MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
V.N. Karazin Kharkiv National University
School of Mathematics and Computer Science
Department of Theoretical and Applied Informatics

Master's Thesis

Prediction of the dynamics COVID19 epidemic process of using the Lasso regression model

Author:
Final year Master's Program student,
group 63
specialty - Computer Sciences and
Information Technologies,
educational program: "Informatics"
Chen Guoxin
Supervisor: Ievgen Meniailov
Reviewer: Kseniia Bazilevych
Adviser: Stas Kachanov

Kharkiv, 2024

Deep Learning-Based COVID-19 Data Analysis and Prediction

Chen Guoxin

Abstract

This study integrates machine learning and deep learning methods to predict the COVID-19 pandemic. Advanced prediction techniques, such as Lasso regression and Long Short-Term Memory (LSTM) networks, were employed to enhance prediction accuracy and reliability. The application of these algorithms for cumulative case prediction was explored, aiming to provide scientific support for public health decision-making. The research details the processes of data preprocessing, model training, evaluation, and visualization to maintain generalization and adaptability in the dynamic pandemic scenario. Additionally, a Multi-Scale LSTM-Attention (MSLA) model was proposed to extract multi-period features from input sequences (e.g., periods of 3, 5, 7, 14, 21 days from the past 30 days). These features are critical for addressing data non-stationarity. Experimental results demonstrate the model's superior long-term prediction performance (30 days), with a 15.22% reduction in MAE compared to LSTM.

Keywords: Lasso, LSTM, COVID-19 prediction, Deep Learning, Attention, Multi-scale

Contents

1 Introduction

1.1 Research Background and Significance

The global outbreak of the COVID-19 pandemic has had a profound impact on public health systems and socio-economic structures worldwide, highlighting the urgent need for effective forecasting tools to aid decision-making [1]. As powerful data analysis tools, machine learning and deep learning have demonstrated significant potential in epidemic forecasting [2]. This study aims to construct a multi-model ensemble framework for epidemic forecasting by integrating methods such as Lasso regression, Long Short-Term Memory (LSTM) networks, and LSTM-Attention models. The framework is designed to capture complex patterns and temporal dependencies in epidemic data, providing more accurate prediction results to support public health decision-making.

Based on epidemic data from China, this research first preprocesses and engineers a large volume of data to ensure data quality and extract meaningful information. Subsequently, it employs Lasso regression as a machine learning method, alongside LSTM and other deep learning models, to predict epidemic trends. By comparing the performance of different models, this study adopts ensemble techniques to improve prediction accuracy and robustness. The paper details the processes of model training, evaluation, and result visualization and explores optimizing predictive performance by adjusting model parameters and architectures. Through this study, we aim to provide a robust forecasting tool for the public health sector, enabling a scientific and precise response to future public health challenges.

1.2 Current Research on Prediction Methods

1.2.1 Traditional Prediction Methods

Traditional forecasting techniques primarily include the moving average method, exponential smoothing, and the Autoregressive Integrated Moving Average (ARIMA) model. The moving average method [3] predicts future infection counts by calculating the arithmetic mean of historical data. While straightforward, it has limitations in capturing dynamic epidemic trends. Exponential smoothing assigns higher weights to more recent data points, making it suitable for handling datasets with trends and seasonal variations. The ARIMA model [4] integrates autoregressive, differencing, and moving average techniques, making it particularly effective for analyzing non-stationary time series data. These methods are generally employed for short-term forecasting and are often used in combination to enhance prediction accuracy.

The exponential smoothing method, a time series analysis technique [5] initially proposed by Robert G. Brown, predicts future infection counts by computing the weighted average of historical data. In this approach, weights vary inversely with the temporal distance to the prediction point, assigning greater weight to more recent data points and less to older ones to improve accuracy. Extensions of this method introduce covariates for joint estimation, validated using daily infection datasets from Spain. Fuzzy theory-optimized exponential smoothing models, employing least squares and simulated annealing algorithms, have been shown to enhance prediction precision [6]. Additionally, gradient optimization techniques have improved the accuracy of triple exponential smoothing.

Kalman filtering, a minimum variance estimation technique [7], assumes system noise follows a normal distribution. It is suitable for linear systems and constructs predictive models using state equations. By reconstructing multivariate phase spaces using chaos theory and the C-C method, Kalman filtering has been applied for predictions [8]. Customized Kalman filter models have been developed to address holiday-specific characteristics, effectively resolving issues caused by small sample sizes and large time spans. However, the primary limitation of Kalman filtering lies in its applicability to linear systems, whereas most real-world systems are nonlinear.

Despite advancements in technology and the increasing popularity of modern methods such as neural networks and machine learning, traditional methods remain valuable. These approaches continue to provide insights for predicting COVID-19 infection counts, especially in scenarios with limited data or low complexity requirements.

1.2.2 Machine Learning-Based Prediction Methods

Machine learning-based methods for predicting COVID-19 infection counts have gained increasing attention in recent years due to advancements in data science technologies. Compared to traditional statistical methods such as time series analysis, machine learning methods exhibit significant advantages in handling complex and nonlinear data. With their strong fitting capabilities [9], these methods can capture intricate patterns in the spread of epidemics. They allow for the integration of various input variables, such as meteorological data, holidays, and population mobility, to improve prediction accuracy. Furthermore, machine learning models can automatically adjust parameters based on new data, adapting to changes in infection trends, and generally provide higher prediction accuracy compared to traditional approaches.

Artificial Neural Networks (ANNs) simulate the learning mechanisms of the human brain and process complex input-output relationships through multi-layer nodes. In COVID-19 infection prediction, ANNs can learn how different factors influence infection counts to make accurate forecasts [10].

Support Vector Machines (SVMs) are supervised learning models particularly suitable for classification and regression tasks. In infection prediction [11], SVMs construct regression models to forecast future trends. Originally proposed by Corinna Cortes, SVMs are effective for small sample sizes and nonlinear problems. The key to SVM performance lies in the choice of kernel functions, which map data to higher-dimensional spaces, significantly impacting prediction accuracy. Combining historical infection data and weather data, SVMs trained with particle swarm optimization (PSO) for parameter adjustment have achieved enhanced accuracy [13]. Wavelet-transformed historical data, optimized by advanced fruit fly optimization algorithms, also demonstrate the effectiveness of SVMs [14]. Online sequential extreme SVM regression algorithms address the computational challenges posed by new data [15]. However, SVMs require significant computation time for large datasets and are sensitive to changes in infection characteristics, necessitating kernel function adjustments, which may impair fitting accuracy.

XGBoost (Extreme Gradient Boosting) is an efficient machine learning algorithm widely applied in predicting COVID-19 infection counts due to its high accuracy, fast execution, and ability to handle missing values. It provides accurate short- and ultra-short-term predictions, which are critical for epidemic control and resource allocation. XGBoost considers various influencing factors, such as meteorological data (e.g., dry-bulb and dew-point temperatures), temporal features (e.g., hour of the day, day of the week, holidays), historical infection data (e.g., counts from the same time the previous week or day), and averages from the preceding 24 hours. Additionally, XGBoost [16] is often combined with other machine learning models, such as Bi-LSTM, to incorporate attention mechanisms for better handling of long-term dependencies in time series data. However, XGBoost's reliance on ensemble methods often leads to overfitting, reducing the model's generalization capability.

Although machine learning algorithms can handle large datasets and automatically learn patterns, they are typically restricted to panel data, limiting their ability to extract trends, causal relationships, and long-term dependencies from time series data. Model accuracy requires further improvement.

1.2.3 Deep Learning-Based Prediction Methods

Deep learning methods have shown significant potential in predicting COVID-19 infection and mortality rates [17], particularly in handling large datasets and capturing complex patterns. As epidemic data often take the form of time series, various time series neural network models have been widely explored.

Long Short-Term Memory (LSTM) networks, a special type of Recurrent Neural Network (RNN), effectively address the long-term dependency challenges in time series data. In COVID-19 infection prediction, LSTMs capture trends and periodic features in time series, achieving accurate forecasts [18–19].

Convolutional Neural Networks (CNNs), primarily used for image recognition, can also extract local features from time series data in infection prediction. Combining CNNs with LSTMs further enhances predictive capabilities [20]. Gated Recurrent Units (GRUs) [21], a simplified version of LSTMs, reduce computational complexity while maintaining strong performance. Attention mechanisms enhance the model's focus on critical parts of input sequences [22].

While LSTM models are well-suited for handling long sequence data, their accuracy decreases as sequence length increases due to challenges in capturing internal relationships in long sequences. Introducing attention mechanisms to LSTM models addresses this limitation by highlighting key features influencing infection counts. Bidirectional mechanisms, such as Bidirectional LSTM (BiLSTM) and Bidirectional GRU (BiGRU), improve the capture of bidirectional semantic dependencies in data.

These methods can be summarized as multivariate time series prediction problems with different feature extraction techniques. However, their performance on data with varying periodicities remains suboptimal.

1.3 Summary of Existing Research

A comprehensive analysis of existing research reveals the following conclusions: Current COVID-19 prediction studies often lack sufficient consideration of spatial features. Accurate COVID-19 forecasting is crucial for public health planning and response strategies, with its core goal being to predict epidemic trends over a future period. However, existing studies predominantly rely on time series analyses of historical data, neglecting the potential impacts of spatial characteristics on epidemic distribution and changes.

Furthermore, current research often overlooks the coupled relationships and periodic interactions between temporal and spatial features in epidemic transmission. Epidemic spread is influenced by complex interactions among factors such as seasonality, time, geography, and weekly cycles.

To enhance COVID-19 prediction accuracy, incorporating external features such as weather and social activities into forecasting models is advisable. For instance, using meteorological data (e.g., temperature, seasonality) as input features alongside historical epidemic data can provide a more comprehensive assessment of these factors' impacts. Time series analysis methods can model relationships between epidemic data and external features, offering deeper insights into these factors' influence mechanisms and optimizing prediction models. Despite this potential, current research often fails to account for these external features adequately, highlighting the need for further exploration, such as multi-scale and multi-period approaches.

1.4 Main Structure of the Paper

This paper focuses on the prediction of COVID-19 data, discussing the research background and current state of the field. Different machine learning and deep learning prediction models are utilized, and various ablation experiments are conducted based on real-world production scenarios. The specific structure of the paper is shown in Figure 1. The content of each chapter is as follows:

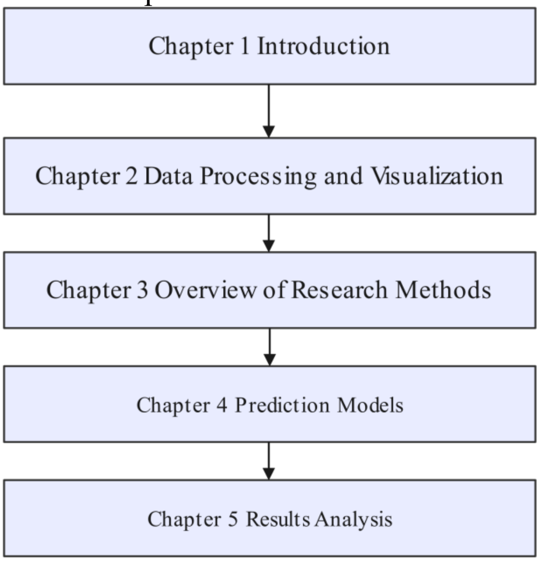

Figure 1: Flowchart of the Paper's Structure

This paper is divided into five main sections. First, in **Chapter 1: Introduction**, we will succinctly explain the purpose, significance, and expected outcomes of the research, laying a solid foundation for the subsequent in-depth discussion. Next, **Chapter 2: Data Processing and Visualization** will focus on the acquisition, cleaning, and preliminary exploration of the raw data, using visual tools such as charts to intuitively present data characteristics for a better understanding and utilization of the data resources. Then, in **Chapter 3: Research Methodology Overview**, we will provide a detailed introduction to the methodological framework used in this study, including, but not limited to, theoretical foundations, technical roadmaps, and key challenges that may be encountered, aiming to give the reader a clear understanding of the overall research strategy. Moving on to **Chapter 4: Prediction Models**, we will meticulously analyze the various prediction models we have constructed, covering their design concepts, algorithm implementations, and performance evaluations in practical applications. Finally, in **Chapter 5: Results Analysis**, we will integrate the findings from the previous sections and, through rigorous logical reasoning and empirical verification, extract core insights or recommendations that hold practical significance, completing the knowledge transfer of this paper.

In conclusion, this paper aims to systematically reveal the inherent patterns and development trends of COVID-19 data prediction research from multiple perspectives, with the goal of providing valuable reference information to practitioners in this field.

2 Relevant Technical Principles

2.1 LSTM Principle

Long Short-Term Memory (LSTM) networks [23] are a special type of Recurrent Neural Network (RNN) designed to address the gradient vanishing or explosion problems encountered by traditional RNNs when processing long sequences of data. LSTMs introduce three key gating mechanisms—the Forget Gate, Input Gate, and Output Gate—whose calculation formulas are given by (1), (2), (3), and (4). These three gates effectively maintain and update the internal states and outputs of the network, allowing LSTMs to capture long-term dependencies in time series data. The detailed structure of the LSTM is shown in Figure 2.

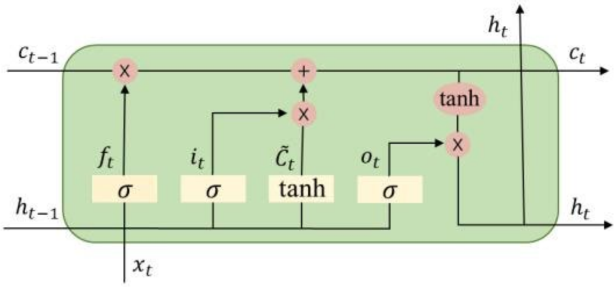

Figure 2: LSTM Structure Diagram

The Input Gate is responsible for deciding how much new information should be added to the cell state at the current time step. The Forget Gate determines what information should be discarded from the cell state at the current time step. The introduction of these two gates allows LSTM to selectively retain or forget information at different time points in the sequence. Specifically, the Forget Gate reads the output from the previous time step and the input at the current time step, then outputs a value between 0 and 1, indicating the degree to which each state unit should be forgotten. Subsequently, the Input Gate determines what new information will be stored in the state, also outputting a value between 0 and 1, which determines the extent of the cell state update.

$$f_t = \sigma\left(W_f h_{t-1} + U_f x_t + b_f\right) \tag{2-1}$$

$$i_t = \sigma\left(W_i h_{t-1} + U_i x_t + b_i\right) \tag{2-2}$$

$$a_t = \tanh\left(W_a h_{t-1} + U_a x_t + b_a\right) \tag{2-3}$$

$$C_t = C_{t-1} \odot f_t + i_t \odot a_t \tag{2-4}$$

The cell state is a core concept in the LSTM network, carrying the sequence information across all time steps of the network. At each time step, the cell state is updated based on the outputs of the Forget Gate and the Input Gate. Specifically, the output of the Forget Gate is element-wise multiplied by the cell state from the previous time step, enabling the selective forgetting of information. Meanwhile, the output of the Input Gate is element-wise multiplied with the new input at the current time step, generating a candidate vector, which is then added to the updated cell state. In this way, the cell state undergoes both information update and filtering.

The Output Gate determines the value of the next hidden state, which can be considered the output of the network at the current time step. The Output Gate reads the updated cell state and the current time step's input, then outputs a value between 0 and 1, which determines the value of the hidden state. The hidden state is a linear transformation of the cell state, calculated using a learned weight matrix and a bias term. Finally, the output of the Output Gate is element-wise multiplied by the hidden state to produce the final output, which can be used for prediction or as the input for the next time step.

2.2 Attention Mechanism

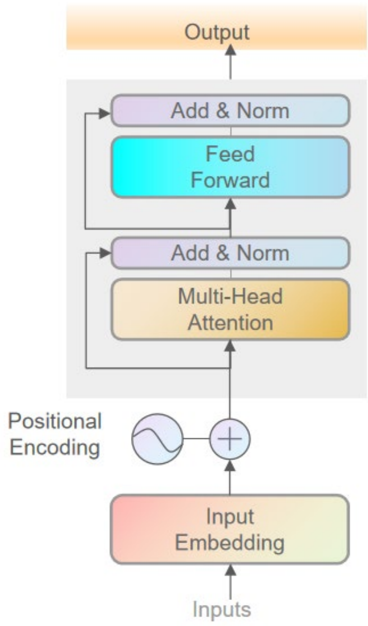

Figure 3: Attention Mechanism Diagram

Figure 3 shows the architecture of the encoder part of a Transformer model [24]. The core of this architecture lies in its Multi-Head Attention mechanism, which allows the model to focus on different pieces of information when processing sequence data, thereby capturing richer contextual relationships.

The input data is first transformed into embedding vectors through the Input Embedding layer. This process maps the features in the sequence to a continuous vector space to capture semantic relationships between different time points. Subsequently, Positional Encoding is added to the embedding vectors to provide positional information for each element in the sequence, as the Transformer model itself lacks a sequence processing mechanism. Next, the data is processed through the Multi-Head Attention mechanism, where multiple attention heads work in parallel. Each head independently calculates attention weights, which are then combined to obtain a richer representation.

After the Multi-Head Attention mechanism, the output is combined with the input via a residual connection and passed through a normalization layer, typically Layer Normalization, to stabilize the training process and accelerate convergence. Then, the data undergoes a non-linear transformation through a Feed Forward layer, which generally consists of two linear transformations with an activation function (e.g., ReLU) in between. The final output is a transformed vector, which can be used for downstream tasks.

Applying the self-attention mechanism to the output of each time step in an LSTM (Long Short-Term Memory) network can further improve the model's performance, especially when handling long sequence data.

2.3 Lasso Regression

Lasso regression is a widely used regression method in machine learning [25]. It introduces an L1 regularization term to promote sparsity in the model coefficients, enabling automatic feature selection. The basic principle of Lasso regression is to add a regularization term to the least squares objective function, as shown in the minimization expression (2-1).

$$\|Y - X\beta\|^2 + \lambda\|\beta\|_1 \tag{2-1}$$

In this expression, y is the vector of observed values, X is the feature matrix, β is the regression coefficient vector, and λ is the hyperparameter that controls the strength of regularization. The introduction of the L1 regularization term forces the coefficients of some features to shrink to zero, thereby achieving feature selection.

When performing Lasso regression, it is necessary to choose an appropriate regularization parameter λ. This can be done using cross-validation, for example, by using LassoCV or GridSearchCV for hyperparameter tuning. During the feature selection process, as λ increases, the coefficients of less important features gradually become zero, while the coefficients of important features remain non-zero. By observing the changes in coefficients for different α (equivalent to λ), it is possible to determine which features are important.

The advantage of Lasso regression in feature selection lies in its interpretability. Since it can set the coefficients of irrelevant or redundant features to zero, the resulting model is more compact. Additionally, Lasso regression is well-suited for high-dimensional datasets, as it can alleviate the curse of dimensionality through feature selection.

3 Data Preprocessing and Visualization

3.1 Data Download

This study selects data from China for research. And download the data from COVID-19 data | WHO COVID-19 dashboard, the daily data link is: https://srhdpeuwpubsa.blob.core.windows.net/whdh/COVID/WHO-COVID-19-global-daily-data.csv. Since the daily data just contains data until 2024-09-22, so I download the weekly data to complete 2024-09-23 ~ 2024-09-30. Weekly data link is: https://srhdpeuwpubsa.blob.core.windows.net/whdh/COVID/WHO-COVID-19-global-data.csv.

3.1 Data Preprocessing

Table 1 shows a portion of the data used in this study, which records the daily reported new and cumulative confirmed COVID-19 cases, as well as new deaths in

China (country code: CN) within the World Health Organization's Western Pacific Region (WPR) from April 13, 2020, to April 21, 2020. Each record includes the report date (**Date_reported**), country code, country name, WHO region, the number of new confirmed cases on that day (**New_cases**), the total number of cumulative confirmed cases (**Cumulative_cases**), and the number of new deaths on that day (**New_deaths**), reflecting the development trend and statistical changes during this period.

Table 1: A Portion of the Raw Data

Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
2020/4/13	CN	China	WPR	115	83597	2	3351
2020/4/14	CN	China	WPR	99	83696	0	3351
2020/4/15	CN	China	WPR	49	83745	1	3352
2020/4/16	CN	China	WPR	52	83797	0	3352
2020/4/17	CN	China	WPR	27	83824	0	3352
2020/4/18	CN	China	WPR	356	84180	1290	4642
2020/4/19	CN	China	WPR	21	84201	0	4642
2020/4/20	CN	China	WPR	36	84237	0	4642
2020/4/21	CN	China	WPR	13	84250	0	4642
2020/4/22	CN	China	WPR	37	84287	0	4642
2020/4/23	CN	China	WPR	15	84302	0	4642

The dataset preprocessing mainly consists of the following steps:

1. **Data Cleaning**: Handling missing values is a crucial task during the data preprocessing phase, as it directly affects the accuracy of subsequent data analysis and modeling. Therefore, this study adopts interpolation methods, which are more suitable for estimating missing values. Linear interpolation is used to fill in the missing data, which is particularly effective for time-series data or data with obvious trends.
2. **Outlier Detection**: To ensure the quality of the data, outlier detection is performed to identify and handle potential extreme values. A common method used is the standard deviation-based detection approach. This method assumes that the data follows a normal distribution, and any data point that is more than a certain number of standard deviations away from the mean is considered an outlier. After the initial processing of the dataset, the mean and standard deviation of each column can be calculated to identify all data points that deviate more than a specified number of standard deviations from the mean. Once outliers are identified, they can be either deleted or replaced with other values (such as the median, mean, or boundary values) depending on the specific situation.

This rule-based outlier detection method effectively improves the reliability of the data and the validity of the analysis results. **Figure 4** shows the before-and-after effects of the data processing, illustrating how the data becomes smoother after preprocessing, with smaller fluctuations and noise removed.

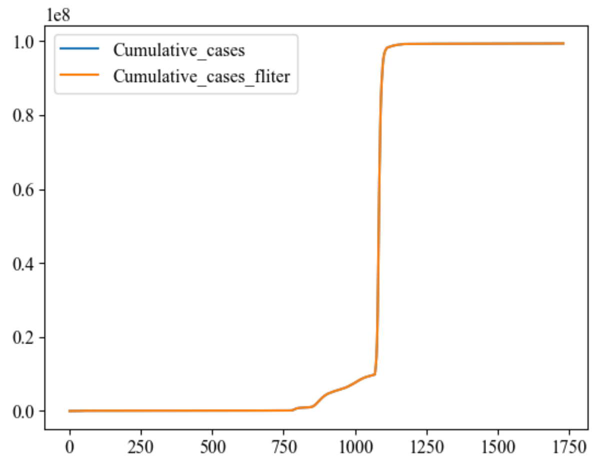

Figure 4: Raw Data vs. Smoothed Data

This study uses the **min-max normalization** method for data normalization. After applying this method, the original data is mapped to the [0, 1] range, reducing the original data range and saving time costs for prediction. The specific calculation formula for data normalization is given in **Equation (3-1)**:

$$y' = \frac{y - y_{\min}}{y_{\max} - y_{\min}} \tag{3-1}$$

Where y represents the original data, $y_{\max}$ is the maximum value in the original data, $y_{\min}$ is the minimum value in the original data, and y' is the result after normalization. Once the prediction model has completed the prediction task, the prediction results need to be **denormalized** to restore the data to its original scale. The specific calculation formula for data denormalization is given in **Equation (3-2)**:

$$y = y' \times \left(y_{\max} - y_{\min}\right) + y_{\min} \tag{3-2}$$

This study uses the **Root Mean Square Error (RMSE)** to evaluate the predictive performance of the model, and it is also used as the loss function for the model. The calculation formula is given in **Equation (3-3)**. Where O_i represents the observed values, P_i represents the predicted values, and N is the total number of samples.

$$RMSE = \sqrt{\frac{1}{N}\left(\sum_{i=1}^{N}\left(O_i - p_i\right)^2\right)} \tag{3-3}$$

Additionally, this study uses **Mean Absolute Error (MAE)** and **Mean Squared Error (MSE)** as evaluation metrics for the model.

- **Mean Squared Error (MSE)** is the sum of the squared differences between predicted values and actual values.
- **Mean Absolute Error (MAE)** is another loss function used for regression models. MAE is the sum of the absolute differences between the target values and the predicted values.

Their specific calculation formulas are given in **Equation (3-4)** and **Equation (3-6)**:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \overline{y}_i\right)^2 \tag{3-4}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|\left(y_i - \overline{y}_i\right)\right| \tag{3-6}$$

The formula for **Relative Error** is given in **Equation (3-7)**: Where Y_{pred} is the predicted result and Y_{real} is the actual value.

$$RE = \frac{Y_{pred}}{Y_{\text{real}}} \times 100\% \tag{3-7}$$

3.2 Data Presentation

Subsequently, this study performs descriptive statistics on the collected data, as shown in **Table 2**. The data covers 1,724 days of epidemic statistics, including daily new confirmed cases, cumulative cases, daily new deaths, and cumulative deaths.

The data shows that, on average, approximately 57,645 new confirmed cases were reported daily, with an average of about 37,676,665 cumulative cases. The average daily new deaths were around 70, while the average cumulative deaths were about 49,697.

In addition, the standard deviations for new cases and cumulative cases were 457,776 and 47,068,774, respectively, indicating significant daily fluctuations in case numbers. Similar trends were observed for new deaths and cumulative deaths, but the magnitude of the fluctuations was smaller.

Table 2: Descriptive Statistics of COVID-19 Data in China

	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
count	1724	1724	1724	1724
mean	57645	37676665	70	49697
std	457776	47068774	1095	53633
min	0	1	0	0
25%	23	102167	0	4848
50%	104	1716445	0	15630
75%	1464	99296718	19	121536
max	6966046	99380363	44047	122358

From the distribution perspective, 25% of the days had fewer than 23 new cases, and 75% of the days had fewer than 1,464 new cases, with a cumulative case number of fewer than 102,167 cases and 99,296,718 cases, respectively. For deaths, 25% of the days had no new deaths reported, with cumulative deaths not exceeding 4,848; 75% of the days had fewer than 19 new deaths, with cumulative deaths not exceeding 121,536. These statistics suggest that during the early stages of the pandemic, new cases and deaths were relatively low, but as the pandemic progressed, these numbers surged, particularly during the peak of the outbreak.

Moreover, the **maximum number of new cases** reached 6,966,046, and the **maximum cumulative cases** totaled 99,380,363, reflecting extreme circumstances on certain days during the pandemic's peak. The **maximum cumulative deaths** reached 122,358, highlighting the severe impact the pandemic had on global health.

Figure 5 shows the distribution of new cases per year from 2020 to 2024. It is evident that the number of new cases was very low in 2020, almost approaching zero. In 2021, the new case count increased slightly, but it remained at a relatively low level. In 2022, new cases saw a significant rise, peaking at over 6 million cases. New cases decreased in 2023 but still remained higher than in 2021, showing some fluctuation. The 2024 data points currently indicate a return to levels close to zero, possibly indicating a return to the same level as in 2020.

Overall, **2022** stands out as an exceptional year, with new cases far higher than in other years. **2023** saw a decline but still remained higher than in 2020 and 2021. The data for **2024** currently shows case numbers returning to lower levels.

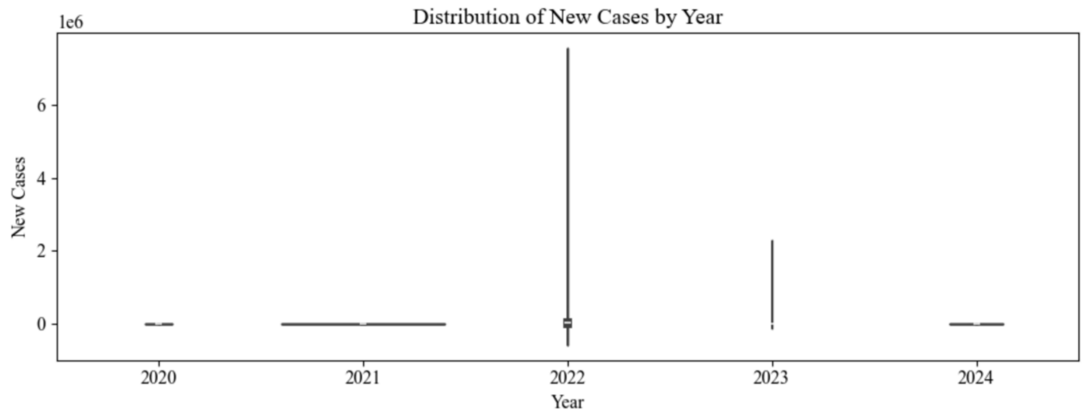

图 5 每年新增感染人数的小提琴图

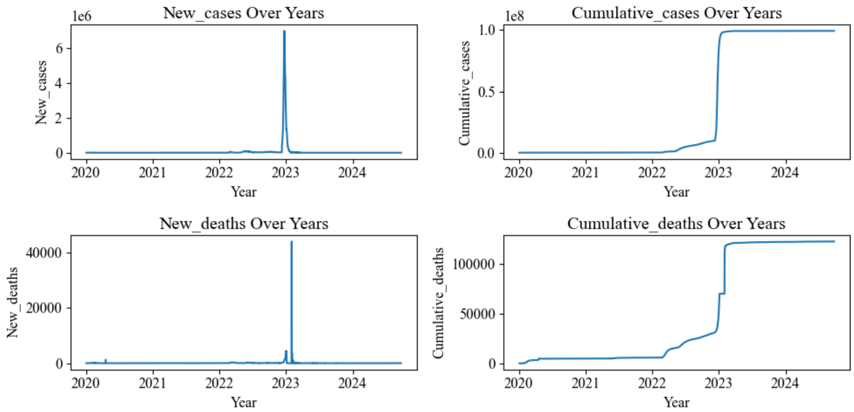

Figure 6: Overall Trend of Four Key Indicators

Figure 6 consists of four subplots, each displaying the trend of new cases, cumulative cases, new deaths, and cumulative deaths from 2020 to 2024. In the "New cases Over Years" and "New deaths Over Years" subplots, we can observe a significant peak in 2023, indicating a large number of new cases and new deaths during that year. In the "Cumulative cases Over Years" and "Cumulative deaths Over Years" subplots, both cumulative cases and cumulative deaths grew slowly before 2023, but experienced a sharp increase in 2023, after which the trends stabilized.

3.3 Sliding Window for Dataset Generation

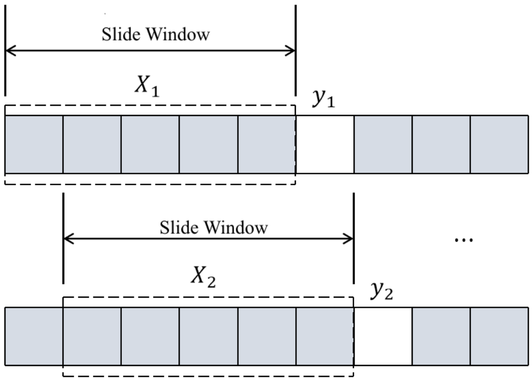

Figure 7: Sliding Window

Figure 7 demonstrates the concept of the Sliding Window. It processes data by moving a fixed-size window across the data sequence, and the data within the window is processed or analyzed each time the window shifts.

In this figure, there are two sliding windows, labeled as X_1 and X_2, along with their corresponding outputs y_1 and y_2. Specifically, X_1 and X_2 represent the portions of the data sequence covered by the sliding window at different positions, and y_1 and y_2 are the output results corresponding to these window positions.

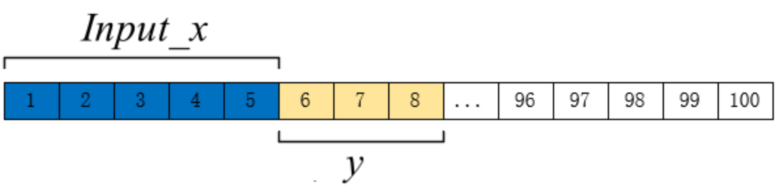

Figure 8： Slide Windows

As shown in Figure 8, the size of the sliding window is not fixed, and the output length is also variable. After each movement, the data inside the window is updated, producing a new output. This allows the algorithm to capture local patterns or trends in the data while maintaining the ability to process the data stream in real-time.

4 Prediction Models and Results Analysis

4.1 Problem Definition

Time series forecasting refers to the prediction of the unknown system states over time. This paper primarily focuses on predicting the cumulative COVID-19 infection cases, which essentially involves building an appropriate model based on collected daily case data to predict the number of new cases in the future. Below is the definition of the cumulative infection case prediction problem.

Forecasting Problem Definition: For the multi-step prediction problem of COVID-19 cumulative cases, assume that we have PPP historical time steps of data observations for the new COVID-19 cases $x = (X_1, X_2, \ldots\ldots, X_p) \in R^{Q \times N}$, Where CCC represents the dimension of the features. Our goal is to predict the target values for the next Q time steps $y = (Y_{p+1}, Y_{p+2}, \ldots\ldots, Y_{p+Q}) \in R^{Q \times N}$, The process of predicting the cumulative number of infections can be represented as equation (4-1).

$$Y_{p+1}, Y_{p+2}, \ldots\ldots, Y_{p+Q} = f\left(G; X_1, X_2, \ldots\ldots, X_p\right) \tag{4-1}$$

4.2 Lasso Regression-based Cumulative Infection Prediction

4.2.1 From 2020 year

The statistic data is showing (for model training, use data from the first day of the pandemic in the selected country):

1. Forecast for 3 days:- Real data end date: 2024-09-27. Dates included in the forecast: 2024-09-28 to 2024-09-30. Mean Absolute error (MAE) is 8575221.91. Relative error (RE) is 8.63%. The specific diagram is as follows 图 9.

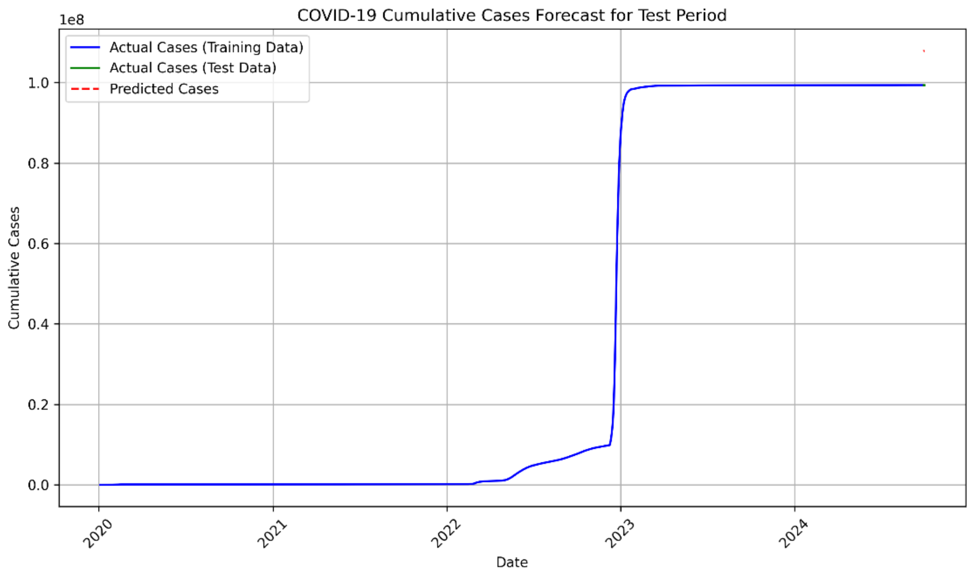

Figure 9： Forecast for 3 days on Lasso from 2020

2. Forecast for 5 days: Real data end date: 2024-09-25. Dates included in the forecast: 2024-09-26 to 2024-09-30. Mean Absolute error (MAE) is 8613817.86. Relative error (RE) is 8.67%. The specific diagram is as follows 图 10.

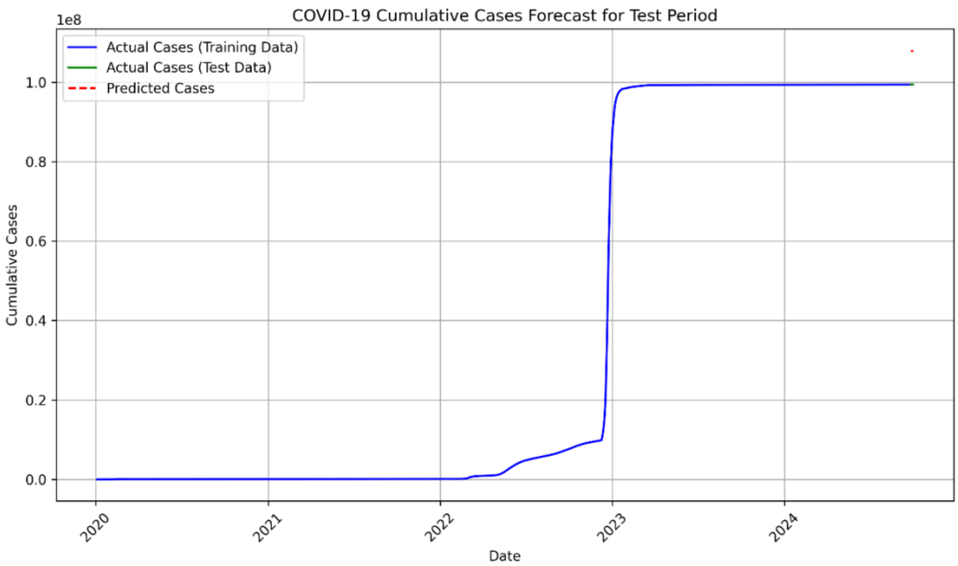

Figure 10： Forecast for 5 days on Lasso from 2020

3. Forecast for 7 days: Real data end date: 2024-09-23. Dates included in the forecast: 2024-09-24 to 2024-09-30. Mean Absolute error (MAE) is 8651950.19. Relative error (RE) is 8.71%. The specific diagram is as follows图 11

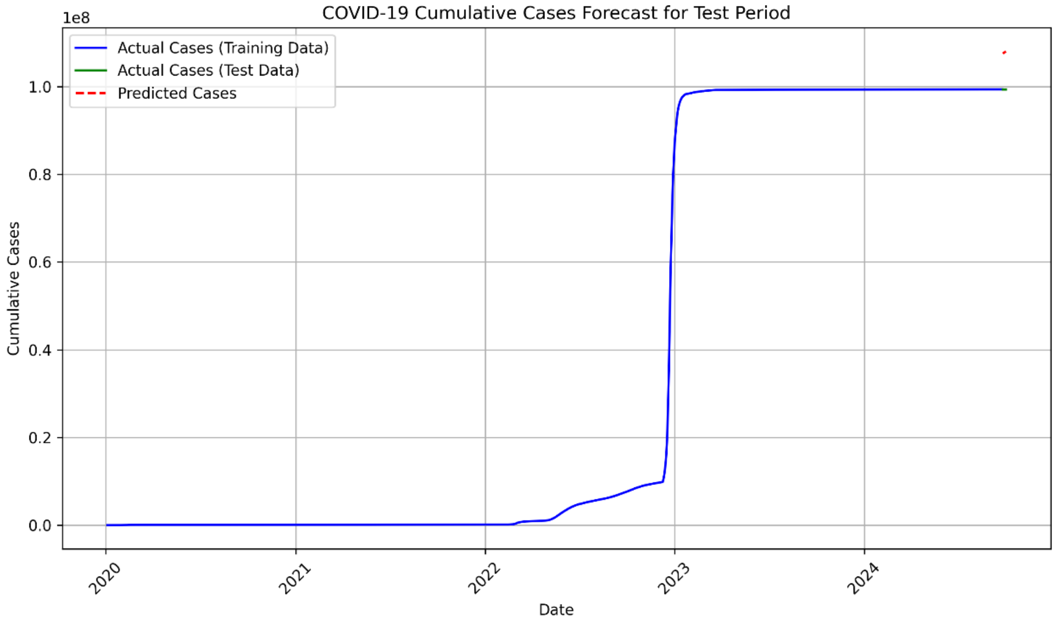

Figure 11： Forecast for 7 days on Lass from 2020

4. Forecast for 10 days: Real data end date: 2024-09-20. Dates included in the forecast: 2024-09-21 to 2024-09-30. Mean Absolute error (MAE) is 8708268.21. Relative error (RE) is 8.76%. The specific diagram is as follows 图 12

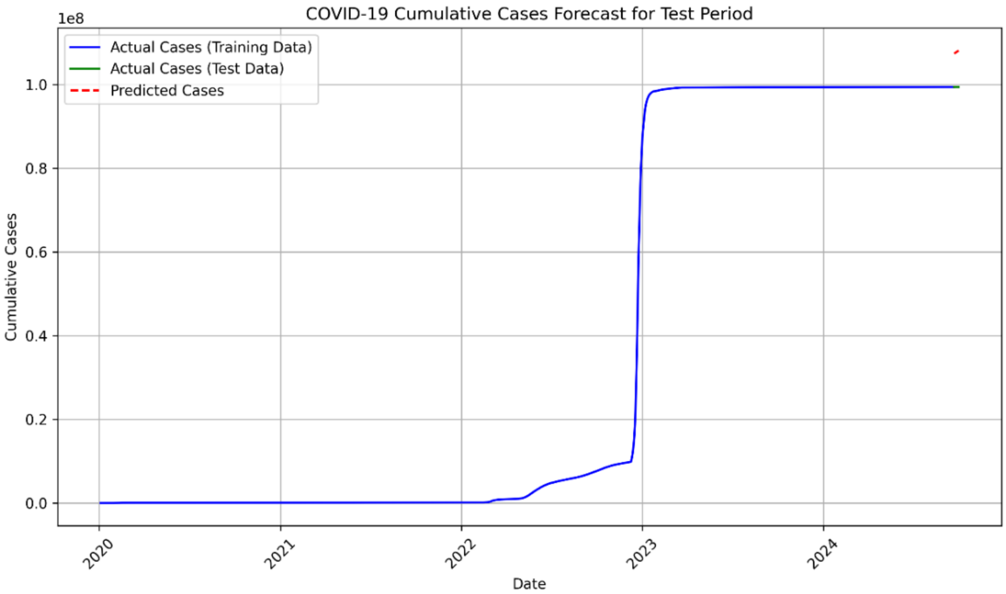

Figure 12： Forecast for 10 days on Lasso from 2020

5. Forecast for 14 days: Real data end date: 2024-09-26. Dates included in the forecast: 2024-09-17 to 2024-09-30. Mean Absolute error (MAE) is 8781684.2. Relative error (RE) is 8.84%. The specific diagram is as follows 图 13

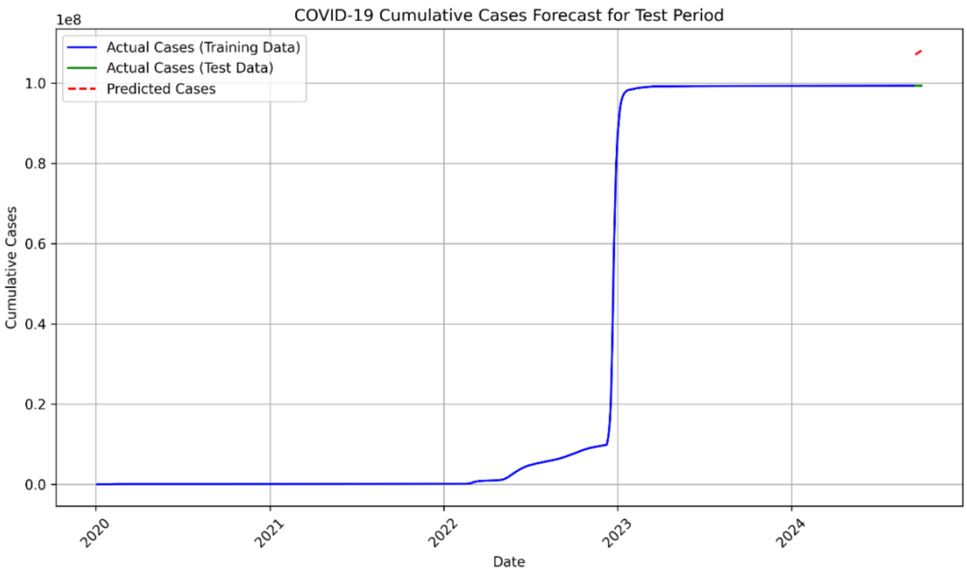

Figure 13： Forecast for 14 days on Lasso from 2020

6. Forecast for 21 days: Real data end date: 2024-09-09. Dates included in the forecast: 2024-09-10 to 2024-09-30. Mean Absolute error (MAE) is 8905424.0. Relative error (RE) is 8.96%. The specific diagram is as follows图 14

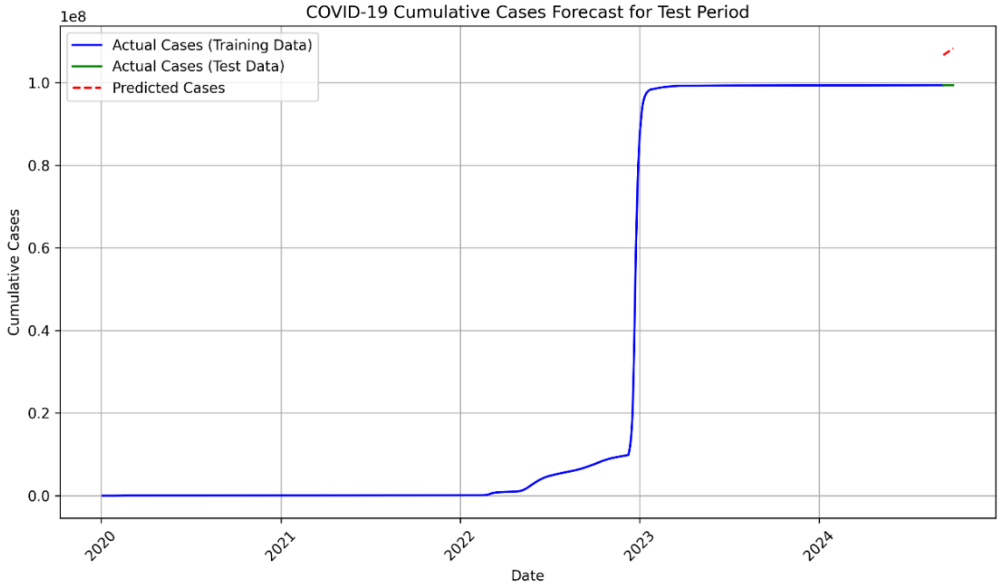

Figure 14： Forecast for 21 days on Lasso from 2020

7. Forecast for 30 days: Real data end date: 2024-08-31. Dates included in the forecast: 2024-09-01 to 2024-09-30. Mean Absolute error (MAE) is 9055299.820. Relative error (RE) is 9.11%. The specific diagram is as follows 图 15.

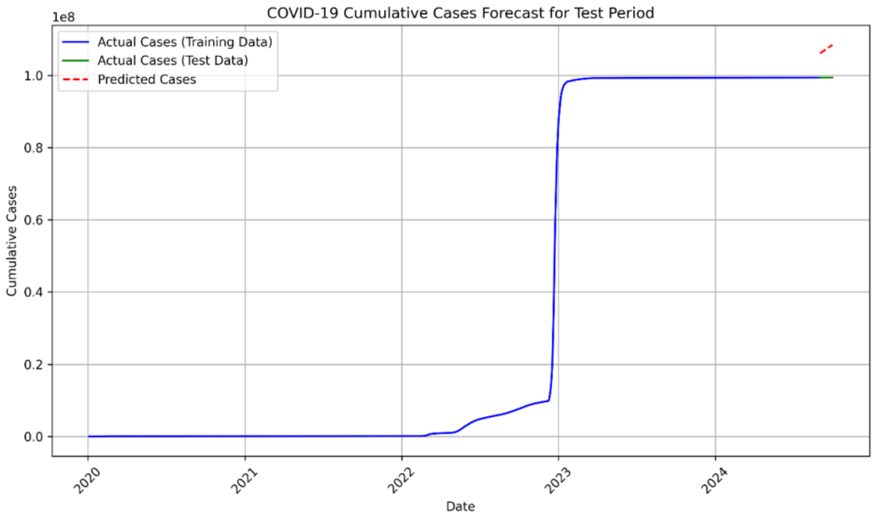

Figure 15： Forecast for 30 days on Lasso from 2020

Table 3: Summary of Lasso From 2020

Horizon	MAE	RE
3 days	8575221.91	8.63%
5 days	8613817.86	8.67%
7 days	8651950.19	8.76%
10 days	8708268.21	8.84%
14 days	8781684.2	8.96%
21 days	8905424.0	8.96%
30 days	9055299.820	9.11%

From Table 3, we can summarize the Mean Absolute Error (MAE) and Relative Error (RE) of the Lasso model for different prediction horizons in 2020. The data indicates that as the prediction horizon increases, both MAE and RE also increase, suggesting that the prediction error of the Lasso model grows as the forecast period extends. Specifically, the 3-day forecast has the smallest MAE and RE, while the 30-day forecast has the largest MAE and RE.

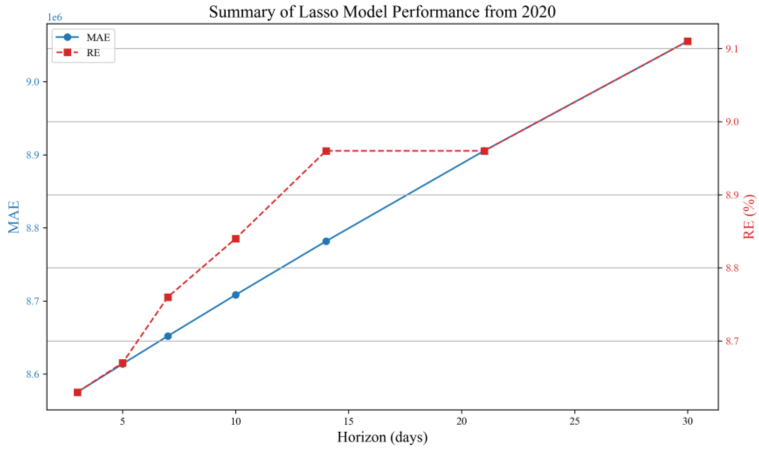

Figure 16: MAE and RE as a Function of Prediction Horizon.

This trend is expected because longer-term forecasts typically involve more uncertainty and variability, making accurate predictions more difficult. The increase in MAE indicates that the average absolute difference between the predicted and actual values is growing, while the rise in RE shows that the error percentage relative to the actual values is also increasing.

4.2.2 From 2024 year

In this section, the input data consists only of the data from 2024, and the specific prediction results are as follows:

1. Forecast for 3 days:- Real data end date: 2024-09-27. Dates included in the forecast: 2024-09-28 to 2024-09-30. Mean Absolute error (MAE) is 5174.52. Relative error (RE) is 0.01%. The specific diagram is as follows .

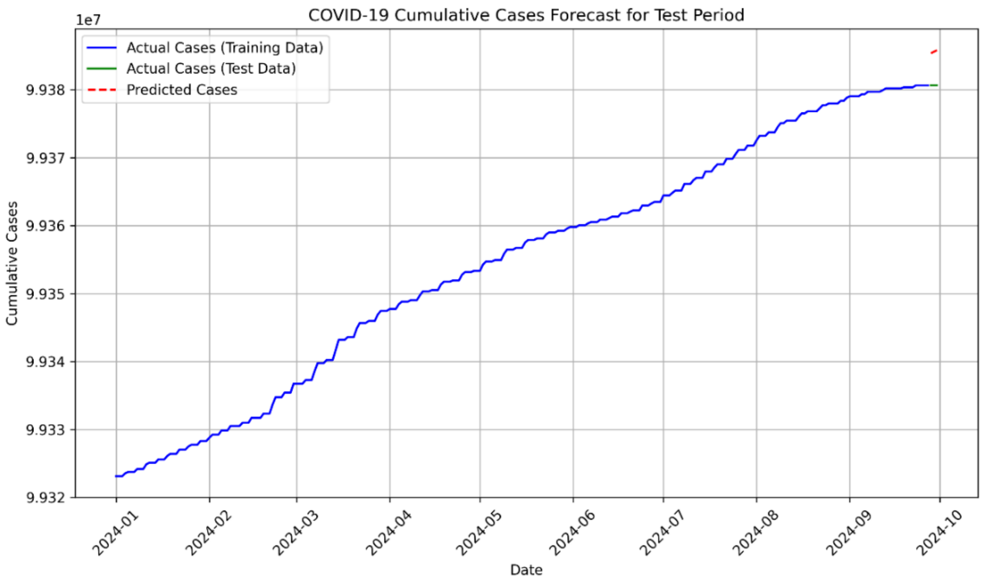

Figure 17: Forecast for 3 days on Lasso from 2024

2. Forecast for 5 days: Real data end date: 2024-09-25. Dates included in the forecast: 2024-09-26 to 2024-09-30. Mean Absolute error (MAE) is 5308.9. Relative error (RE) is 0.01%. The specific diagram is as follows

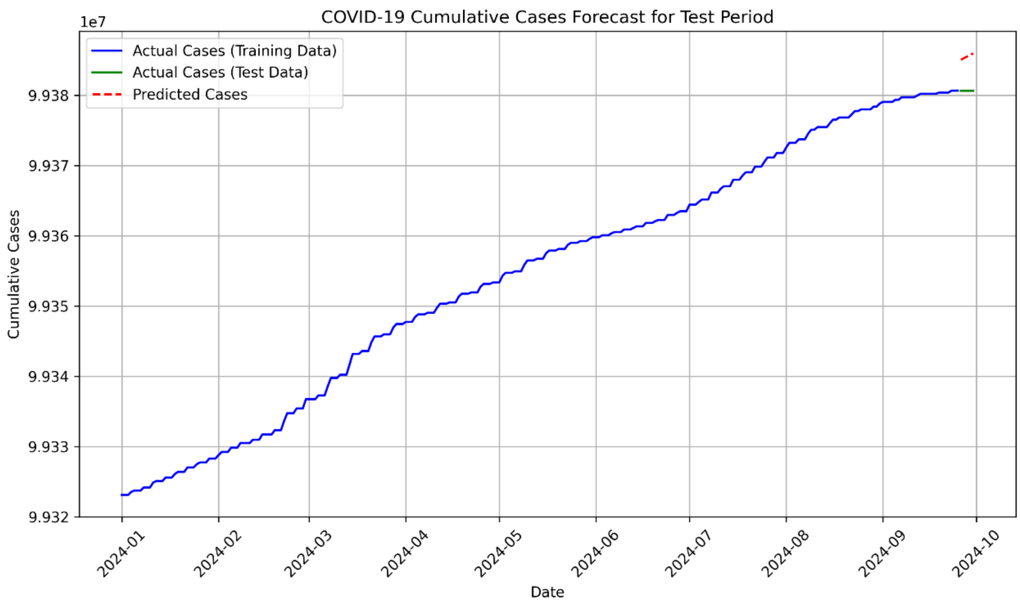

Figure 18: Forecast for 5 days on Lasso from 2024

3. Forecast for 7 days: Real data end date: 2024-09-25. Dates included in the forecast: 2024-09-26 to 2024-09-30. Mean Absolute error (MAE) is 5435.72. Relative error (RE) is 0.01%.

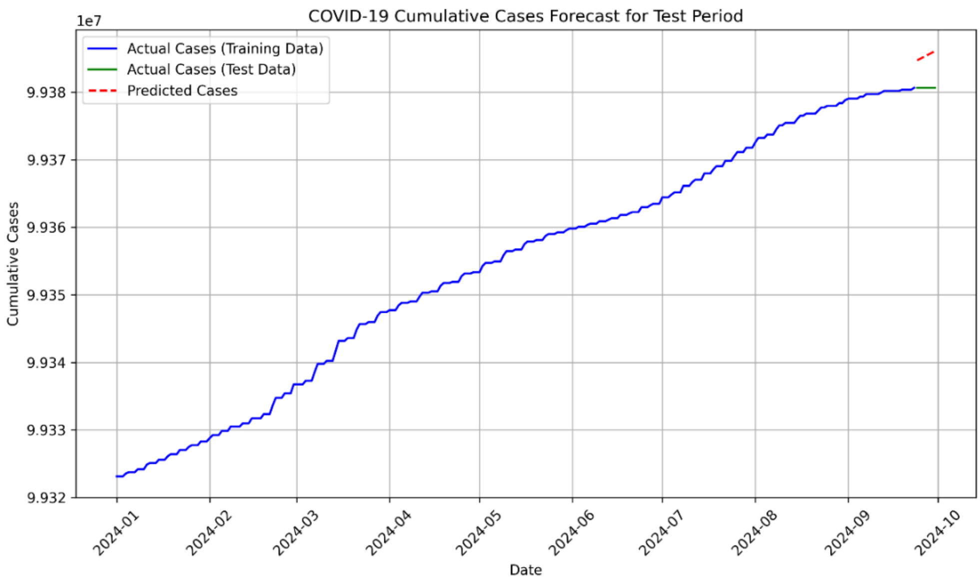

Figure 19: Forecast for 7 days on Lasso from 2024

4. Forecast for 10 days: Real data end date: 2024-09-20. Dates included in the forecast: 2024-09-21 to 2024-09-30. Mean Absolute error (MAE) is 5619.37. Relative error (RE) is 0.01%.

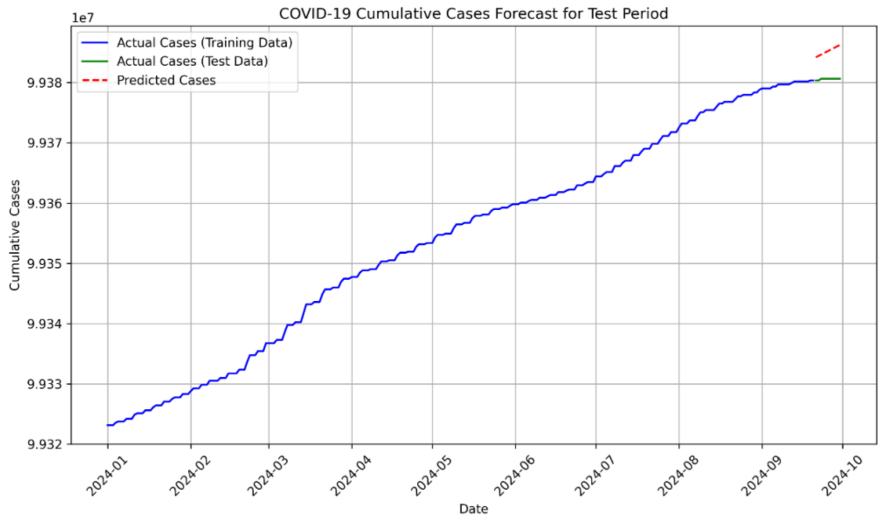

Figure 20: Forecast for 10 days on Lasso from 2024

5. Forecast for 14 days: Real data end date: 2024-09-16. Dates included in the forecast: 2024-09-17 to 2024-09-30. Mean Absolute error (MAE) is 5844.47. Relative error (RE) is 0.01%.

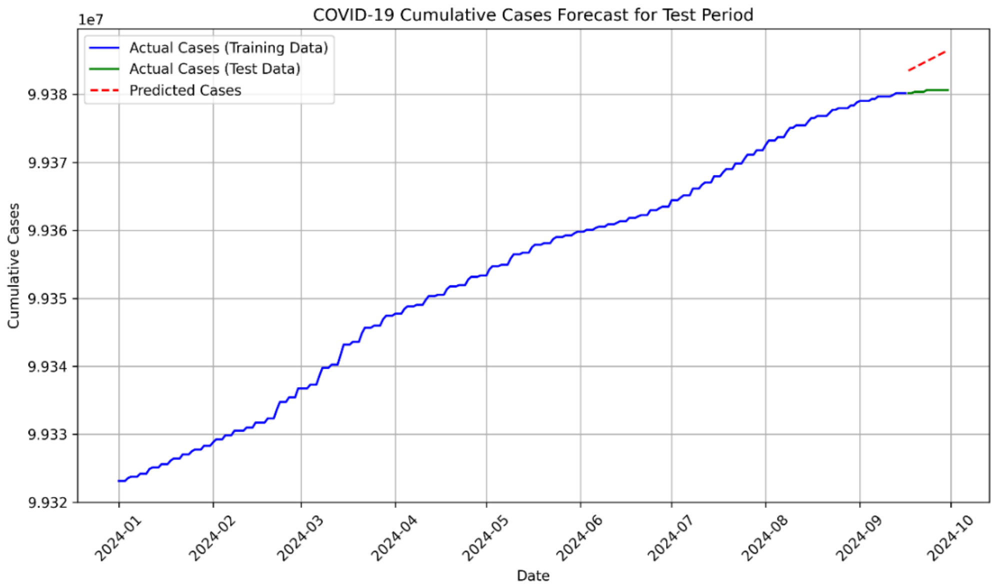

Figure 21:Forecast for 14 days on Lasso from 2024

6. Forecast for 21 days: Real data end date: 2024-09-09. Dates included in the forecast: 2024-09-10 to 2024-09-30. Mean Absolute error (MAE) is 6169.5. Relative error (RE) is 0.01%.

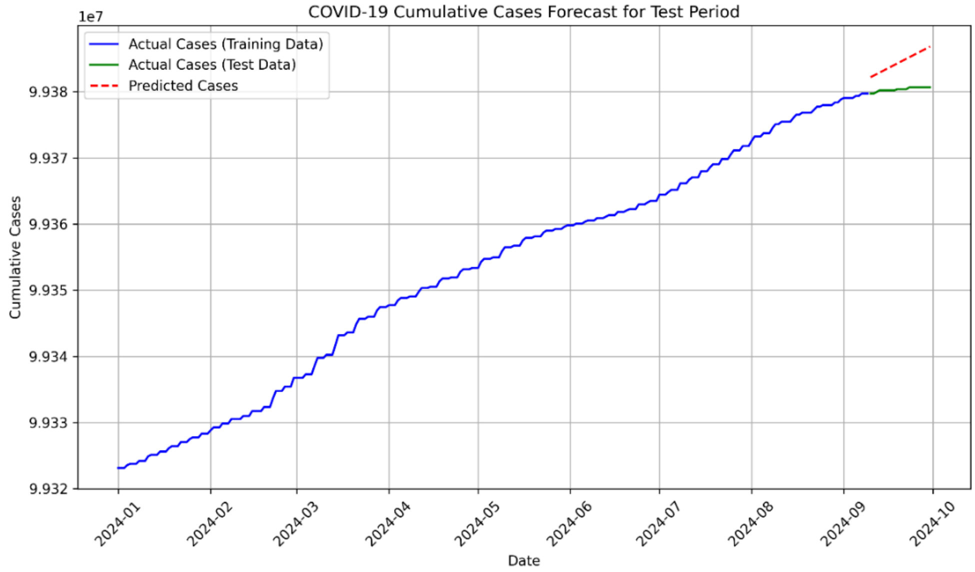

Figure 22: Forecast for 21 days on Lasso from 2024

7. Forecast for 30 days: Real data end date: 2024-08-31. Dates included in the forecast: 2024-09-01 to 2024-09-30. Mean Absolute error (MAE) is 6474.06. Relative error (RE) is 0.01%.

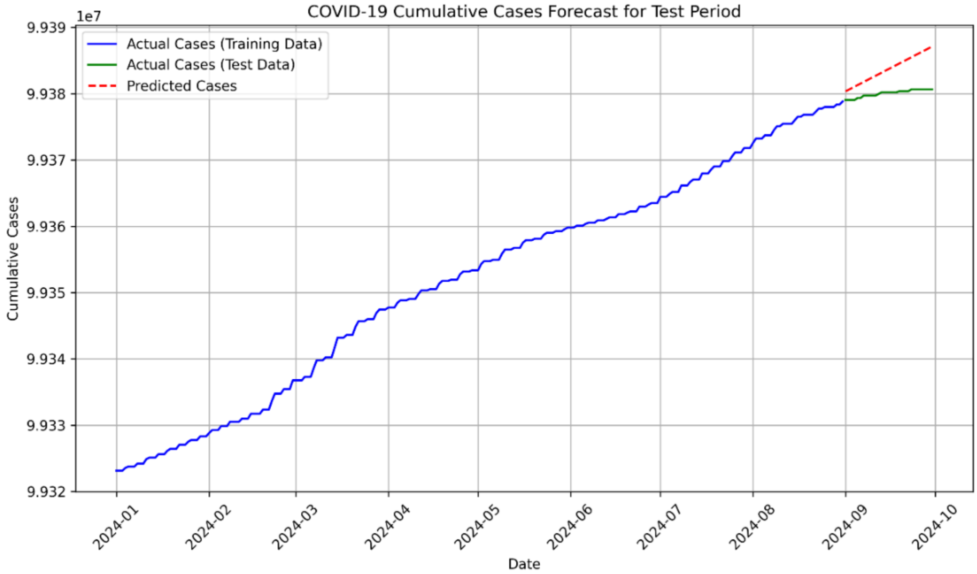

Figure 23: Forecast for 30 days on Lasso from 2024

Table 4: Summary of Lasso From 2024

Horizon	MAE	RE
3 days	5172.52	0.01%
5 days	5308.9	0.01%
7 days	5435.72.	0.01%
10 days	5619.37	0.01%
14 days	5844.47	0.01%
21 days	6169.5	0.01%
30 days	6474.06	0.01%

As shown in Table 4, with the increase in the prediction horizon, the MAE also increases, but the RE remains unchanged at 0.01%. This indicates that although the absolute error increases as the prediction time extends, the proportion of the error relative to the actual values remains constant.

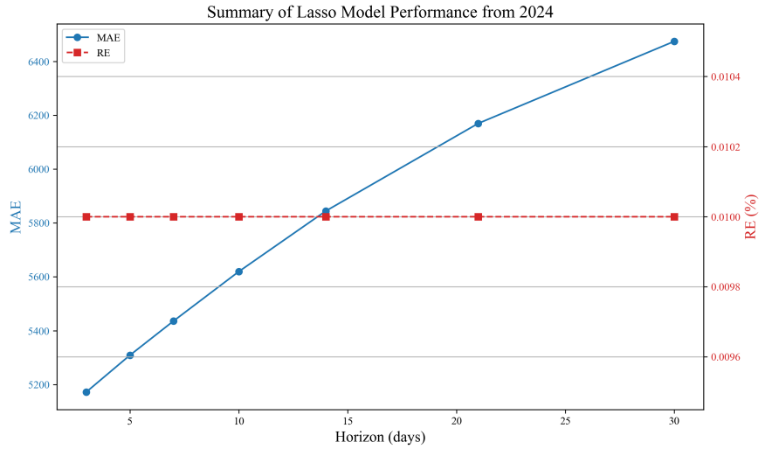

Figure 24: MAE and RE Variation with Prediction Length

Figure 24 summarizes the performance of the Lasso model in 2024, showing that as the prediction horizon increases from 3 days to 30 days, the Mean Absolute Error (MAE) exhibits a linear growth trend, while the Relative Error (RE) remains stable at around 0.01%. This indicates that although the model's absolute prediction error increases with the extension of the forecast period, the error proportion relative to the actual values remains constant, demonstrating the model's stability across different time horizons.

The model's error is the lowest when I use the 2024 datasets to predict just 3 days, suggesting that the Lasso regression model is particularly effective for short-term forecasts. This optimal performance over a limited forecast horizon highlights the importance of recent data in achieving accurate predictive results, while longer periods may introduce too much variability and uncertainty that reduce predictive accuracy

4.3 Prediction of Cumulative Infections Based on LSTM

Although traditional Lasso regression can also predict the future cumulative number of infections, the error gradually increases as the prediction horizon expands, leading to a decrease in performance. Meanwhile, both the MAE and RE of the Lasso model are relatively large, making it unsuitable for accurate applications.

Therefore, this study uses the deep learning LSTM model for predicting cumulative infections. Compared to the Lasso model, the LSTM model is capable of extracting trend features from the time series and automatically quantifying these features, which can then be used to predict the number of infections for a given future time horizon.

Table 5 LSTM Network Architecture Diagram

Network Layer Name	Network Layer Shape	Meaning
InputLayer	(None,LookBack,1)	Input as historical LookBack time steps
Dense	(None,LookBack,64)	Non-linear Mapping
LSTM	(None,32)	LSTM Features
Dense	(None,Horizon)	Prediction Layer

In Table 5, the input layer receives historical data, where "None" represents any batch size, LookBack indicates the number of historical time steps considered, and "1" means each time step has one feature, i.e., the number of infections. Next is a fully connected layer that maps the input data into a 3D tensor with LookBack time steps and 64 neurons. This is followed by an LSTM layer, a long short-term memory network layer specifically designed to capture long-term dependencies in time series data, with an output feature size of 32. Finally, there is a Dense prediction layer that maps the output of the LSTM layer to the predicted results, where Horizon represents the number of future time points the model needs to forecast. The overall design of the model enables it to effectively process and predict time series data.

4.3.1 From 2020 year

Similarly, the model is first trained using all the data from 2020 to 2024. Due to the large volume of data, which includes many changing trends and spans a long period, factors like non-stationarity may exist.

1. For the LSTM prediction with a length of 3 days, the RMSE is 1,317,720.23, the MAPE is 1.2%, and the MAE is 1,222,682. The comparison between the predicted and actual values is shown in Figures 25 and 26.

1. 。

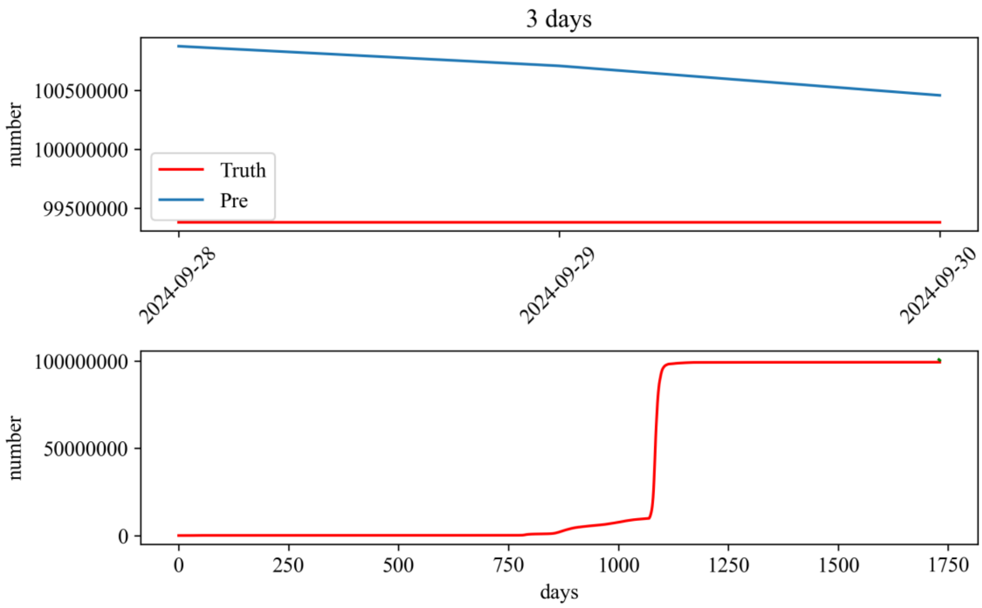

Figure 25: Forecast for 3 days on LSTM from 2020

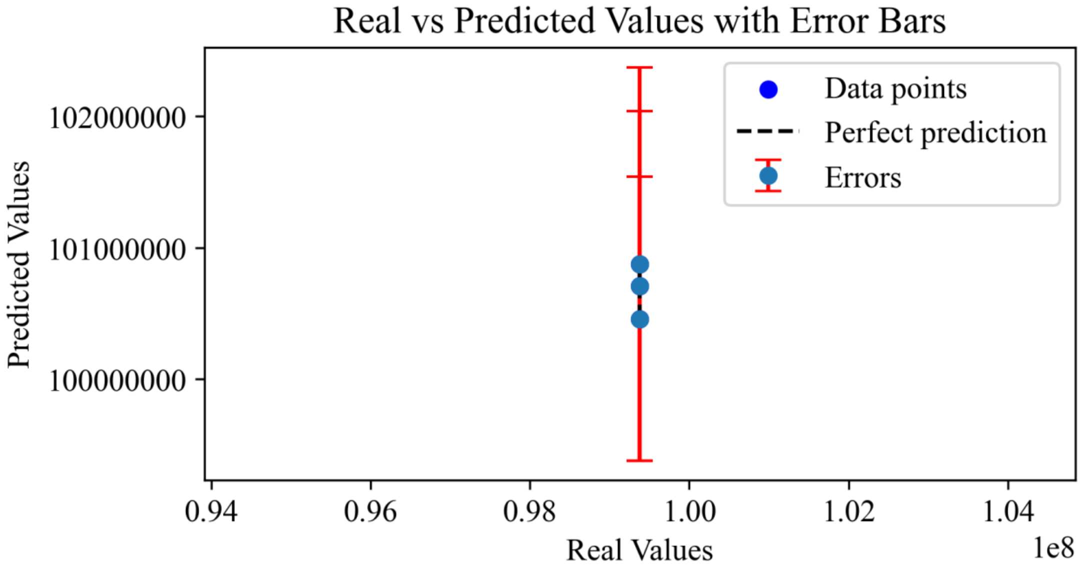

Figure 26: Error scatter for 3 days on LSTM from 2020

2. For the LSTM prediction with a length of 5 days, the RMSE is 3,518,784.26, the MAPE is 3.2%, and the MAE is 3,307,105.2. The comparison between the predicted and actual values is shown in Figures 27 and 28.

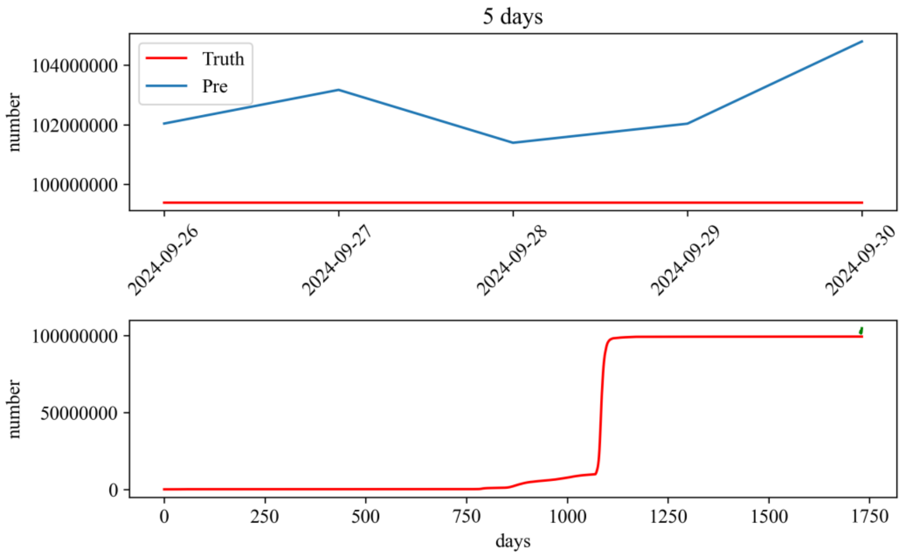

Figure 27:Forecast for 5 days on LSTM from 2020

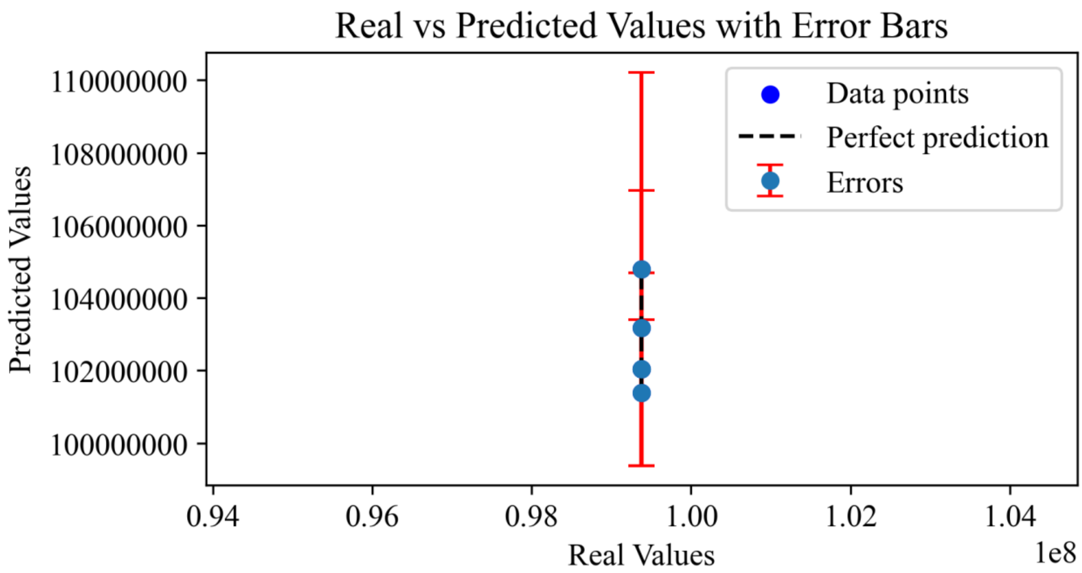

Figure 28: Error scatter for 5 days on LSTM from 2020

3. For the LSTM prediction with a length of 7 days, the RMSE is 3,284,963.90, the MAPE is 3.27%, and the MAE is 3,140,938.28. The comparison between the predicted and actual values is shown in Figures 29 and 30.

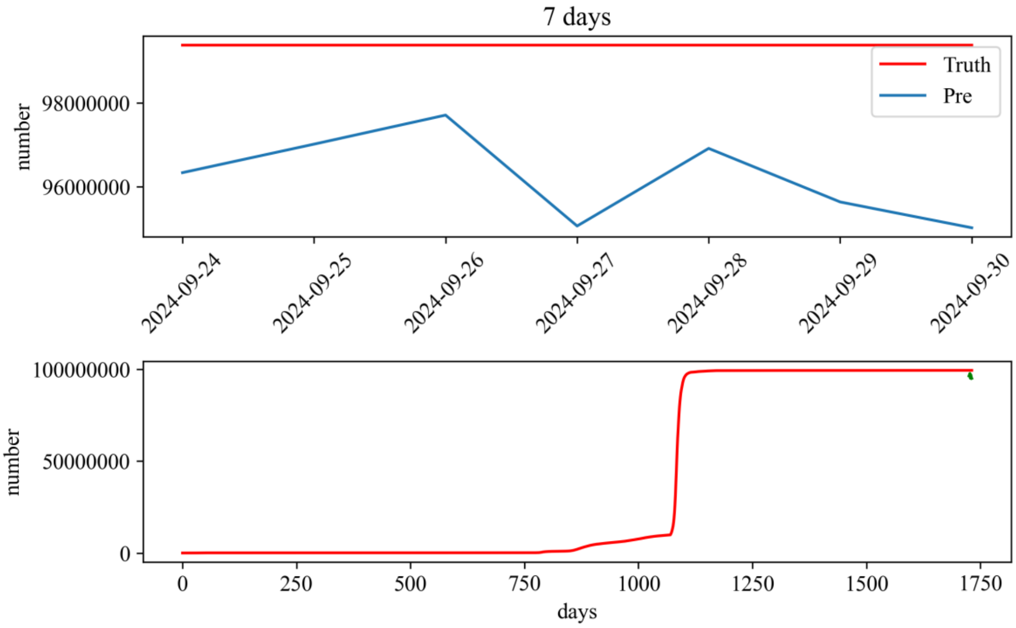

Figure 29: Forecast for 7 days on LSTM from 2020

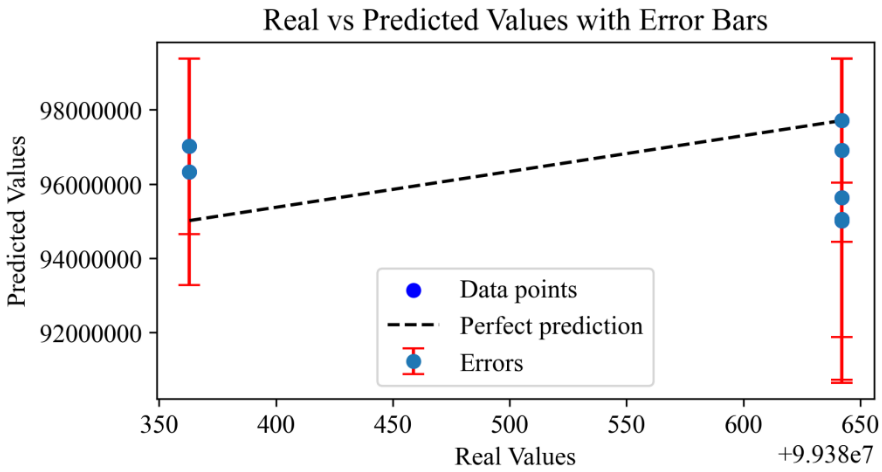

Figure 30: Error scatter for 7 days on LSTM from 2020

4. For the LSTM prediction with a length of 10 days, the RMSE is 28,015,500, the MAPE is 2.7%, and the MAE is 2,619,542.4. The comparison between the predicted and actual values is shown in Figures 31 and 32.

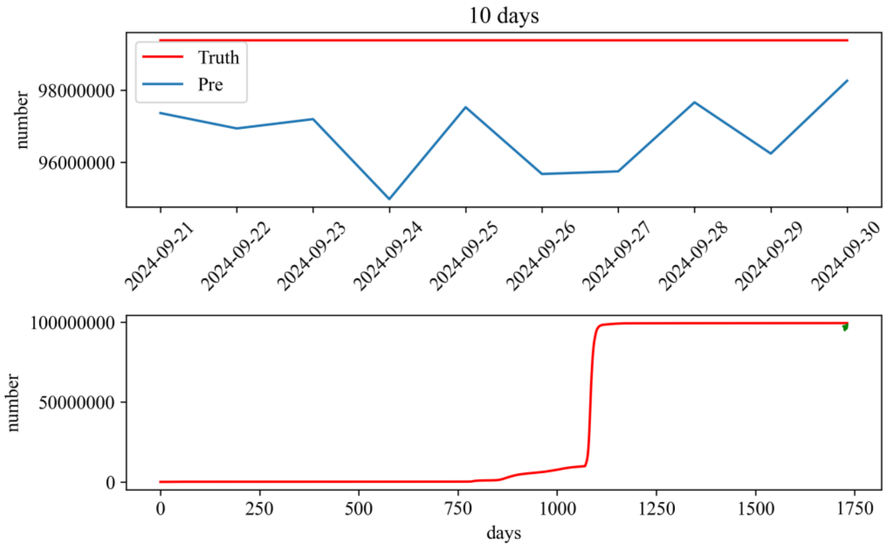

Figure 31: Forecast for 10 days on LSTM from 2020

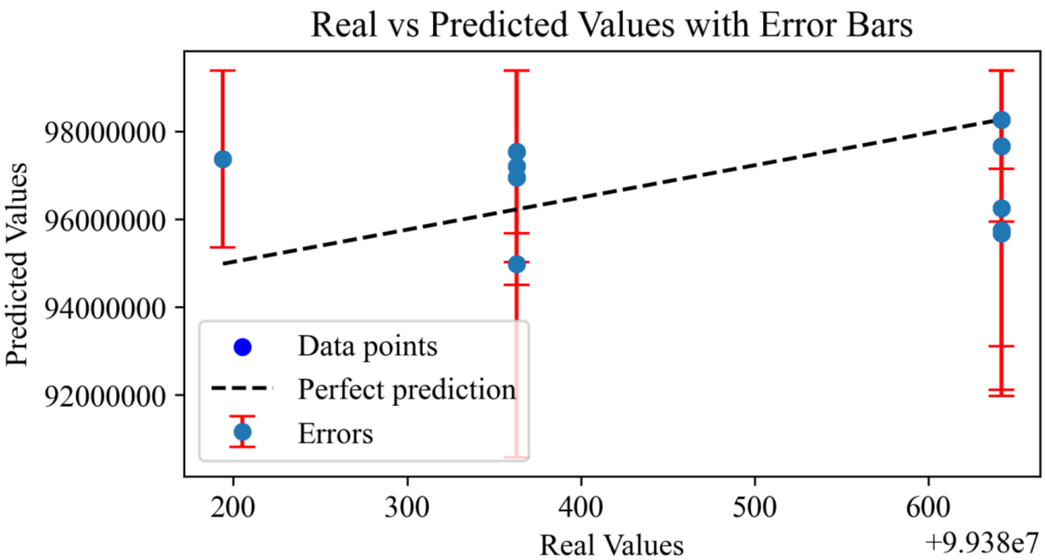

Figure 32: Error scatter for 10 days on LSTM from 2020

5. For the LSTM prediction with a length of 14 days, the RMSE is 1,126,747.26, the MAPE is 0.9%, and the MAE is 3,260,368.28. The comparison between the predicted and actual values is shown in Figures 33 and 34.

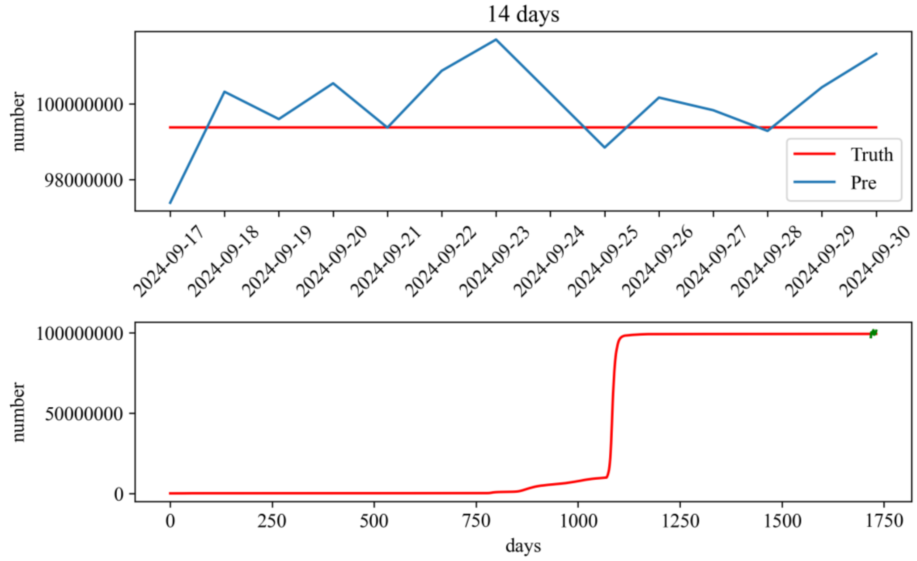

Figure 33: Forecast for 14 days on LSTM from 2020

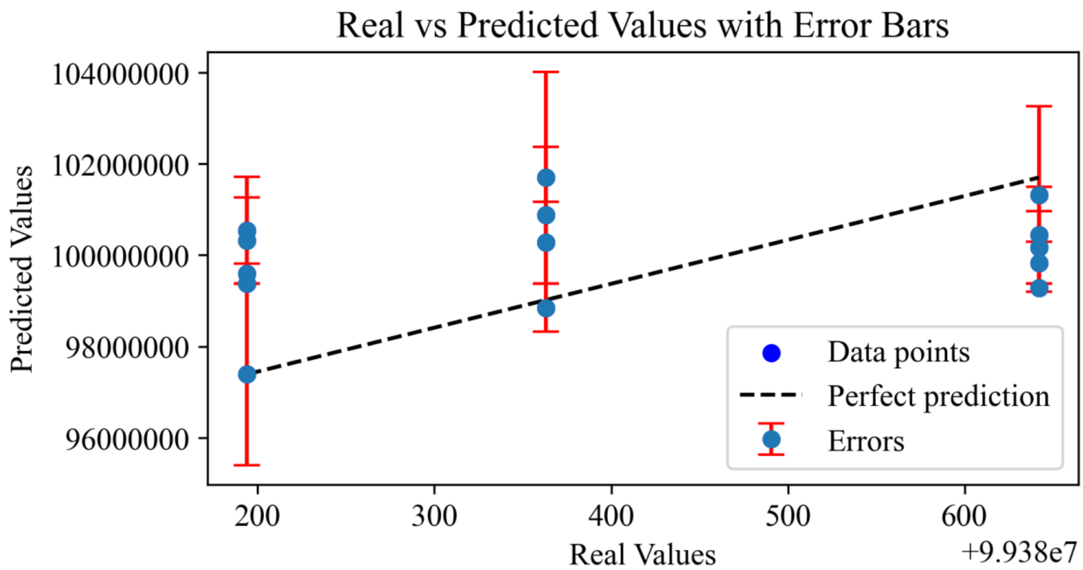

Figure 34: Error scatter for 14 days on LSTM from 2020

6. For the LSTM prediction with a length of 21 days, the RMSE is 1,216,357.26, the MAPE is 0.11%, and the MAE is 993,624.71. The comparison between the predicted and actual values is shown in Figures 35 and 36.

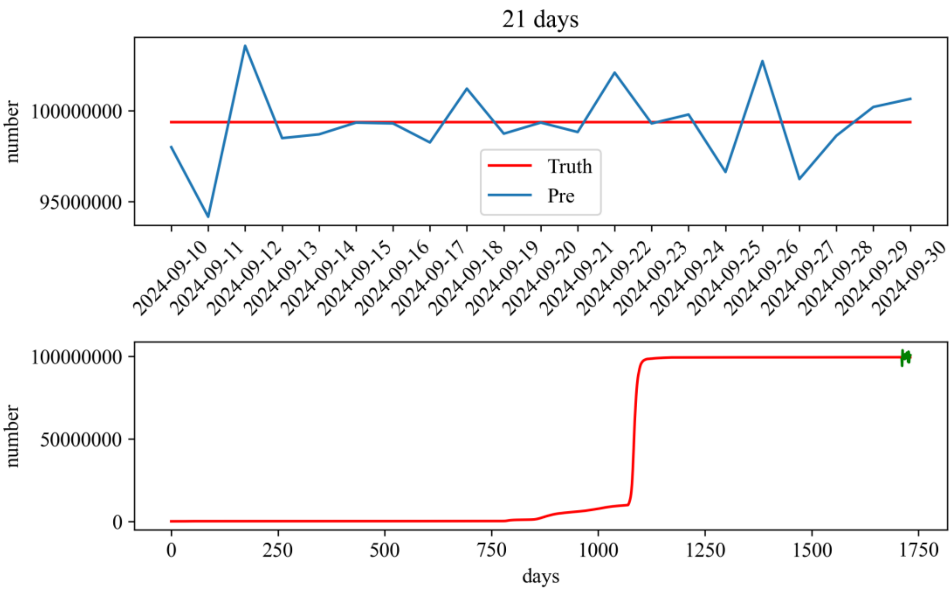

Figure 35: Forecast for 21 days on LSTM from 2020

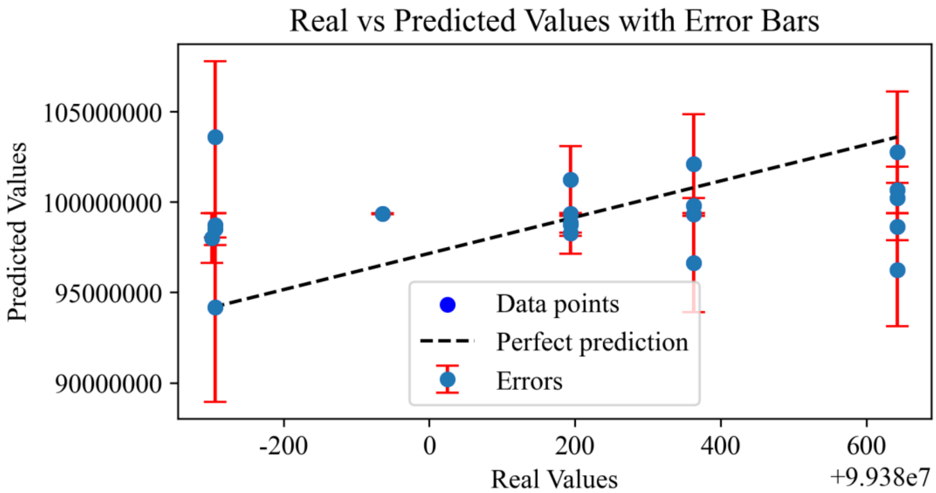

Figure 36: Error scatter for 21 days on LSTM from 2020

7. For the LSTM prediction with a length of 30 days, the RMSE is 1,962,509.347, the MAPE is 1.6%, and the MAE is 1,586,796.86. The comparison between the predicted and actual values is shown in Figures 37 and 38.

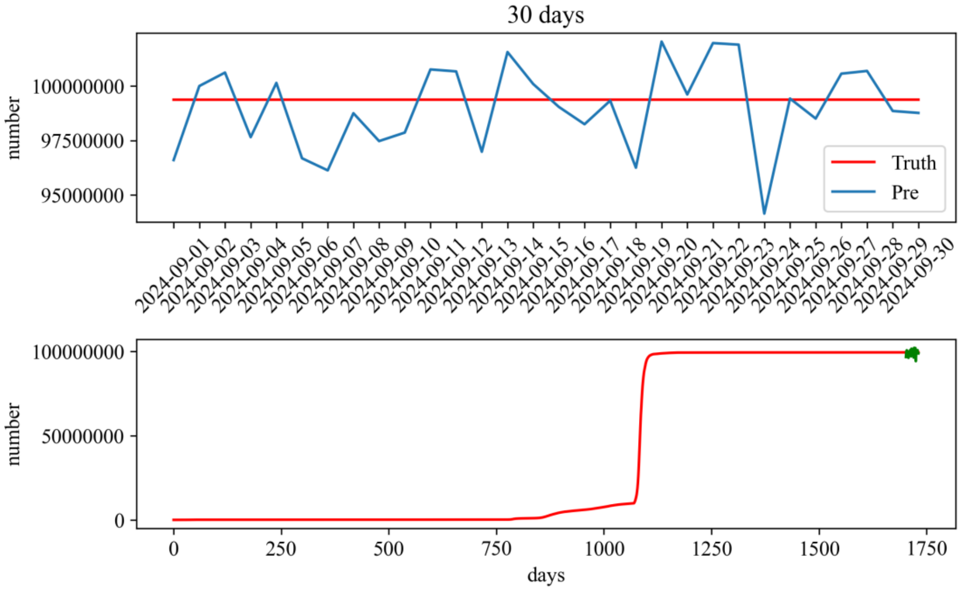

Figure 37: Forecast for30 days on LSTM from 2020

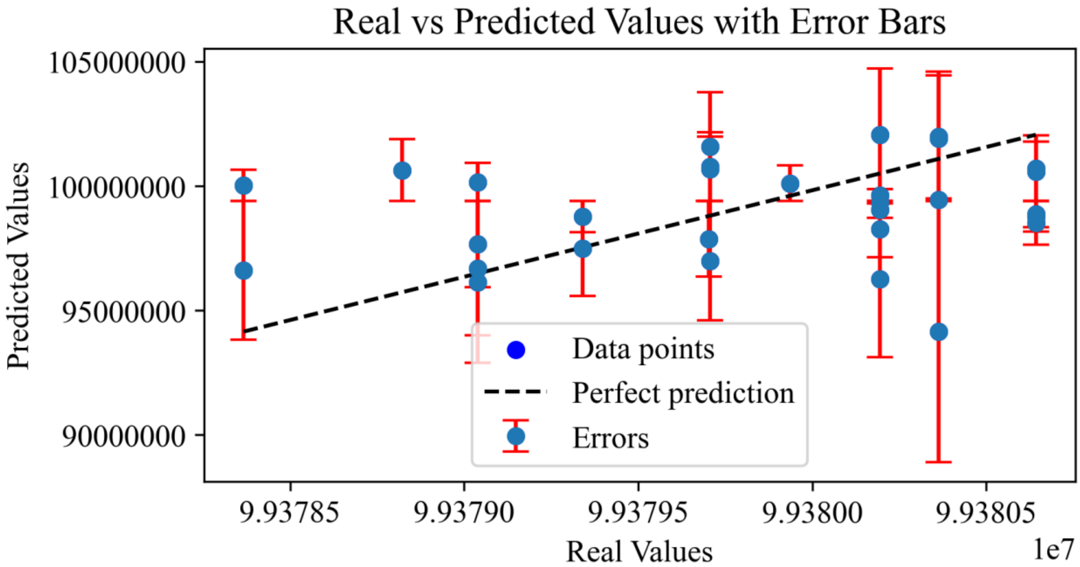

Figure 38: Error scatter for 30 days on LSTM from 2020

Table 6 Summary of LSTM From 2020

Horizon	MAE	RE
3 days	3307105.52	1.2%
5 days	3260368.28	3.2%
7 days	3140938.28	3.27%
10 days	2619542.4	2.7%
14 days	984467.71	0.9%
21 days	993624.71	0.11%
30 days	1586796.86	1.6%

In Table 6, the MAE gradually decreases, indicating that the LSTM model achieves higher prediction accuracy in the short term, with the error decreasing over time. For example, the MAE for 3 days is 3,307,105.52, while for 30 days, the MAE drops to 1,586,796.86. The relative error (RE) is higher in the short term, with the RE for 5 days at 3.2%, while the RE for the longer 30-day forecast decreases to 1.6%. This may suggest that there are larger fluctuations in the market in the short term, making predictions more difficult.

Regarding long-term predictive capabilities, the results for the 21-day and 30-day predictions show relatively low MAE and RE values, which may indicate that the LSTM model has a certain advantage in long-term forecasting, especially in the 30-day forecast where the RE is only 0.11%, demonstrating the model's high accuracy in long-term predictions.

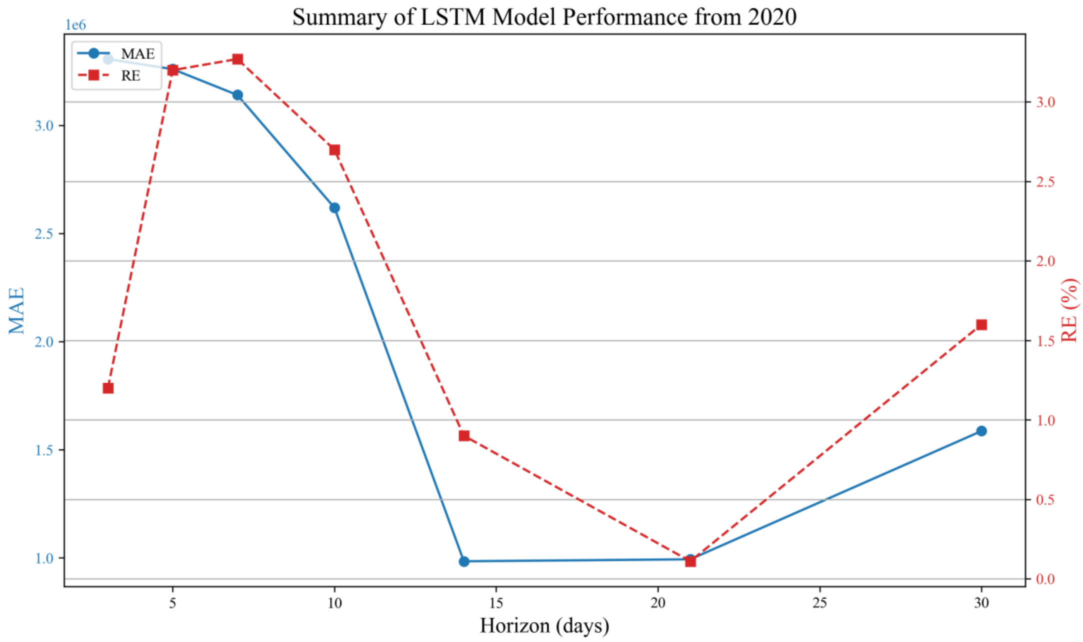

Figure 39: LSTM Model MAE and RE Trend with Varying Prediction Lengths.

Figure 39 shows the trend of MAE and RE for different forecast horizons. It can be observed that the values fluctuate significantly, which may be closely related to seasonality or periodic patterns.

4.3.2 From 2024 year

Then, the model was trained using all the data from 2024. Due to the shorter time span, the model may be less sensitive to factors like non-stationarity.

1. For the LSTM model with a prediction length of 3 days, the RMSE is 1311.45, MAPE is 0.002%, and MAE is 1220. The comparison between the predicted and actual values is shown in Figure 40 and Figure 41.

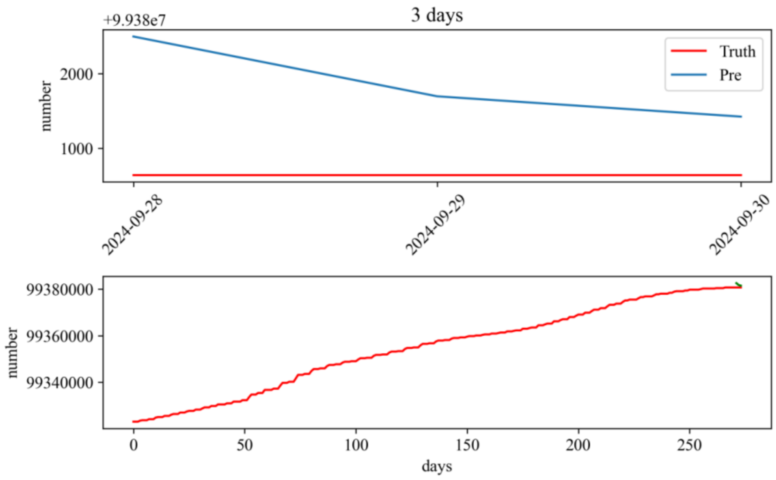

Figure 40: Forecast for 3 days on LSTM from 2024

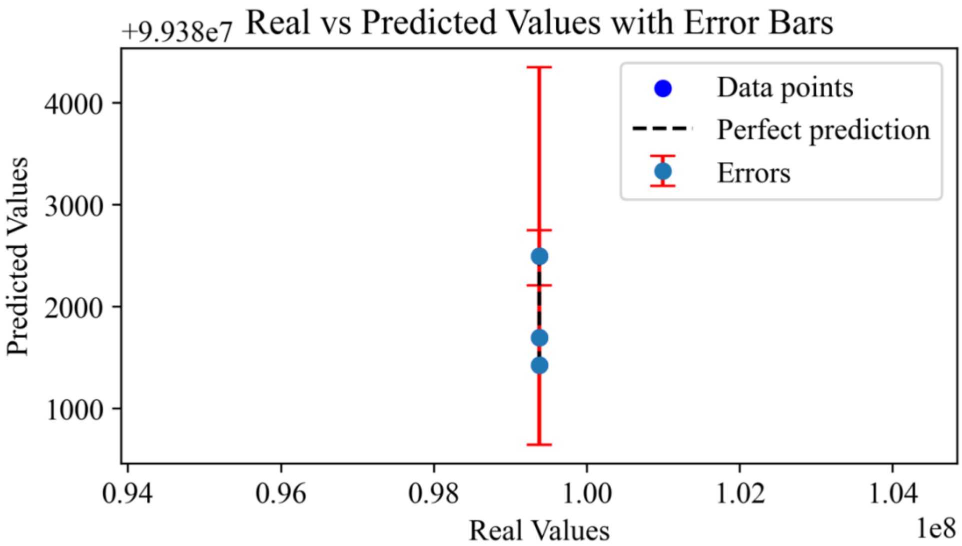

Figure 41: Error scatter for 3 days on LSTM from 2024

2. For the LSTM model with a prediction length of 5 days, the RMSE is 664.32, MAPE is 0.0002%, and MAE is 565.2. The comparison between the predicted and actual values is shown in Figure 42 and Figure 43.

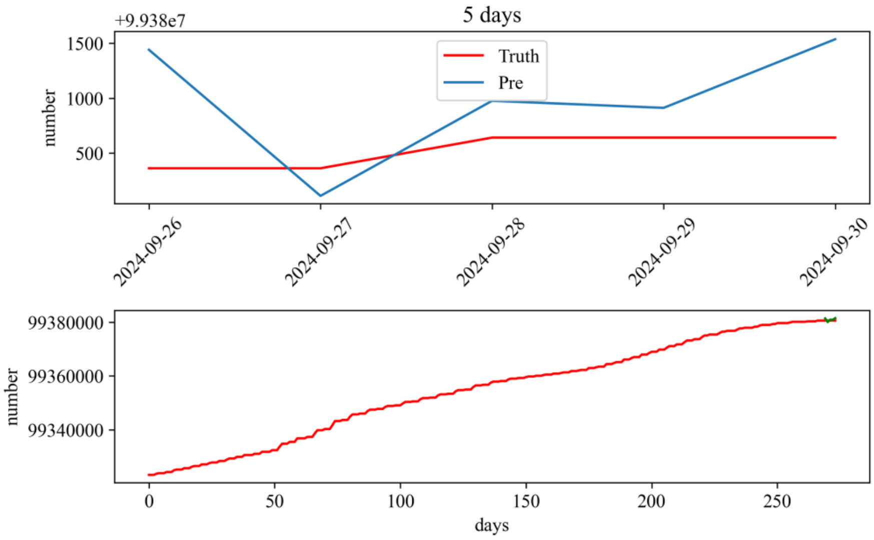

Figure 42: Forecast for 5 days on LSTM from 2024

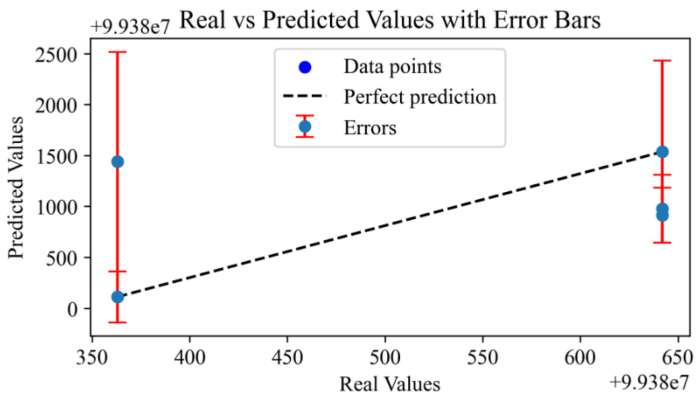

Figure 43: Error scatter for 5 days on LSTM from 2024

3. For the LSTM model with a prediction length of 7 days, the RMSE is 863.29, MAPE is 0.0001%, and MAE is 719.7. The comparison between the predicted and actual values is shown in Figure 44 and Figure 45.

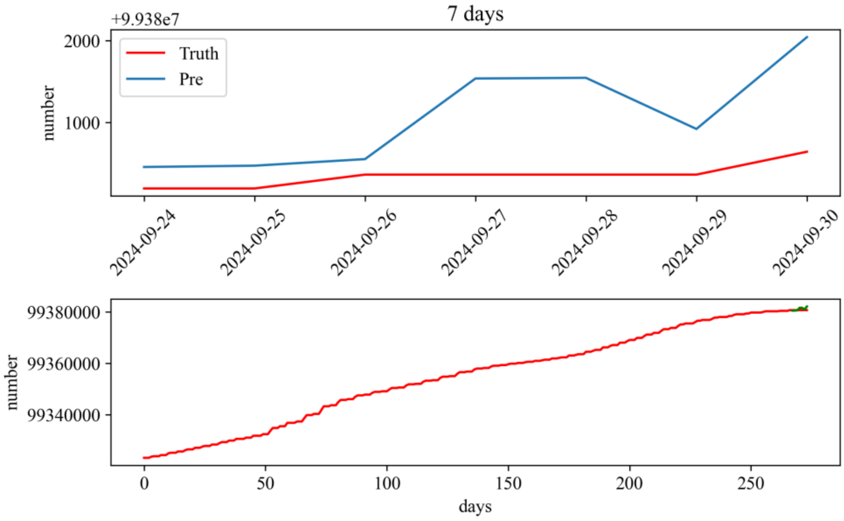

Figure 44: Forecast for 7 days on LSTM from 2024

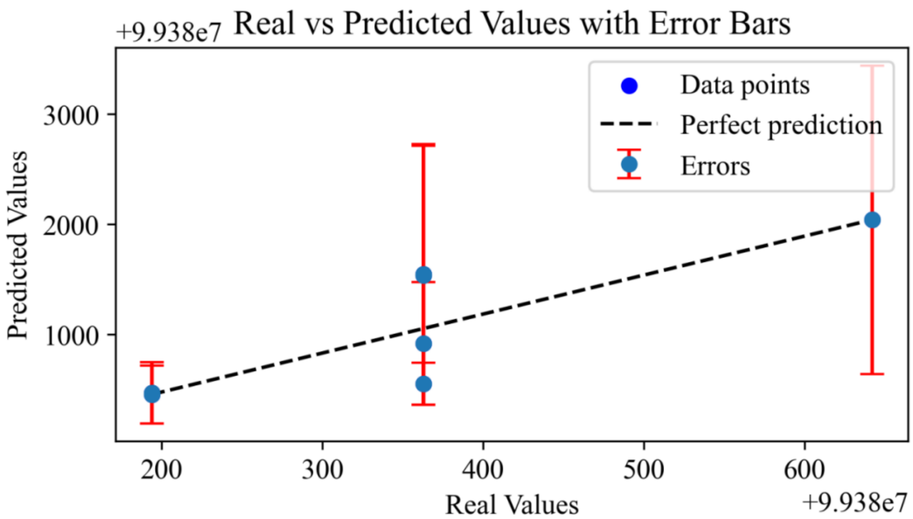

Figure 45: Error scatter for 7 days on LSTM from 2024

4. For the LSTM model with a prediction length of 10 days, the RMSE is 1600.32, MAPE is 0.001%, and MAE is 1576.4. The comparison between the predicted and actual values is shown in Figure 46 and Figure 48.

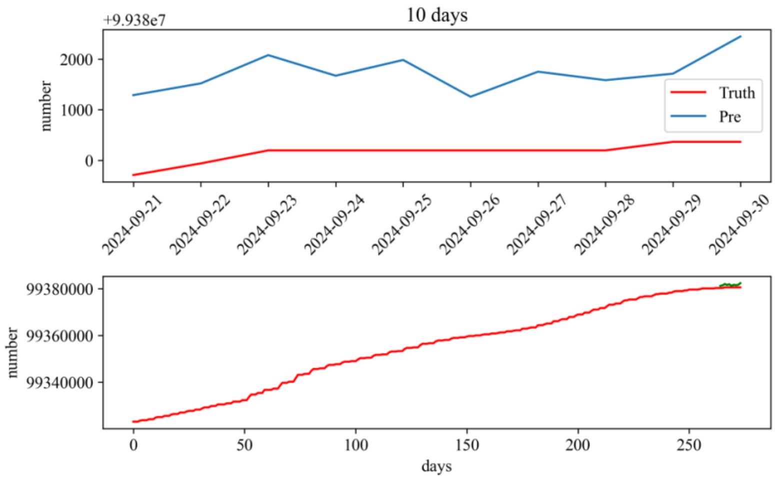

Figure 46: Forecast for 10 days on LSTM from 2024

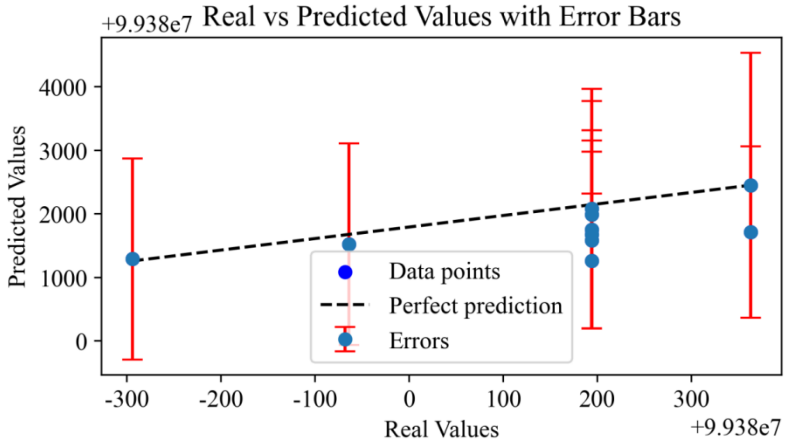

Figure 47: Error scatter for 10 days on LSTM from 2024

5. For the LSTM model with a prediction length of 14 days, the RMSE is 1170.07, MAPE is 0.0002%, and MAE is 1076.4. The comparison between the predicted and actual values is shown in Figure 48 and Figure 49.

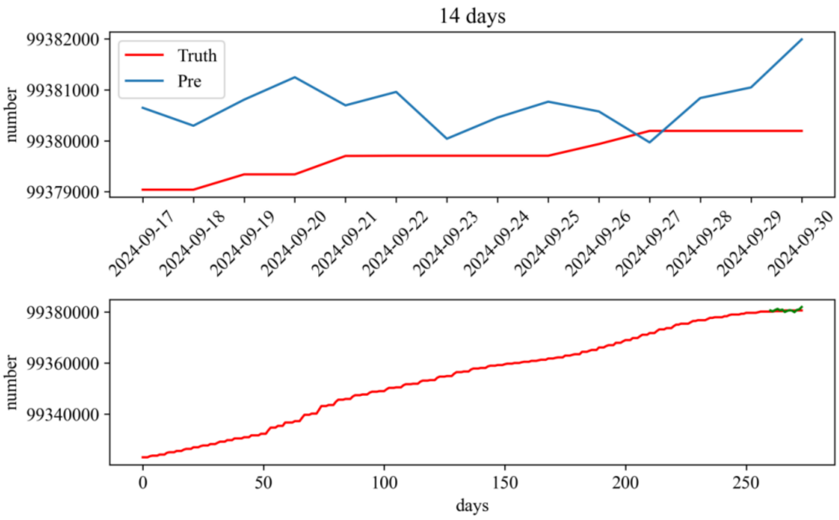

Figure 48: Forecast for 14 days on LSTM from 2024

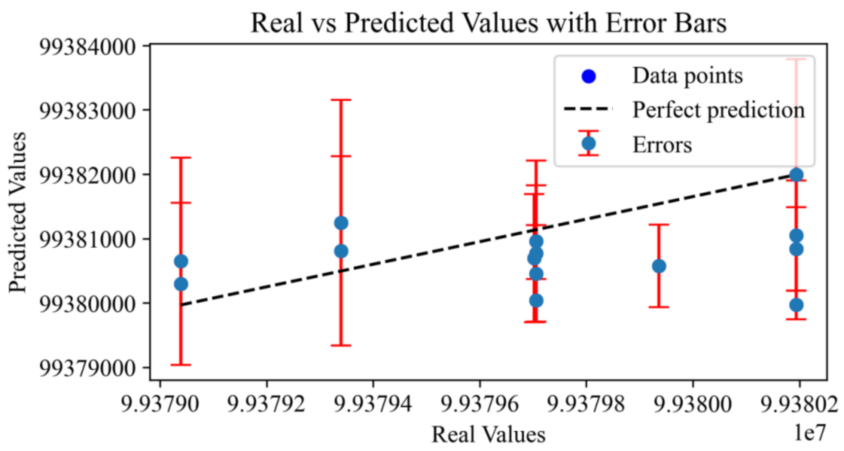

Figure 49: Error scatter for 14 days on LSTM from 2024

6. For the LSTM model with a prediction length of 21 days, the RMSE is 2802.6, MAPE is 0.0002%, and MAE is 2740.0. The comparison between the predicted and actual values is shown in Figure 50 and Figure 51.

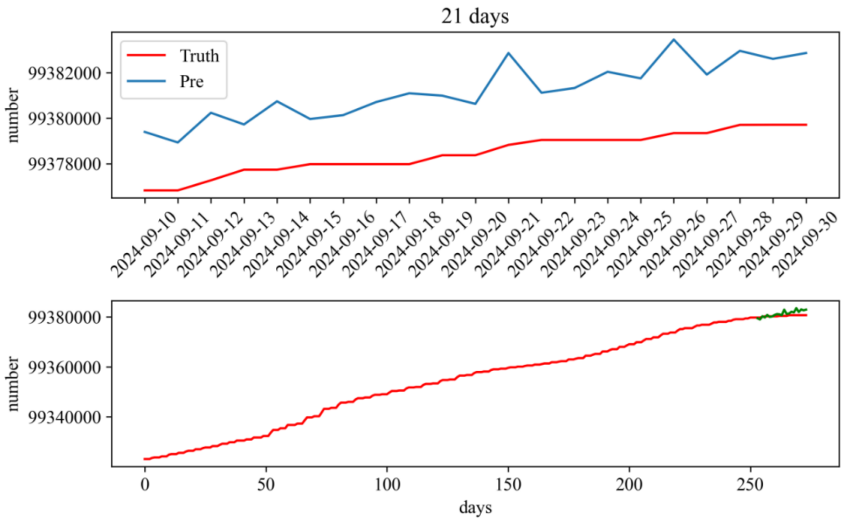

Figure 50: Forecast for 21 days on LSTM from 2024

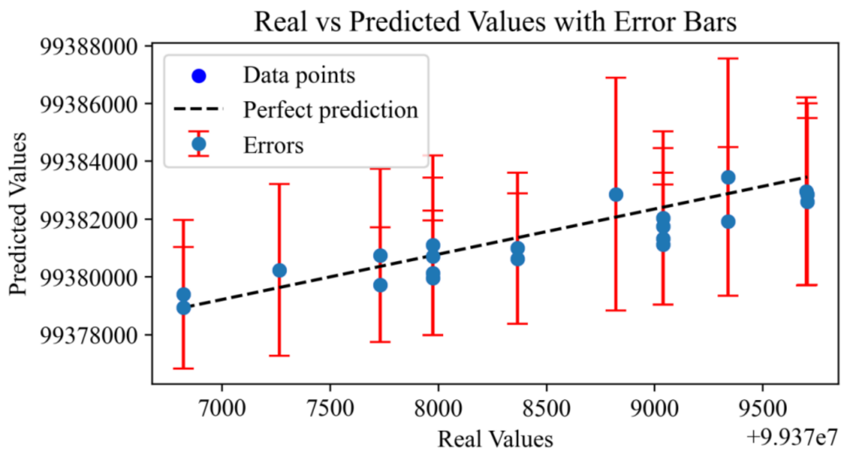

Figure 51: Error scatter for 21 days on LSTM from 2024

7. For the LSTM model with a prediction length of 30 days, the RMSE is 1071.86, MAPE is 0.00002%, and MAE is 927.66. The comparison between the predicted and actual values is shown in Figure 52 and Figure 53.

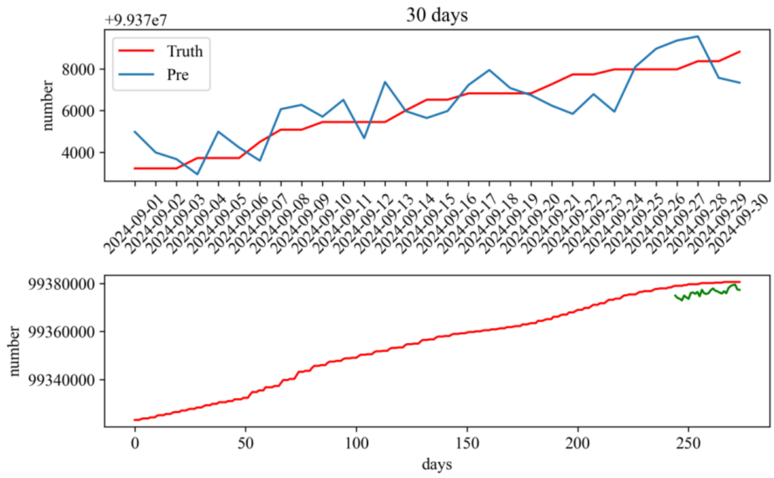

Figure 52: Forecast for 21 days on LSTM from 2024

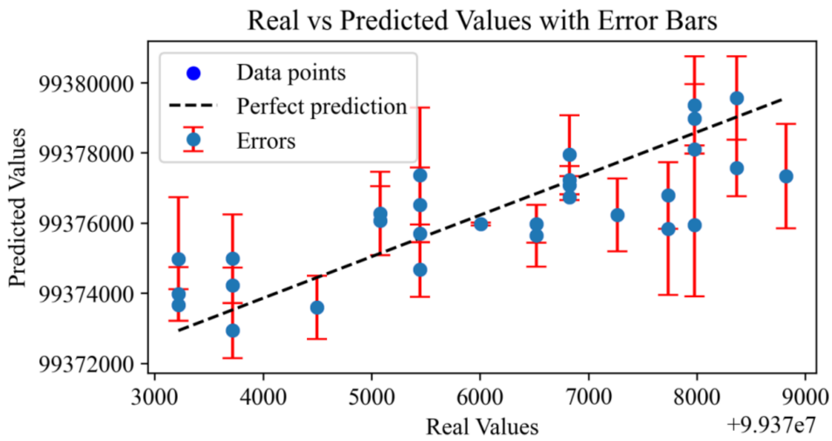

Figure 53: Error scatter for 21 days on LSTM from 2024

Table 7 Summary of LSTM From 2024

Horizon	MAE	RE
3 days	1220	0.002%
5 days	565.2	0.0002%
7 days	719.7	0.0001%
10 days	1576.4	0.001%
14 days	1076.4	0.0002%
21 days	2740.0	0.0002%
30 days	927.66	0.00002%

In Table 6, the LSTM model performs exceptionally well in time series forecasting for 2024, particularly in long-term predictions. For example, the MAE for a 30-day forecast is just 927.66, with the RE as low as 0.00002%, demonstrating extremely high prediction accuracy and stability. Although there are some fluctuations in prediction accuracy in the short term, overall, the model's performance is excellent, with both the Mean Absolute Error (MAE) and Relative Error (RE) remaining at low levels.

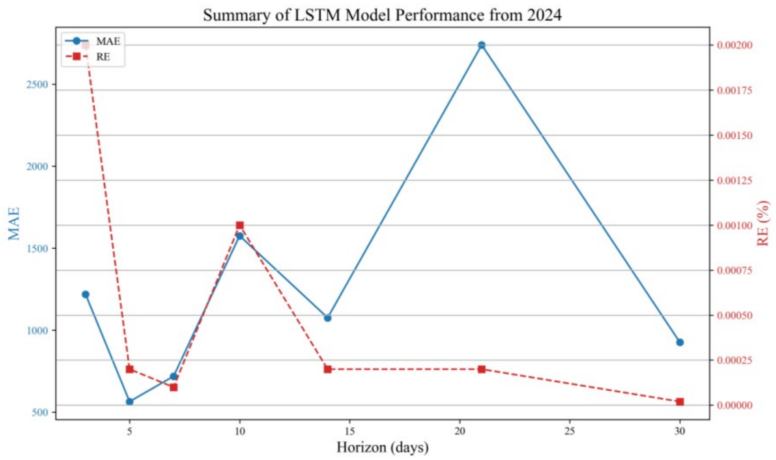

Figure 54: MAE and RE Variation with Prediction Length

Figure 54 summarizes the performance of the 2024 LSTM model across different prediction time horizons, measured by MAE (Mean Absolute Error) and RE (Relative Error). MAE peaks at 3 days and 20 days, exceeding 1200 and 2700, respectively, while it drops to around 500 at 5 days and 15 days, indicating more accurate predictions at these time points. RE remains at a low level, with the highest close to 0.0003% and the lowest near 0.00002%, suggesting that the prediction errors relative to the actual values are very small, demonstrating high model stability.

In terms of trend, MAE is higher in the short term, decreases in the mid-term, rises again in the long term, and eventually drops at even longer time horizons. This suggests that the model performs well for both short-term and long-term predictions, but improvements may be needed for certain mid-term time points. Although RE shows excellent performance, the fluctuations in MAE highlight areas where prediction capabilities at specific time points could be improved. Overall, the LSTM model maintains relatively low prediction errors in most cases.

4.4 Comparison between Lasso and LSTM

Sections 4.3 "Cumulative Infection Prediction Based on LSTM" and 4.2 "Cumulative Infection Prediction Based on Lasso Regression" use two models to predict the cumulative number of COVID-19 infections. The data ranges are from 2020 to the present and from the beginning of 2024 to the present, respectively.

The data from 2020 to the present spans a long period and exhibits significant non-stationary characteristics, which leads to diverse feature distributions. This makes it challenging for both models, resulting in lower performance.

In contrast, the data from the beginning of 2024 to the present has a shorter time span with fewer random factors, leading to better model performance, with significant reductions in both MAE and RE metrics.

For the data from the beginning of 2024 to the present, LSTM outperforms Lasso, proving that deep learning, especially LSTM models designed for time series data, performs better. Lasso, on the other hand, only processes uncorrelated features in the input sequence and cannot extract time-dependent trend features.

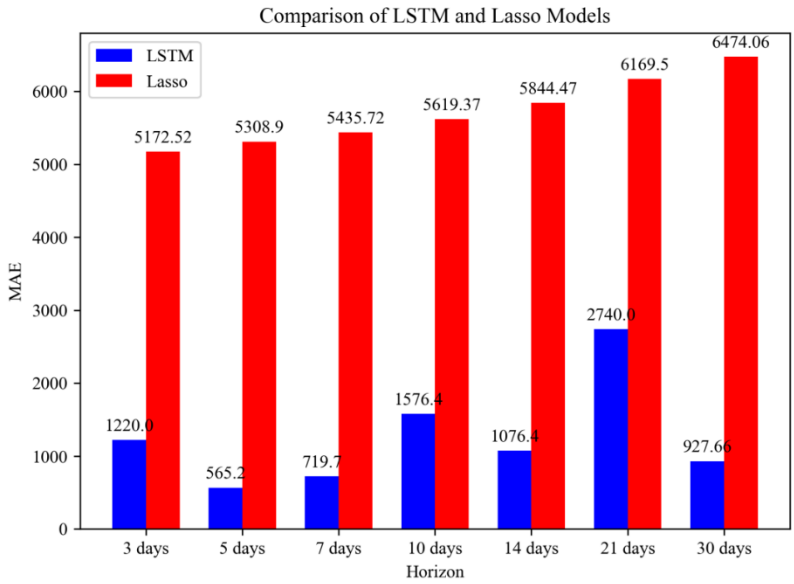

Figure 55: shows a comparison of MAE between LSTM and Lasso models at different prediction horizons based on the 2024 data.

Figure 55 shows a comparison of the performance of the LSTM and Lasso models at different prediction horizons (from 3 days to 30 days), using Mean Squared Error (MSE) as the evaluation metric. From the chart, it is evident that the LSTM model (blue bars) exhibits lower MSE values across all prediction horizons, indicating its superior accuracy in forecasting compared to the Lasso model (red bars). Specifically, as the prediction horizon increases, both models show an increase in MSE values, but the increase in the LSTM model's MSE is relatively smaller, demonstrating its stability over longer prediction periods. For example, at a 3-day prediction horizon, the LSTM model's MSE is 1220.0, while the Lasso model's MSE is 5172.52; at a 30-day prediction horizon, the LSTM model's MSE is 927.66, while the Lasso model's MSE is 6474.06. This indicates that the LSTM model performs better in handling time series forecasting tasks, particularly over longer prediction periods.

4.4 Prediction of Cumulative Infections Based on MultiScale-LSTM-Attention

From the previous chapters, it can be observed that LSTM performs better than machine learning methods. However, there are still fluctuations in LSTM predictions, which may be due to non-stationary factors at different periods. Existing models often only consider fixed-length historical inputs and extract overall time series trend features, ignoring the impact of different cycles within the data. These cyclical factors are crucial for addressing non-stationarity. For example, if the input data consists of the past 30 days, there are multiple cycles such as 3, 5, 7, 14, and 21 days. Traditional LSTMs only consider the overall 30-day sequence, and some CNN-LSTM models can only consider a fixed number of days, failing to account for changing days.

Therefore, building on the previous work in the chapters, this chapter introduces a multi-scale LSTM model with an attention mechanism to enhance the robustness of the model. Figure 56 shows the specific structure of the Multi-Scale Long Short-Term Memory (LSTM) network. It is designed to capture features at different time scales.

The analysis of the figure is as follows:

1. **Input Sequence**: The input sequence at the top, denoted as $X_1 - X_t$, is processed by individual LSTM units.
2. **Multi-Scale Processing**: The figure shows three LSTM layers at different time scales, labeled as $T=3$, $T=5$, $T=7$. These layers process input sequences with varying lengths of time windows. The $T=3$ layer processes three consecutive inputs, $T=5$ processes five consecutive inputs, and $T=7$ processes seven consecutive inputs.
3. **Feature Extraction**: Each LSTM layer at each scale extracts features from its corresponding time window, such as patterns, trends, or other important information in the sequence.
4. **Output**: After the lowest layer LSTM, there is an output layer that generates the final prediction or classification result.

In summary, the advantage of the multi-scale LSTM model lies in its ability to simultaneously capture both short-term and long-term temporal dependencies. By combining features from these different scales, the model's performance is improved.

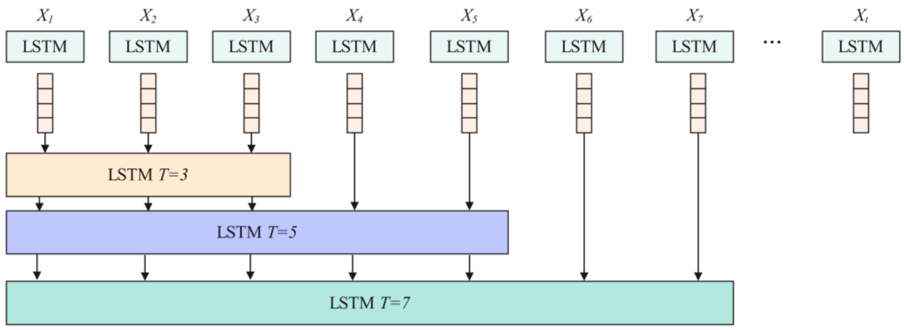

Figure 56: MultiScale-LSTM-Attention Model

4.4.1 The results of 30 days on MSLA from 2024

Figures 57 and 58 show the 30-day prediction results of the MLSA model, where the green line represents the predicted values and the red line represents the actual cumulative infection numbers. It can be observed that the green and red curves almost overlap, and in the error scatter plot, the error points form a straight line, proving that the MLSA model's prediction performance is very good.

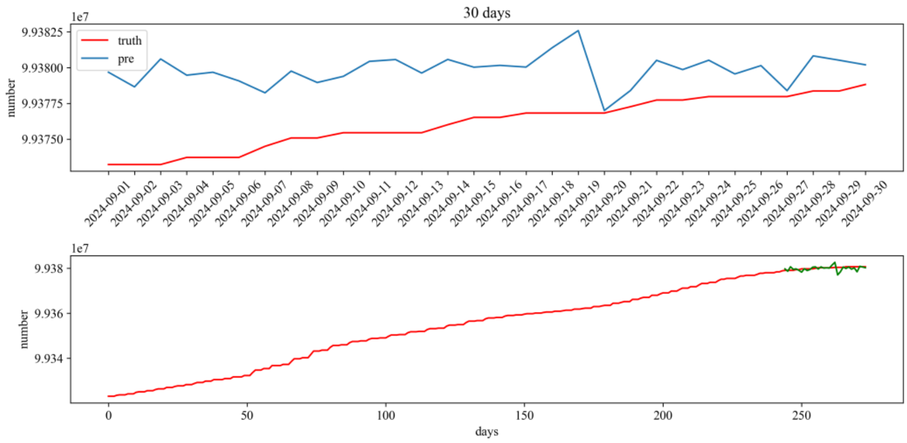

Figure 57: MSLA Model Prediction Results

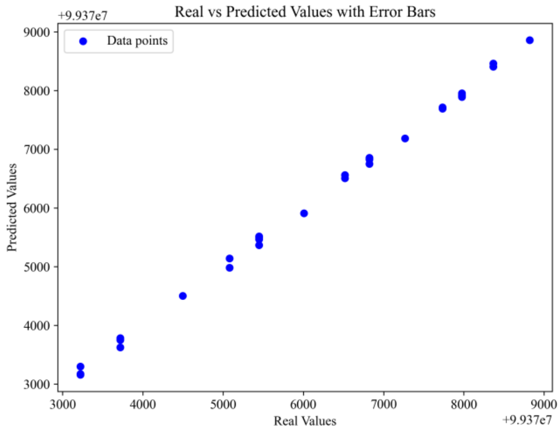

Figure 58: MSLA Model Prediction Error Scatter Plot

4.4.2 Comparison of Results between LSTM and MSLA

LSTM demonstrated good performance in previous prediction results, and MLSA further improved upon LSTM by adding multi-scale and attention mechanisms. To validate the effectiveness of MLSA and the added modules, an ablation comparison was conducted.

Table 8: MAE Comparison between LSTM and MLSA

Horizon	LSTM	MLSA
30 days	927.66	786.4

In Table 8, for the 30-day forecast period, the prediction result from LSTM is 927.66, while the prediction result from MLSA is 786.4, indicating that MLSA performs significantly better than LSTM.

Table 9 shows the results of the ablation experiments for the MLSA model.

Module	MAE
+Attention	803.4
+MultiScale-3	850.32
+MultiScale-5	845.33
+MultiScale-7	820.34
+MultiScale-10	812.56
+MultiScale-14	810.73
+MultiScale-21	800.65
MLSA	786.40

When evaluating the prediction performance of different modules, the Mean Absolute Error (MAE) is used as the metric. MAE represents the average absolute difference between the predicted values and the actual values; the lower the MAE, the higher the prediction accuracy of the model.

Figures 59 and Table 9 show significant differences in performance for each module over a 30-day prediction horizon. First, the "+Attention" module achieved the lowest MAE value of 803.4, indicating that the attention mechanism significantly improves prediction performance. Next, the MultiScale series of modules demonstrated a trend of decreasing MAE as the scale parameter increased. Specifically, the MAE decreased from 850.32 for MultiScale-3 to 800.65 for MultiScale-21, showing a consistent downward trend. This suggests that larger scale parameters help improve the model's prediction accuracy. However, even the optimal MultiScale-21 configuration did not outperform the +Attention and MLSA models in terms of MAE. This result indicates that, although the MultiScale model can be gradually optimized by adjusting the scale parameter, it does not provide higher prediction accuracy than the +Attention/MLSA configuration in the current setup.

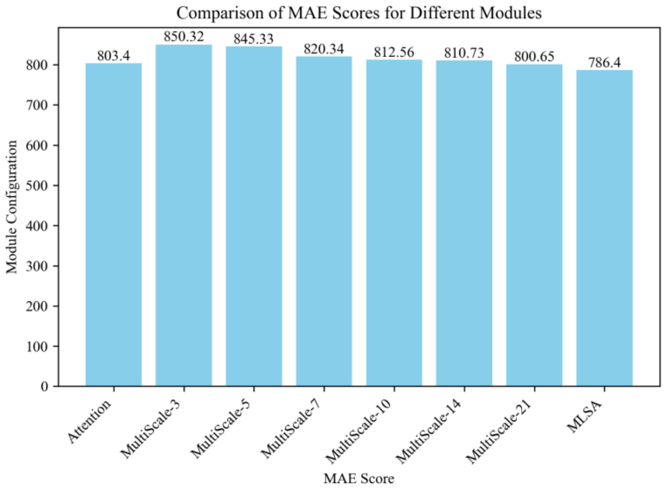

Figure 59: The ablation results of different modules

Based on the above analysis, if the goal is to select the most accurate prediction model, the +Attention or MLSA models should be prioritized based on the MAE metric. Both of these models not only offer the lowest prediction errors, but their performance is also entirely consistent, providing a reliable choice for practical applications. Additionally, incorporating features from different scales significantly improves prediction performance, especially when the scales align with non-stationary cycles. For example, when the non-stationary cycle is 7, using a scale of 7 significantly enhances the prediction performance.

5 Conclusions

This study successfully developed a comprehensive multi-model COVID-19 prediction system by integrating machine learning and deep learning techniques. The system combines Lasso regression, Long Short-Term Memory (LSTM) networks, and a novel Multi-Scale Cumulative Infection Prediction model based on an attention mechanism (MSLA), significantly enhancing prediction accuracy and reliability. Relying on COVID-19 data from China, the study carefully performed data preprocessing and feature engineering to ensure data accuracy and adequate information extraction. The training, evaluation, and visualization of the models were thoroughly documented, providing clear guidance for future model optimization.

The integration of the MSLA model allows this research to simultaneously account for multiple periodic factors, effectively addressing the non-stationary nature of epidemic data. Experimental results reveal that the MSLA model outperforms the traditional LSTM model in 30-day long-term predictions, reducing the Mean Absolute Error (MAE) by 15.22%, offering more accurate scientific support for public health decision-making.

Overall, this study not only provides an innovative and effective tool for COVID-19 prediction but also offers valuable experience and methodology for addressing similar pandemics in the public health field. As the pandemic continues to evolve, the methodology and framework presented in this study will provide strong support for global epidemic monitoring and response, helping to mitigate the impact of the epidemic on social economies and public health systems.

6 Reference

[1] Ciotti, M., Ciccozzi, M., Terrinoni, A., Jiang, W. C., Wang, C. B., & Bernardini, S. (2020). The COVID-19 pandemic. Critical reviews in clinical laboratory sciences, 57(6), 365-388.

[2] Santosh, K. C. (2020). COVID-19 prediction models and unexploited data. Journal of medical systems, 44(9), 170.

[3] Singh, R. K., Rani, M., Bhagavathula, A. S., Sah, R., Rodriguez-Morales, A. J., Kalita, H., ... & Kumar, P. (2020). Prediction of the COVID-19 pandemic for the top 15 affected countries: Advanced autoregressive integrated moving average (ARIMA) model. JMIR public health and surveillance, 6(2), e19115.

[4] Alabdulrazzaq, H., Alenezi, M. N., Rawajfih, Y., Alghannam, B. A., Al-Hassan, A. A., & Al-Anzi, F. S. (2021). On the accuracy of ARIMA based prediction of COVID-19 spread. Results in Physics, 27, 104509.

[5] Yonar, H., Yonar, A., Tekindal, M. A., & Tekindal, M. (2020). Modeling and Forecasting for the number of cases of the COVID-19 pandemic with the Curve Estimation Models, the Box-Jenkins and Exponential Smoothing Methods. EJMO, 4(2), 160-165.

[6] Van Tinh, N. (2020). Forecasting of COVID-19 confirmed cases in Vietnam using fuzzy time series model combined with particle swarm optimization. Comput Res Prog Appl Sci Eng, 6(2), 114-120.

[7] Song, J., Xie, H., Gao, B., Zhong, Y., Gu, C., & Choi, K. S. (2021). Maximum likelihood-based extended Kalman filter for COVID-19 prediction. Chaos, Solitons & Fractals, 146, 110922.

[8] Islam, M. S., Hoque, M. E., & Amin, M. R. (2021). Integration of Kalman filter in the epidemiological model: a robust approach to predict COVID-19 outbreak in Bangladesh. International Journal of Modern Physics C, 32(08), 2150108.

[9] Heidari, A., Jafari Navimipour, N., Unal, M., & Toumaj, S. (2022). Machine learning applications for COVID-19 outbreak management. Neural Computing and Applications, 34(18), 15313-15348.

[10] Niazkar, H. R., & Niazkar, M. (2020). Application of artificial neural networks to predict the COVID-19 outbreak. Global Health Research and Policy, 5, 1-11.

[11] Rochim, A. F., Widyaningrum, K., & Eridani, D. (2021, December). Performance Comparison of Support Vector Machine Kernel Functions in Classifying COVID-19 Sentiment. In 2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI) (pp. 224-228). IEEE.

[12] Gupta, M., Bansal, A., Jain, B., Rochelle, J., Oak, A., & Jalali, M. S. (2021). Whether the weather will help us weather the COVID-19 pandemic: Using machine learning to measure twitter users' perceptions. International journal of medical informatics, 145, 104340.

[13] Sheela, M. S., & Arun, C. A. (2022). Hybrid PSO–SVM algorithm for Covid-19 screening and quantification. International Journal of Information Technology,

14(4), 2049-2056.
[14] Gaur, P., Malaviya, V., Gupta, A., Bhatia, G., Pachori, R. B., & Sharma, D. (2022). COVID-19 disease identification from chest CT images using empirical wavelet transformation and transfer learning. Biomedical Signal Processing and Control, 71, 103076.
[15] Abdelkader, H. E., Gad, A. G., Abohany, A. A., & Sorour, S. E. (2022). An efficient data mining technique for assessing satisfaction level with online learning for higher education students during the COVID-19. IEEE Access, 10, 6286-6303.
[16] Luo, J., Zhang, Z., Fu, Y., & Rao, F. (2021). Time series prediction of COVID-19 transmission in America using LSTM and XGBoost algorithms. Results in Physics, 27, 104462.
[17] Alakus, T. B., & Turkoglu, I. (2020). Comparison of deep learning approaches to predict COVID-19 infection. Chaos, Solitons & Fractals, 140, 110120.
[18] Shahid, F., Zameer, A., & Muneeb, M. (2020). Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM. Chaos, Solitons & Fractals, 140, 110212.
[19] Islam, M. Z., Islam, M. M., & Asraf, A. (2020). A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. Informatics in medicine unlocked, 20, 100412.
[20] Shah, P. M., Ullah, F., Shah, D., Gani, A., Maple, C., Wang, Y., ... & Islam, S. U. (2021). Deep GRU-CNN model for COVID-19 detection from chest X-rays data. Ieee Access, 10, 35094-35105.
[21] Pandianchery, M. S., Sowmya, V., Gopalakrishnan, E. A., Ravi, V., & Soman, K. P. (2023). Centralized CNN–GRU model by federated learning for COVID-19 prediction in India. IEEE Transactions on Computational Social Systems, 11(1), 1362-1371.
[22] Ullah, Z., Usman, M., Latif, S., & Gwak, J. (2023). Densely attention mechanism based network for COVID-19 detection in chest X-rays. Scientific Reports, 13(1), 261.
[23] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. Neural computation, 31(7), 1235-1270.
[24] Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., & Wang, Y. (2021). Transformer in transformer. Advances in neural information processing systems, 34, 15908-15919.
[25] Ranstam, J., & Cook, J. A. (2018). LASSO regression. Journal of British Surgery, 105(10), 1348-1348.