, 1, 2), Trend=t, MAE=0.9192, MSE=0.8448, RMSE=0.9192
Month 2021-05: Order=(0, 1, 1), Trend=c, MAE=0.0515, MSE=0.0027, RMSE=0.0515
Month 2021-06: Order=(1, 0, 0), Trend=c, MAE=0.0854, MSE=0.0073, RMSE=0.0854
Month 2021-07: Order=(1, 0, 2), Trend=t, MAE=0.0667, MSE=0.0045, RMSE=0.0667
Month 2021-08: Order=(0, 1, 0), Trend=n, MAE=0.2443, MSE=0.0597, RMSE=0.2443
Month 2021-09: Order=(1, 0, 1), Trend=ct, MAE=1.5078, MSE=2.2734, RMSE=1.5078
Month 2021-10: Order=(1, 0, 1), Trend=ct, MAE=0.0836, MSE=0.0070, RMSE=0.0836
Month 2021-11: Order=(2, 1, 2), Trend=ct, MAE=0.1225, MSE=0.0150, RMSE=0.1225
Month 2021-12: Order=(0, 1, 0), Trend=c, MAE=0.0090, MSE=0.0001, RMSE=0.0090
Month 2022-01: Order=(2, 0, 2), Trend=c, MAE=0.9151, MSE=0.8374, RMSE=0.9151
Month 2022-02: Order=(2, 1, 1), Trend=ct, MAE=0.0123, MSE=0.0002, RMSE=0.0123
Month 2022-03: Order=(0, 1, 0), Trend=t, MAE=0.1165, MSE=0.0136, RMSE=0.1165
Month 2022-04: Order=(1, 0, 1), Trend=ct, MAE=0.1009, MSE=0.0102, RMSE=0.1009
Month 2022-05: Order=(0, 0, 2), Trend=c, MAE=0.0220, MSE=0.0005, RMSE=0.0220
Month 2022-06: Order=(1, 0, 1), Trend=t, MAE=0.0165, MSE=0.0003, RMSE=0.0165
Month 2022-07: Order=(2, 0, 2), Trend=n, MAE=0.8167, MSE=0.6670, RMSE=0.8167
Month 2022-08: Order=(1, 0, 2), Trend=t, MAE=0.0905, MSE=0.0082, RMSE=0.0905
Month 2022-09: Order=(1, 1, 2), Trend=ct, MAE=0.0004, MSE=0.0000, RMSE=0.0004
Month 2022-10: Order=(2, 0, 1), Trend=n, MAE=0.6443, MSE=0.4151, RMSE=0.6443
Month 2022-11: Order=(1, 0, 1), Trend=c, MAE=0.0701, MSE=0.0049, RMSE=0.0701
Month 2022-12: Order=(2, 0, 2), Trend=n, MAE=0.1598, MSE=0.0256, RMSE=0.1598
Month 2023-01: Order=(2, 0, 2), Trend=t, MAE=0.0871, MSE=0.0076, RMSE=0.0871
Month 2023-02: Order=(2, 1, 2), Trend=n, MAE=0.0083, MSE=0.0001, RMSE=0.0083
Month 2023-03: Order=(2, 0, 2), Trend=t, MAE=0.0297, MSE=0.0009, RMSE=0.0297
Month 2023-04: Order=(1, 0, 2), Trend=t, MAE=0.0061, MSE=0.0000, RMSE=0.0061
Month 2023-05: Order=(2, 0, 0), Trend=c, MAE=0.0988, MSE=0.0098, RMSE=0.0988
Month 2023-06: Order=(2, 0, 1), Trend=c, MAE=0.0013, MSE=0.0000, RMSE=0.0013
Month 2023-07: Order=(2, 0, 2), Trend=t, MAE=0.0062, MSE=0.0000, RMSE=0.0062
Month 2023-08: Order=(0, 0, 0), Trend=ct, MAE=1.4204, MSE=2.0174, RMSE=1.4204
Month 2023-09: Order=(2, 0, 1), Trend=c, MAE=0.4168, MSE=0.1737, RMSE=0.4168
Month 2023-10: Order=(1, 0, 0), Trend=t, MAE=0.0288, MSE=0.0008, RMSE=0.0288
Month 2023-11: Order=(2, 1, 1), Trend=ct, MAE=0.0102, MSE=0.0001, RMSE=0.0102
Month 2023-12: Order=(2, 0, 0), Trend=c, MAE=0.0014, MSE=0.0000, RMSE=0.0014
Month 2024-01: Order=(1, 1, 1), Trend=t, MAE=0.0007, MSE=0.0000, RMSE=0.0007
Month 2024-02: Order=(2, 0, 2), Trend=n, MAE=0.0084, MSE=0.0001, RMSE=0.0084
```

D Grid search result of UK LSTM SARIMAX hybrid model

```
Month 2015-01: Order=(0, 1, 0), Trend=t, MAE=0.0538, MSE=0.0029, RMSE=0.0538
Month 2015-02: Order=(1, 1, 2), Trend=t, MAE=0.0738, MSE=0.0054, RMSE=0.0738
Month 2015-03: Order=(1, 1, 1), Trend=t, MAE=0.5356, MSE=0.2869, RMSE=0.5356
Month 2015-04: Order=(0, 0, 1), Trend=t, MAE=0.0397, MSE=0.0016, RMSE=0.0397
Month 2015-05: Order=(1, 1, 2), Trend=n, MAE=0.0099, MSE=0.0001, RMSE=0.0099
Month 2015-06: Order=(0, 1, 2), Trend=t, MAE=0.0470, MSE=0.0022, RMSE=0.0470
Month 2015-07: Order=(2, 0, 1), Trend=n, MAE=0.1287, MSE=0.0166, RMSE=0.1287
Month 2015-08: Order=(0, 1, 2), Trend=n, MAE=0.0781, MSE=0.0061, RMSE=0.0781
Month 2015-09: Order=(0, 1, 1), Trend=n, MAE=0.0337, MSE=0.0011, RMSE=0.0337
Month 2015-10: Order=(0, 0, 2), Trend=n, MAE=0.5290, MSE=0.2798, RMSE=0.5290
Month 2015-11: Order=(2, 0, 0), Trend=t, MAE=0.0355, MSE=0.0013, RMSE=0.0355
Month 2015-12: Order=(1, 1, 0), Trend=t, MAE=0.0511, MSE=0.0026, RMSE=0.0511
Month 2016-01: Order=(2, 1, 0), Trend=n, MAE=0.2585, MSE=0.0668, RMSE=0.2585
Month 2016-02: Order=(1, 1, 0), Trend=t, MAE=0.0481, MSE=0.0023, RMSE=0.0481
Month 2016-03: Order=(2, 0, 0), Trend=t, MAE=0.0527, MSE=0.0028, RMSE=0.0527
Month 2016-04: Order=(0, 0, 2), Trend=c, MAE=0.1513, MSE=0.0229, RMSE=0.1513
Month 2016-05: Order=(2, 1, 1), Trend=ct, MAE=0.0053, MSE=0.0000, RMSE=0.0053
Month 2016-06: Order=(2, 0, 0), Trend=c, MAE=0.3454, MSE=0.1193, RMSE=0.3454
Month 2016-07: Order=(1, 1, 0), Trend=c, MAE=0.0620, MSE=0.0039, RMSE=0.0620
Month 2016-08: Order=(0, 0, 2), Trend=ct, MAE=0.3668, MSE=0.1346, RMSE=0.3668
Month 2016-09: Order=(0, 1, 1), Trend=c, MAE=0.0038, MSE=0.0000, RMSE=0.0038
Month 2016-10: Order=(2, 1, 1), Trend=ct, MAE=0.0449, MSE=0.0020, RMSE=0.0449
Month 2016-11: Order=(1, 0, 2), Trend=c, MAE=0.0300, MSE=0.0009, RMSE=0.0300
Month 2016-12: Order=(1, 0, 1), Trend=t, MAE=0.1294, MSE=0.0167, RMSE=0.1294
Month 2017-01: Order=(1, 0, 0), Trend=n, MAE=0.0058, MSE=0.0000, RMSE=0.0058
Month 2017-02: Order=(1, 0, 2), Trend=t, MAE=0.3507, MSE=0.1230, RMSE=0.3507
Month 2017-03: Order=(0, 0, 1), Trend=ct, MAE=0.0169, MSE=0.0003, RMSE=0.0169
Month 2017-04: Order=(2, 1, 0), Trend=ct, MAE=0.0037, MSE=0.0000, RMSE=0.0037
Month 2017-05: Order=(2, 0, 0), Trend=ct, MAE=0.0268, MSE=0.0007, RMSE=0.0268
Month 2017-06: Order=(1, 0, 2), Trend=ct, MAE=0.0070, MSE=0.0000, RMSE=0.0070
Month 2017-07: Order=(0, 1, 0), Trend=n, MAE=0.0815, MSE=0.0066, RMSE=0.0815
Month 2017-08: Order=(2, 1, 1), Trend=t, MAE=0.0100, MSE=0.0001, RMSE=0.0100
Month 2017-09: Order=(2, 1, 2), Trend=n, MAE=0.0046, MSE=0.0000, RMSE=0.0046
Month 2017-10: Order=(0, 0, 2), Trend=ct, MAE=0.1271, MSE=0.0162, RMSE=0.1271
Month 2017-11: Order=(0, 0, 0), Trend=c, MAE=0.0181, MSE=0.0003, RMSE=0.0181
Month 2017-12: Order=(2, 0, 0), Trend=ct, MAE=0.0002, MSE=0.0000, RMSE=0.0002
Month 2018-01: Order=(1, 1, 2), Trend=c, MAE=0.0122, MSE=0.0001, RMSE=0.0122
Month 2018-02: Order=(0, 0, 1), Trend=ct, MAE=0.2797, MSE=0.0782, RMSE=0.2797
Month 2018-03: Order=(1, 1, 2), Trend=ct, MAE=0.0371, MSE=0.0014, RMSE=0.0371
Month 2018-04: Order=(1, 0, 2), Trend=t, MAE=0.0398, MSE=0.0016, RMSE=0.0398
Month 2018-05: Order=(1, 0, 1), Trend=t, MAE=0.1274, MSE=0.0162, RMSE=0.1274
Month 2018-06: Order=(1, 1, 1), Trend=ct, MAE=0.0011, MSE=0.0000, RMSE=0.0011
Month 2018-07: Order=(1, 1, 1), Trend=n, MAE=0.0022, MSE=0.0000, RMSE=0.0022
Month 2018-08: Order=(0, 1, 1), Trend=n, MAE=0.0874, MSE=0.0076, RMSE=0.0874
Month 2018-09: Order=(1, 0, 2), Trend=ct, MAE=0.0541, MSE=0.0029, RMSE=0.0541
Month 2018-10: Order=(0, 0, 1), Trend=c, MAE=0.0179, MSE=0.0003, RMSE=0.0179
Month 2018-11: Order=(0, 0, 2), Trend=t, MAE=0.6395, MSE=0.4090, RMSE=0.6395
Month 2018-12: Order=(0, 1, 1), Trend=n, MAE=0.0017, MSE=0.0000, RMSE=0.0017
Month 2019-01: Order=(2, 1, 0), Trend=ct, MAE=0.0046, MSE=0.0000, RMSE=0.0046
Month 2019-02: Order=(0, 1, 0), Trend=c, MAE=0.0124, MSE=0.0002, RMSE=0.0124
Month 2019-03: Order=(2, 0, 0), Trend=t, MAE=0.1635, MSE=0.0267, RMSE=0.1635
Month 2019-04: Order=(2, 1, 2), Trend=n, MAE=0.0186, MSE=0.0003, RMSE=0.0186
Month 2019-05: Order=(2, 1, 1), Trend=c, MAE=0.0131, MSE=0.0002, RMSE=0.0131
Month 2019-06: Order=(2, 0, 2), Trend=c, MAE=0.2261, MSE=0.0511, RMSE=0.2261
Month 2019-07: Order=(1, 0, 2), Trend=t, MAE=0.0192, MSE=0.0004, RMSE=0.0192
Month 2019-08: Order=(1, 0, 1), Trend=c, MAE=0.1237, MSE=0.0153, RMSE=0.1237
Month 2019-09: Order=(0, 0, 0), Trend=ct, MAE=0.0980, MSE=0.0096, RMSE=0.0980
Month 2019-10: Order=(2, 1, 1), Trend=n, MAE=0.0802, MSE=0.0064, RMSE=0.0802
Month 2019-11: Order=(0, 1, 0), Trend=t, MAE=0.0769, MSE=0.0059, RMSE=0.0769
Month 2019-12: Order=(0, 0, 2), Trend=c, MAE=0.0294, MSE=0.0009, RMSE=0.0294
Month 2020-01: Order=(1, 0, 0), Trend=n, MAE=0.5512, MSE=0.3038, RMSE=0.5512
Month 2020-02: Order=(1, 0, 1), Trend=n, MAE=0.0283, MSE=0.0008, RMSE=0.0283
Month 2020-03: Order=(0, 1, 1), Trend=n, MAE=0.0176, MSE=0.0003, RMSE=0.0176
Month 2020-04: Order=(1, 0, 0), Trend=c, MAE=0.0036, MSE=0.0000, RMSE=0.0036
Month 2020-05: Order=(0, 0, 1), Trend=n, MAE=0.3694, MSE=0.1364, RMSE=0.3694
Month 2020-06: Order=(2, 1, 2), Trend=ct, MAE=0.0234, MSE=0.0005, RMSE=0.0234
Month 2020-07: Order=(2, 0, 1), Trend=t, MAE=0.3413, MSE=0.1165, RMSE=0.3413
Month 2020-08: Order=(1, 0, 2), Trend=ct, MAE=0.2491, MSE=0.0620, RMSE=0.2491
Month 2020-09: Order=(2, 1, 1), Trend=ct, MAE=0.0899, MSE=0.0081, RMSE=0.0899
Month 2020-10: Order=(0, 0, 2), Trend=c, MAE=0.1934, MSE=0.0374, RMSE=0.1934
Month 2020-11: Order=(2, 1, 1), Trend=n, MAE=0.0029, MSE=0.0000, RMSE=0.0029
```

```
Month 2020-12: Order=(2, 1, 1), Trend=ct, MAE=0.0087, MSE=0.0001, RMSE=0.0087
Month 2021-01: Order=(2, 0, 2), Trend=c, MAE=0.1750, MSE=0.0306, RMSE=0.1750
Month 2021-02: Order=(1, 0, 0), Trend=t, MAE=0.0148, MSE=0.0002, RMSE=0.0148
Month 2021-03: Order=(1, 0, 1), Trend=ct, MAE=0.6291, MSE=0.3958, RMSE=0.6291
Month 2021-04: Order=(0, 1, 0), Trend=t, MAE=0.0068, MSE=0.0000, RMSE=0.0068
Month 2021-05: Order=(2, 0, 1), Trend=n, MAE=1.1078, MSE=1.2273, RMSE=1.1078
Month 2021-06: Order=(0, 1, 1), Trend=c, MAE=0.0401, MSE=0.0016, RMSE=0.0401
Month 2021-07: Order=(0, 1, 0), Trend=ct, MAE=0.0136, MSE=0.0002, RMSE=0.0136
Month 2021-08: Order=(0, 1, 1), Trend=t, MAE=0.1412, MSE=0.0199, RMSE=0.1412
Month 2021-09: Order=(1, 0, 2), Trend=c, MAE=0.0669, MSE=0.0045, RMSE=0.0669
Month 2021-10: Order=(2, 0, 0), Trend=n, MAE=0.1057, MSE=0.0112, RMSE=0.1057
Month 2021-11: Order=(0, 1, 0), Trend=c, MAE=0.0632, MSE=0.0040, RMSE=0.0632
Month 2021-12: Order=(1, 0, 1), Trend=c, MAE=0.5778, MSE=0.3338, RMSE=0.5778
Month 2022-01: Order=(0, 1, 0), Trend=t, MAE=0.0387, MSE=0.0015, RMSE=0.0387
Month 2022-02: Order=(2, 0, 1), Trend=c, MAE=0.4863, MSE=0.2365, RMSE=0.4863
Month 2022-03: Order=(2, 0, 0), Trend=n, MAE=0.1341, MSE=0.0180, RMSE=0.1341
Month 2022-04: Order=(1, 0, 1), Trend=c, MAE=0.0286, MSE=0.0008, RMSE=0.0286
Month 2022-05: Order=(1, 0, 1), Trend=c, MAE=0.1954, MSE=0.0382, RMSE=0.1954
Month 2022-06: Order=(0, 0, 2), Trend=ct, MAE=0.7934, MSE=0.6295, RMSE=0.7934
Month 2022-07: Order=(1, 1, 1), Trend=ct, MAE=0.0166, MSE=0.0003, RMSE=0.0166
Month 2022-08: Order=(0, 0, 1), Trend=c, MAE=0.0755, MSE=0.0057, RMSE=0.0755
Month 2022-09: Order=(1, 0, 1), Trend=t, MAE=0.6752, MSE=0.4559, RMSE=0.6752
Month 2022-10: Order=(2, 1, 0), Trend=c, MAE=0.1975, MSE=0.0390, RMSE=0.1975
Month 2022-11: Order=(2, 0, 1), Trend=n, MAE=3.2737, MSE=10.7168, RMSE=3.2737
Month 2022-12: Order=(1, 0, 2), Trend=c, MAE=0.2231, MSE=0.0498, RMSE=0.2231
Month 2023-01: Order=(0, 1, 1), Trend=c, MAE=0.4947, MSE=0.2447, RMSE=0.4947
Month 2023-02: Order=(0, 0, 1), Trend=c, MAE=0.4713, MSE=0.2222, RMSE=0.4713
Month 2023-03: Order=(2, 0, 2), Trend=c, MAE=0.0068, MSE=0.0000, RMSE=0.0068
Month 2023-04: Order=(0, 1, 0), Trend=ct, MAE=0.1723, MSE=0.0297, RMSE=0.1723
Month 2023-05: Order=(2, 1, 1), Trend=c, MAE=0.0079, MSE=0.0001, RMSE=0.0079
Month 2023-06: Order=(2, 1, 1), Trend=n, MAE=0.0348, MSE=0.0012, RMSE=0.0348
Month 2023-07: Order=(2, 1, 2), Trend=t, MAE=0.2108, MSE=0.0445, RMSE=0.2108
Month 2023-08: Order=(0, 0, 2), Trend=c, MAE=0.4189, MSE=0.1755, RMSE=0.4189
Month 2023-09: Order=(2, 0, 1), Trend=t, MAE=0.3739, MSE=0.1398, RMSE=0.3739
Month 2023-10: Order=(0, 0, 0), Trend=ct, MAE=0.1321, MSE=0.0174, RMSE=0.1321
Month 2023-11: Order=(2, 1, 2), Trend=t, MAE=0.0219, MSE=0.0005, RMSE=0.0219
Month 2023-12: Order=(1, 0, 0), Trend=t, MAE=0.3932, MSE=0.1546, RMSE=0.3932
Month 2024-01: Order=(2, 1, 2), Trend=ct, MAE=0.0053, MSE=0.0000, RMSE=0.0053
Month 2024-02: Order=(2, 1, 0), Trend=t, MAE=0.0010, MSE=0.0000, RMSE=0.0010
Month 2024-03: Order=(0, 1, 2), Trend=c, MAE=0.0284, MSE=0.0008, RMSE=0.0284
```

E Grid search result of HK BERT SARIMAX hybrid model

```
Month 2015-01: Order=(0, 1, 2), Trend=t, MAE=0.1035, MSE=0.0107, RMSE=0.1035
Month 2015-02: Order=(0, 1, 1), Trend=t, MAE=0.5397, MSE=0.2912, RMSE=0.5397
Month 2015-03: Order=(0, 0, 1), Trend=t, MAE=0.0115, MSE=0.0001, RMSE=0.0115
Month 2015-04: Order=(1, 1, 2), Trend=n, MAE=0.0085, MSE=0.0001, RMSE=0.0085
Month 2015-05: Order=(0, 1, 2), Trend=t, MAE=0.0448, MSE=0.0020, RMSE=0.0448
Month 2015-06: Order=(2, 0, 0), Trend=t, MAE=0.1700, MSE=0.0289, RMSE=0.1700
Month 2015-07: Order=(0, 1, 2), Trend=n, MAE=0.0781, MSE=0.0061, RMSE=0.0781
Month 2015-08: Order=(0, 1, 1), Trend=n, MAE=0.0337, MSE=0.0011, RMSE=0.0337
Month 2015-09: Order=(2, 0, 1), Trend=n, MAE=0.2859, MSE=0.0817, RMSE=0.2859
Month 2015-10: Order=(2, 1, 0), Trend=n, MAE=0.1168, MSE=0.0136, RMSE=0.1168
Month 2015-11: Order=(1, 1, 0), Trend=t, MAE=0.0512, MSE=0.0026, RMSE=0.0512
Month 2015-12: Order=(0, 0, 2), Trend=n, MAE=0.0610, MSE=0.0037, RMSE=0.0610
Month 2016-01: Order=(2, 0, 0), Trend=ct, MAE=0.0140, MSE=0.0002, RMSE=0.0140
Month 2016-02: Order=(1, 1, 2), Trend=t, MAE=0.1955, MSE=0.0382, RMSE=0.1955
Month 2016-03: Order=(1, 0, 1), Trend=t, MAE=0.0679, MSE=0.0046, RMSE=0.0679
Month 2016-04: Order=(0, 0, 0), Trend=c, MAE=0.0104, MSE=0.0001, RMSE=0.0104
Month 2016-05: Order=(1, 0, 0), Trend=c, MAE=0.4680, MSE=0.2190, RMSE=0.4680
Month 2016-06: Order=(2, 1, 1), Trend=t, MAE=0.0070, MSE=0.0000, RMSE=0.0070
Month 2016-07: Order=(2, 0, 2), Trend=n, MAE=0.0784, MSE=0.0061, RMSE=0.0784
Month 2016-08: Order=(2, 1, 2), Trend=c, MAE=0.0249, MSE=0.0006, RMSE=0.0249
Month 2016-09: Order=(0, 1, 1), Trend=c, MAE=0.0167, MSE=0.0003, RMSE=0.0167
Month 2016-10: Order=(1, 1, 2), Trend=ct, MAE=0.0584, MSE=0.0034, RMSE=0.0584
Month 2016-11: Order=(2, 0, 0), Trend=ct, MAE=0.0541, MSE=0.0029, RMSE=0.0541
Month 2016-12: Order=(2, 1, 1), Trend=n, MAE=0.0967, MSE=0.0093, RMSE=0.0967
Month 2017-01: Order=(2, 0, 0), Trend=t, MAE=0.1159, MSE=0.0134, RMSE=0.1159
Month 2017-02: Order=(1, 0, 1), Trend=ct, MAE=0.1076, MSE=0.0116, RMSE=0.1076
Month 2017-03: Order=(0, 1, 1), Trend=t, MAE=0.0168, MSE=0.0003, RMSE=0.0168
Month 2017-04: Order=(2, 0, 0), Trend=ct, MAE=0.0340, MSE=0.0012, RMSE=0.0340
Month 2017-05: Order=(0, 0, 2), Trend=n, MAE=0.1319, MSE=0.0174, RMSE=0.1319
Month 2017-06: Order=(2, 0, 0), Trend=c, MAE=0.0239, MSE=0.0006, RMSE=0.0239
Month 2017-07: Order=(1, 1, 1), Trend=n, MAE=0.0039, MSE=0.0000, RMSE=0.0039
Month 2017-08: Order=(2, 1, 1), Trend=n, MAE=0.0122, MSE=0.0001, RMSE=0.0122
Month 2017-09: Order=(1, 0, 2), Trend=ct, MAE=0.2006, MSE=0.0402, RMSE=0.2006
Month 2017-10: Order=(0, 0, 1), Trend=ct, MAE=0.0838, MSE=0.0070, RMSE=0.0838
Month 2017-11: Order=(0, 1, 2), Trend=ct, MAE=0.0250, MSE=0.0006, RMSE=0.0250
Month 2017-12: Order=(1, 1, 2), Trend=ct, MAE=0.0152, MSE=0.0002, RMSE=0.0152
Month 2018-01: Order=(0, 0, 1), Trend=ct, MAE=0.5483, MSE=0.3006, RMSE=0.5483
Month 2018-02: Order=(1, 0, 1), Trend=ct, MAE=0.0285, MSE=0.0008, RMSE=0.0285
Month 2018-03: Order=(2, 1, 0), Trend=t, MAE=2.5195, MSE=6.3477, RMSE=2.5195
Month 2018-04: Order=(0, 0, 1), Trend=n, MAE=0.1954, MSE=0.0382, RMSE=0.1954
Month 2018-05: Order=(1, 0, 1), Trend=c, MAE=0.0027, MSE=0.0000, RMSE=0.0027
Month 2018-06: Order=(1, 1, 1), Trend=ct, MAE=0.0496, MSE=0.0025, RMSE=0.0496
Month 2018-07: Order=(0, 1, 0), Trend=n, MAE=0.1864, MSE=0.0348, RMSE=0.1864
Month 2018-08: Order=(2, 0, 1), Trend=ct, MAE=0.0331, MSE=0.0011, RMSE=0.0331
Month 2018-09: Order=(0, 0, 0), Trend=c, MAE=0.0225, MSE=0.0005, RMSE=0.0225
Month 2018-10: Order=(0, 0, 2), Trend=t, MAE=0.5694, MSE=0.3242, RMSE=0.5694
Month 2018-11: Order=(2, 0, 0), Trend=n, MAE=0.2300, MSE=0.0529, RMSE=0.2300
Month 2018-12: Order=(2, 0, 0), Trend=n, MAE=0.0168, MSE=0.0003, RMSE=0.0168
Month 2019-01: Order=(0, 1, 0), Trend=t, MAE=0.0297, MSE=0.0009, RMSE=0.0297
Month 2019-02: Order=(1, 0, 2), Trend=ct, MAE=0.2976, MSE=0.0886, RMSE=0.2976
Month 2019-03: Order=(0, 0, 0), Trend=c, MAE=0.0909, MSE=0.0083, RMSE=0.0909
Month 2019-04: Order=(2, 1, 1), Trend=c, MAE=0.0593, MSE=0.0035, RMSE=0.0593
Month 2019-05: Order=(2, 0, 2), Trend=ct, MAE=0.1402, MSE=0.0197, RMSE=0.1402
Month 2019-06: Order=(1, 0, 0), Trend=c, MAE=0.1285, MSE=0.0165, RMSE=0.1285
Month 2019-07: Order=(1, 1, 2), Trend=n, MAE=0.0754, MSE=0.0057, RMSE=0.0754
Month 2019-08: Order=(2, 1, 2), Trend=n, MAE=0.1498, MSE=0.0224, RMSE=0.1498
Month 2019-09: Order=(2, 1, 1), Trend=c, MAE=0.1654, MSE=0.0274, RMSE=0.1654
Month 2019-10: Order=(0, 1, 0), Trend=t, MAE=0.1694, MSE=0.0287, RMSE=0.1694
Month 2019-11: Order=(1, 0, 1), Trend=c, MAE=0.5920, MSE=0.3504, RMSE=0.5920
Month 2019-12: Order=(2, 0, 0), Trend=ct, MAE=0.4497, MSE=0.2022, RMSE=0.4497
Month 2020-01: Order=(2, 0, 2), Trend=n, MAE=0.0874, MSE=0.0076, RMSE=0.0874
Month 2020-02: Order=(0, 0, 2), Trend=t, MAE=0.1037, MSE=0.0108, RMSE=0.1037
Month 2020-03: Order=(0, 1, 1), Trend=t, MAE=0.0013, MSE=0.0000, RMSE=0.0013
Month 2020-04: Order=(1, 0, 2), Trend=t, MAE=0.0267, MSE=0.0007, RMSE=0.0267
Month 2020-05: Order=(0, 1, 2), Trend=t, MAE=0.0845, MSE=0.0071, RMSE=0.0845
Month 2020-06: Order=(0, 1, 2), Trend=t, MAE=0.0002, MSE=0.0000, RMSE=0.0002
Month 2020-07: Order=(0, 0, 2), Trend=ct, MAE=0.3822, MSE=0.1461, RMSE=0.3822
Month 2020-08: Order=(0, 0, 0), Trend=ct, MAE=0.4778, MSE=0.2283, RMSE=0.4778
Month 2020-09: Order=(0, 1, 1), Trend=c, MAE=0.0101, MSE=0.0001, RMSE=0.0101
Month 2020-10: Order=(0, 1, 1), Trend=n, MAE=0.0123, MSE=0.0002, RMSE=0.0123
Month 2020-11: Order=(0, 1, 1), Trend=ct, MAE=0.0016, MSE=0.0000, RMSE=0.0016
```

```
Month 2020-12: Order=(2, 0, 1), Trend=c, MAE=0.0845, MSE=0.0071, RMSE=0.0845
Month 2021-01: Order=(2, 1, 2), Trend=t, MAE=0.0365, MSE=0.0013, RMSE=0.0365
Month 2021-02: Order=(2, 0, 1), Trend=ct, MAE=0.0630, MSE=0.0040, RMSE=0.0630
Month 2021-03: Order=(2, 0, 0), Trend=n, MAE=0.0525, MSE=0.0028, RMSE=0.0525
Month 2021-04: Order=(0, 0, 0), Trend=n, MAE=0.0009, MSE=0.0000, RMSE=0.0009
Month 2021-05: Order=(2, 0, 0), Trend=n, MAE=0.0230, MSE=0.0005, RMSE=0.0230
Month 2021-06: Order=(1, 0, 2), Trend=c, MAE=0.0368, MSE=0.0014, RMSE=0.0368
Month 2021-07: Order=(2, 1, 2), Trend=c, MAE=0.5349, MSE=0.2862, RMSE=0.5349
Month 2021-08: Order=(0, 1, 0), Trend=t, MAE=0.0511, MSE=0.0026, RMSE=0.0511
Month 2021-09: Order=(0, 0, 2), Trend=c, MAE=0.6908, MSE=0.4771, RMSE=0.6908
Month 2021-10: Order=(1, 0, 0), Trend=n, MAE=0.0130, MSE=0.0002, RMSE=0.0130
Month 2021-11: Order=(2, 0, 0), Trend=ct, MAE=0.0237, MSE=0.0006, RMSE=0.0237
Month 2021-12: Order=(2, 0, 0), Trend=n, MAE=0.2940, MSE=0.0864, RMSE=0.2940
Month 2022-01: Order=(0, 0, 0), Trend=c, MAE=0.9208, MSE=0.8479, RMSE=0.9208
Month 2022-02: Order=(1, 0, 2), Trend=t, MAE=0.0287, MSE=0.0008, RMSE=0.0287
Month 2022-03: Order=(1, 0, 1), Trend=n, MAE=0.1011, MSE=0.0102, RMSE=0.1011
Month 2022-04: Order=(1, 0, 0), Trend=ct, MAE=0.1341, MSE=0.0180, RMSE=0.1341
Month 2022-05: Order=(1, 0, 1), Trend=c, MAE=1.1499, MSE=1.3223, RMSE=1.1499
Month 2022-06: Order=(2, 0, 1), Trend=c, MAE=0.0550, MSE=0.0030, RMSE=0.0550
Month 2022-07: Order=(2, 0, 2), Trend=c, MAE=0.2157, MSE=0.0465, RMSE=0.2157
Month 2022-08: Order=(1, 0, 2), Trend=t, MAE=0.6191, MSE=0.3832, RMSE=0.6191
Month 2022-09: Order=(1, 1, 1), Trend=t, MAE=0.0088, MSE=0.0001, RMSE=0.0088
Month 2022-10: Order=(1, 0, 0), Trend=ct, MAE=0.2555, MSE=0.0653, RMSE=0.2555
Month 2022-11: Order=(2, 0, 1), Trend=n, MAE=0.0244, MSE=0.0006, RMSE=0.0244
Month 2022-12: Order=(1, 0, 0), Trend=c, MAE=0.3564, MSE=0.1270, RMSE=0.3564
Month 2023-01: Order=(2, 0, 0), Trend=c, MAE=0.1177, MSE=0.0138, RMSE=0.1177
Month 2023-02: Order=(2, 1, 1), Trend=c, MAE=0.0168, MSE=0.0003, RMSE=0.0168
Month 2023-03: Order=(1, 0, 2), Trend=c, MAE=0.0264, MSE=0.0007, RMSE=0.0264
Month 2023-04: Order=(1, 0, 0), Trend=n, MAE=0.0128, MSE=0.0002, RMSE=0.0128
Month 2023-05: Order=(0, 1, 0), Trend=n, MAE=0.0261, MSE=0.0007, RMSE=0.0261
Month 2023-06: Order=(2, 1, 2), Trend=ct, MAE=0.1692, MSE=0.0286, RMSE=0.1692
Month 2023-07: Order=(1, 0, 2), Trend=c, MAE=0.0126, MSE=0.0002, RMSE=0.0126
Month 2023-08: Order=(2, 0, 0), Trend=t, MAE=0.0214, MSE=0.0005, RMSE=0.0214
Month 2023-09: Order=(2, 0, 0), Trend=n, MAE=0.2243, MSE=0.0503, RMSE=0.2243
Month 2023-10: Order=(1, 1, 2), Trend=ct, MAE=0.0031, MSE=0.0000, RMSE=0.0031
Month 2023-11: Order=(2, 0, 2), Trend=t, MAE=0.0307, MSE=0.0009, RMSE=0.0307
Month 2023-12: Order=(2, 1, 1), Trend=c, MAE=0.0041, MSE=0.0000, RMSE=0.0041
Month 2024-01: Order=(0, 1, 2), Trend=t, MAE=0.0575, MSE=0.0033, RMSE=0.0575
Month 2024-02: Order=(2, 1, 2), Trend=n, MAE=0.0247, MSE=0.0006, RMSE=0.0247
```

F Grid search result of UK BERT SARIMAX hybrid model

```
Month 2018-01: Order=(1, 0, 0), Trend=n, MAE=0.0358, MSE=0.0013, RMSE=0.0358
Month 2018-02: Order=(1, 1, 1), Trend=ct, MAE=0.0569, MSE=0.0032, RMSE=0.0569
Month 2018-03: Order=(1, 0, 1), Trend=c, MAE=0.1008, MSE=0.0102, RMSE=0.1008
Month 2018-04: Order=(2, 1, 2), Trend=c, MAE=0.0039, MSE=0.0000, RMSE=0.0039
Month 2018-05: Order=(0, 1, 2), Trend=c, MAE=0.0045, MSE=0.0000, RMSE=0.0045
Month 2018-06: Order=(2, 1, 0), Trend=c, MAE=0.0342, MSE=0.0012, RMSE=0.0342
Month 2018-07: Order=(2, 1, 0), Trend=n, MAE=0.0578, MSE=0.0033, RMSE=0.0578
Month 2018-08: Order=(2, 0, 0), Trend=ct, MAE=0.0162, MSE=0.0003, RMSE=0.0162
Month 2018-09: Order=(1, 1, 1), Trend=n, MAE=0.0228, MSE=0.0005, RMSE=0.0228
Month 2018-10: Order=(2, 1, 0), Trend=c, MAE=0.0412, MSE=0.0017, RMSE=0.0412
Month 2018-11: Order=(1, 1, 1), Trend=ct, MAE=0.0487, MSE=0.0024, RMSE=0.0487
Month 2018-12: Order=(2, 0, 1), Trend=c, MAE=0.0014, MSE=0.0000, RMSE=0.0014
Month 2019-01: Order=(1, 1, 2), Trend=n, MAE=0.0128, MSE=0.0002, RMSE=0.0128
Month 2019-02: Order=(1, 0, 0), Trend=ct, MAE=0.1819, MSE=0.0331, RMSE=0.1819
Month 2019-03: Order=(2, 0, 2), Trend=c, MAE=0.0156, MSE=0.0002, RMSE=0.0156
Month 2019-04: Order=(1, 1, 2), Trend=t, MAE=0.0043, MSE=0.0000, RMSE=0.0043
Month 2019-05: Order=(1, 1, 0), Trend=t, MAE=0.0221, MSE=0.0005, RMSE=0.0221
Month 2019-06: Order=(2, 0, 0), Trend=n, MAE=0.2370, MSE=0.0562, RMSE=0.2370
Month 2019-07: Order=(2, 1, 0), Trend=ct, MAE=0.0183, MSE=0.0003, RMSE=0.0183
Month 2019-08: Order=(0, 1, 1), Trend=n, MAE=0.0186, MSE=0.0003, RMSE=0.0186
Month 2019-09: Order=(0, 0, 1), Trend=ct, MAE=0.0046, MSE=0.0000, RMSE=0.0046
Month 2019-10: Order=(1, 0, 2), Trend=n, MAE=0.0003, MSE=0.0000, RMSE=0.0003
Month 2019-11: Order=(1, 1, 0), Trend=c, MAE=0.0031, MSE=0.0000, RMSE=0.0031
Month 2019-12: Order=(1, 1, 2), Trend=ct, MAE=0.0028, MSE=0.0000, RMSE=0.0028
Month 2020-01: Order=(1, 0, 1), Trend=n, MAE=0.0045, MSE=0.0000, RMSE=0.0045
Month 2020-02: Order=(2, 0, 0), Trend=c, MAE=0.0226, MSE=0.0005, RMSE=0.0226
Month 2020-03: Order=(1, 0, 2), Trend=c, MAE=0.5051, MSE=0.2552, RMSE=0.5051
Month 2020-04: Order=(0, 0, 0), Trend=c, MAE=0.0201, MSE=0.0004, RMSE=0.0201
Month 2020-05: Order=(2, 1, 2), Trend=n, MAE=0.0036, MSE=0.0000, RMSE=0.0036
Month 2020-06: Order=(0, 1, 2), Trend=c, MAE=0.1822, MSE=0.0332, RMSE=0.1822
Month 2020-07: Order=(2, 0, 1), Trend=c, MAE=0.0957, MSE=0.0092, RMSE=0.0957
Month 2020-08: Order=(1, 0, 1), Trend=c, MAE=1.6085, MSE=2.5873, RMSE=1.6085
Month 2020-09: Order=(0, 0, 2), Trend=ct, MAE=0.3542, MSE=0.1255, RMSE=0.3542
Month 2020-10: Order=(0, 0, 2), Trend=ct, MAE=0.0749, MSE=0.0056, RMSE=0.0749
Month 2020-11: Order=(2, 1, 2), Trend=c, MAE=0.0019, MSE=0.0000, RMSE=0.0019
Month 2020-12: Order=(0, 1, 0), Trend=c, MAE=0.0077, MSE=0.0001, RMSE=0.0077
Month 2021-01: Order=(0, 1, 0), Trend=t, MAE=0.0039, MSE=0.0000, RMSE=0.0039
Month 2021-02: Order=(2, 0, 2), Trend=t, MAE=0.0920, MSE=0.0085, RMSE=0.0920
Month 2021-03: Order=(0, 0, 1), Trend=t, MAE=0.4358, MSE=0.1899, RMSE=0.4358
Month 2021-04: Order=(0, 1, 2), Trend=n, MAE=0.0115, MSE=0.0001, RMSE=0.0115
Month 2021-05: Order=(0, 0, 0), Trend=c, MAE=1.4336, MSE=2.0551, RMSE=1.4336
Month 2021-06: Order=(0, 0, 1), Trend=c, MAE=0.2035, MSE=0.0414, RMSE=0.2035
Month 2021-07: Order=(0, 1, 2), Trend=c, MAE=0.1151, MSE=0.0132, RMSE=0.1151
Month 2021-08: Order=(2, 0, 0), Trend=t, MAE=2.1306, MSE=4.5396, RMSE=2.1306
Month 2021-09: Order=(1, 0, 2), Trend=c, MAE=0.0289, MSE=0.0008, RMSE=0.0289
Month 2021-10: Order=(1, 0, 1), Trend=ct, MAE=0.0155, MSE=0.0002, RMSE=0.0155
Month 2021-11: Order=(2, 1, 0), Trend=ct, MAE=0.0874, MSE=0.0076, RMSE=0.0874
Month 2021-12: Order=(1, 1, 1), Trend=c, MAE=0.1183, MSE=0.0140, RMSE=0.1183
Month 2022-01: Order=(2, 1, 1), Trend=t, MAE=0.0295, MSE=0.0009, RMSE=0.0295
Month 2022-02: Order=(0, 1, 2), Trend=n, MAE=0.0648, MSE=0.0042, RMSE=0.0648
Month 2022-03: Order=(2, 1, 2), Trend=n, MAE=0.0545, MSE=0.0030, RMSE=0.0545
Month 2022-04: Order=(1, 0, 2), Trend=c, MAE=0.0784, MSE=0.0062, RMSE=0.0784
Month 2022-05: Order=(2, 0, 2), Trend=c, MAE=0.4779, MSE=0.2284, RMSE=0.4779
Month 2022-06: Order=(2, 0, 1), Trend=ct, MAE=1.0392, MSE=1.0799, RMSE=1.0392
Month 2022-07: Order=(2, 1, 2), Trend=c, MAE=0.1280, MSE=0.0164, RMSE=0.1280
Month 2022-08: Order=(1, 0, 2), Trend=n, MAE=0.0054, MSE=0.0000, RMSE=0.0054
Month 2022-09: Order=(2, 0, 1), Trend=ct, MAE=0.0116, MSE=0.0001, RMSE=0.0116
Month 2022-10: Order=(2, 0, 1), Trend=n, MAE=0.0879, MSE=0.0077, RMSE=0.0879
Month 2022-11: Order=(0, 1, 2), Trend=ct, MAE=0.0234, MSE=0.0005, RMSE=0.0234
Month 2022-12: Order=(1, 1, 2), Trend=n, MAE=0.0723, MSE=0.0052, RMSE=0.0723
Month 2023-01: Order=(2, 0, 0), Trend=c, MAE=0.0950, MSE=0.0090, RMSE=0.0950
Month 2023-02: Order=(0, 0, 2), Trend=n, MAE=0.2872, MSE=0.0825, RMSE=0.2872
Month 2023-03: Order=(2, 1, 2), Trend=t, MAE=0.0120, MSE=0.0001, RMSE=0.0120
Month 2023-04: Order=(0, 1, 1), Trend=ct, MAE=0.0048, MSE=0.0000, RMSE=0.0048
Month 2023-05: Order=(2, 0, 2), Trend=n, MAE=0.0631, MSE=0.0040, RMSE=0.0631
Month 2023-06: Order=(1, 0, 1), Trend=t, MAE=0.0064, MSE=0.0000, RMSE=0.0064
Month 2023-07: Order=(0, 1, 2), Trend=n, MAE=0.0889, MSE=0.0079, RMSE=0.0889
Month 2023-08: Order=(1, 0, 1), Trend=ct, MAE=0.0154, MSE=0.0002, RMSE=0.0154
Month 2023-09: Order=(1, 0, 2), Trend=t, MAE=0.0999, MSE=0.0100, RMSE=0.0999
Month 2023-10: Order=(2, 0, 2), Trend=t, MAE=0.0740, MSE=0.0055, RMSE=0.0740
Month 2023-11: Order=(0, 1, 2), Trend=n, MAE=0.1858, MSE=0.0345, RMSE=0.1858
```

```
Month 2023-12: Order=(1, 1, 0), Trend=n, MAE=0.0138, MSE=0.0002, RMSE=0.0138
Month 2024-01: Order=(1, 0, 0), Trend=n, MAE=0.0041, MSE=0.0000, RMSE=0.0041
Month 2024-02: Order=(0, 1, 2), Trend=c, MAE=0.0147, MSE=0.0002, RMSE=0.0147
```

G Data Access Authorisation

FW: Something else - Request for Permission to Use JIL's Hong Kong Property Market Monitor and monthly reports in Academic Research

Chung, Cathie <Cathie.Chung@jll.com>
Mon 19/02/2024 03:41
To: LUK, DAVID (PGT) <David.Luk@warwick.ac.uk>
Cc: Yip, Louisiana <Louisiana.Yip@jll.com>

Hi David

Thank you for reaching out to us regarding the use of our research report content and data for your dissertation. We appreciate your interest in our work and are happy to accommodate your request.

We would be happy to grant you permission to use our research report content and data in your academic research. As you mentioned, we kindly ask that you ensure proper citation and attribution is provided for any content used, as stated in our terms and conditions. Additionally, we appreciate your commitment to respecting all copyright considerations and restrict the use of our work strictly to academic purposes.

Thank you for adhering to our terms and conditions and your understanding of the importance of proper citation and copyright compliance. Should you have any further questions, please feel free to reach out.

Best wishes with your dissertation!

Best regards

Cathie Chung, MRICS
Senior Director, Hong Kong Research
JLL
7/F One Taikoo Place
979 King's Road
Hong Kong SAR, China
T +852 2846 5237
F +852 2845 9117
jll.com.hk

Figure 21: Data Access Authorisation from JLL

RE: [EXTERNAL] get-in-touch

Angela Fung <angela.fung@hk.knightfrank.com>
Tue 20/02/2024 02:48
To: LUK, DAVID (PGT) <David.Luk@warwick.ac.uk>

Hi David

Thank you for your email. We are delighted to hear that you found our monthly reports' content useful for your dissertation.

We kindly request that you properly acknowledge the sources when utilising our data/content in your project.

Thank you.

Regards
Angela

Knight Frank 萊坊

Angela Fung
Senior Director, Public Relations
Marketing & Communications, Greater China
4/F Shui On Centre
6-8 Harbour Road
Wanchai
Hong Kong

D +852 2846 7175
M +852 9858 9191
T +852 2840 1177
F +852 2840 0600

Knight Frank Petty Limited | EAA (Company) Lic No C-010431
Knight Frank Hong Kong Limited | EAA (Company) Lic No C-013197
Knight Frank (Services) Limited | EAA (Company) Lic No C-012848
C P Property Management Limited
Knight Frank Asset Appraisal Limited

Figure 22: Data Access Authorisation from Knight Frank