VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6569

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

COMPARATIVE OF LSTM AND GRU FOR TRAFFIC PREDICTION AT ADIPURA INTERSECTION, BANDAR LAMPUNG

Ilham Firman Ashari[1*]; Verlina Agustine[2]; Aidil Afriansah[3]; Nela Agustin Kurnianingsih[4]; Andre Febrianto[5]; Eko Dwi Nugroho[6]

Informatics Engineering, Faculty of Industrial Technology[1,3,5,6]
Urban and Regional Planning, Faculty of Infrastructure and Regional Technology[2,4]
Institut Teknologi Sumatera, South Lampung, Indonesia[1,2,3,4,5,6]
itera.ac.id[1,2,3,4,5,6]
firman.ashari@if.itera.ac.id[1*] , verlina.agustine@pwk.itera.ac.id[2], aidil.afriansyah@if.itera.ac.id[3], nela.agustine@pwk.itera.ac.id[4], andre.febrianto@if.itera.ac.id[5], eko.nugroho@if.itera.ac.id[6]

(*) Corresponding Author
(Responsible for the Quality of Paper Content)

Abstract—*The Tugu Adipura intersection in Bandar Lampung is a vital traffic hub connecting four major roads. Rapid population growth and increasing vehicle numbers challenge traffic flow and urban quality of life. Despite its importance, there is limited research using predictive models to analyze traffic patterns at complex intersections in mid-sized Indonesian cities. This study addresses that gap by examining traffic growth on four connected roads using deep learning models. Traffic data were collected hourly from June 1, 2021, to July 31, 2023. A comparative analysis of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models was conducted, with SGD and Adam as optimizers. Results show the GRU model with Adam achieved the lowest RMSE (0.23) on road section 1, indicating its superior ability to model short-term fluctuations and non-linear growth in traffic volume. The study offers practical implications for traffic management by highlighting GRU's capacity to capture seasonal trends and rapid growth, supporting proactive infrastructure planning and congestion mitigation strategies. These findings demonstrate the value of data-driven approaches in enhancing transportation systems in growing urban areas.*

Keywords: *GRU, intersection, LSTM, roads, traffic prediction.*

Intisari— Simpang Tugu Adipura di Kota Bandar Lampung merupakan simpul lalu lintas penting yang menghubungkan empat jalan utama. Pertumbuhan penduduk dan peningkatan jumlah kendaraan menimbulkan tantangan terhadap kelancaran lalu lintas dan kualitas hidup masyarakat. Meskipun peran simpang ini vital, masih sedikit penelitian yang menggunakan model prediktif untuk menganalisis pola pertumbuhan lalu lintas di simpang kompleks di kota menengah Indonesia. Penelitian ini bertujuan mengisi celah tersebut dengan menganalisis pertumbuhan lalu lintas di empat ruas jalan menggunakan model pembelajaran mendalam. Data lalu lintas dikumpulkan setiap jam dari 1 Juni 2021 hingga 31 Juli 2023. Analisis komparatif dilakukan terhadap model Long Short-Term Memory (LSTM) dan Gated Recurrent Unit (GRU) dengan optimasi menggunakan algoritma SGD dan Adam. Hasil menunjukkan bahwa model GRU dengan Adam menghasilkan RMSE terendah (0,23) pada ruas jalan 1, menunjukkan keunggulan dalam memodelkan fluktuasi jangka pendek dan pertumbuhan non-linear. Temuan ini memiliki implikasi praktis dalam manajemen lalu lintas, terutama dalam mendeteksi tren musiman dan pertumbuhan, sehingga mendukung perencanaan infrastruktur dan strategi mitigasi kemacetan secara proaktif. Pendekatan berbasis data ini terbukti efektif untuk perencanaan transportasi di wilayah perkotaan yang berkembang.

Kata Kunci: *GRU, persimpangan, LSTM, jalan, prediksi lalu lintas.*

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6569

INTRODUCTION

In the context of urban growth and population increase, especially in large urban city center areas, space resources are increasingly becoming a significant limitation [1]. This phenomenon is triggered by high urban activity and population mobility, which has an impact on the accessibility and efficiency of transportation in the city center [2][3]. Traffic, as an integral element in the urban transportation system, plays a crucial role in determining the level of city accessibility. The minimal obstacles to vehicles in traveling indicate that the level of transportation accessibility is good and can support community mobility and urban activities [3]. Therefore, accurate and efficient transportation planning is a must to create and maintain optimal accessibility in limited urban space. Traffic prediction aims to predict future road traffic conditions from historical traffic data, which is very important for urban planning and construction [4]. Accurate traffic pattern prediction is an important element in optimal transportation planning. Given the complexity of spatial phenomena that influence traffic, such as public interest, transit connections, and similarities in historical traffic patterns, a holistic and innovative approach is needed [5].

Bandar Lampung City is the urban center of Lampung Province which has several intersections, one of which is the Tugu Adipura intersection. The Tugu Adipura intersection is included in an important traffic node in the context of urban mobility in Bandar Lampung because it connects 4 roads, namely Jl. Jendral Sudirman, Jl. Raden Intan, Jl. Diponegoro, and Jl. Ahmad Yani. The increasing traffic density in this area, accompanied by population growth and an increase in the number of vehicles, poses significant challenges to the smooth flow of traffic and the quality of life of the community. In 2022, the level of congestion or degree of saturation at the Tugu Adipura intersection is predicted to not meet the standards of the Indonesian Road Capacity Manual (MKJI) [6]. Management is needed to overcome this transportation problem. This research aims to develop a predictive model that can project traffic patterns at the Tugu Adipura intersection. It is hoped that the research results will provide insight for authorities in designing more efficient traffic management strategies, as well as provide a basis for better decision making in urban traffic management.

In terms of making predictions, there are various algorithms that have unique and special abilities. One of them is Long Short-Term Memory (LSTM), a type of recurrence architecture specifically designed to handle the vanishing gradient problem in recurrent neural networks (RNN) [7][8][9]. The Long Short Term Memory algorithm is an algorithm resulting from further development of RNN with 3 additional gates (forget gate, input gate and output gate) as a regulator of information flow when studying long-term dependencies and certain information in the data to make predictions. LSTM capabilities are very effective in understanding and predicting complex patterns in time series data [7]. Gated Recurrent Unit (GRU), as another variant of RNN, also focuses on dealing with the vanishing gradient problem [8]. The gate recurrent unit has a simpler architecture than LSTM because it only has 2 gates (reset gate and update gate) to regulate the flow of information in the neural network, making GRU able to overcome overfitting problems when learning model learning [8][9]. The parameters that will be identified in this research are intersection roads, time, and number of vehicles, where the intersection roads that will be evaluated are roads two, three, and four. Data taken from June 2021 to July 2023. Data is taken every day where the observation range is once every 1 hour.

Several related studies have been carried out by Dey P, et al, where the researchers compared the performance of the forecasting results of the RNN, LSTM and GRU algorithms in predicting stock prices using different data trends with the results of the mean absolute error of RNN being 5.26118, LSTM being 6.20282 and GRU being 3.70209 for the data with high variance, then for data with low variance RNN is 0.64351, LSTM is 0.53824 and GRU is 0.55344 [10]. Another research conducted by Shahid F, Zameer A & Muneeb M with the title "Predictions for COVID-19 with deep learning models" carried out a comparison between LSTM and GRU to predict cases of infection, death and recovery using COVID-19 case data from various regions of the world with the results, LSTM has an MAE value predicting infected cases of 2.0463 and deaths of 0.0095 while the MAE GRU value of infected cases is 2.8553, deaths are 0.0321 and recoveries are 7.04867 [11]. Another research conducted by Karyadi, Y and Santoso, H by the application of the LSTM and LSTM Bidirectional models showed better results than the Gated Recurrent Unit (GRU) model in handling air quality time series data. Performance evaluation using RMSE on both LSTM and Bidirectional LSTM models shows a smaller comparison compared to the standard deviation of the test dataset [9]. The next research is from A. Nilsen, who predicted LQ45 stock prices and the model was evaluated using

Root Mean Square Error, Mean Square Error, and Mean Absolute Error. In this research, the same hyperparameters are used for all models, namely {epoch=200, batch size=32, and units=24}. From the average Root Mean Square Error (RMSE), Mean Square Error (MSE), and Mean Absolute Error (MAE) produced from the three models, it is concluded that the GRU model has better accuracy than the Recurrent Neural Network (RNN) model and Long Short-Term Memory (LSTM) [12].

This research presents a novelty in applying time series prediction models—specifically Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)—in a comparative study at a four-way intersection in a mid-sized Indonesian city, namely the Tugu Adipura intersection in Bandar Lampung. Unlike previous studies that mainly focused on stock price forecasting, air quality, or pandemic case predictions, this study emphasizes the complexity of urban traffic dynamics, which are influenced by daily fluctuations and seasonal patterns. LSTM and GRU were selected due to their proven ability to capture long-term dependencies in time series data and their effectiveness in addressing the vanishing gradient problem commonly found in standard Recurrent Neural Networks (RNNs). LSTM uses a more complex architecture with three gates (input, forget, and output), while GRU offers a simpler structure with two gates (reset and update), making it more computationally efficient in certain cases. Classical models such as ARIMA or Support Vector Regression (SVR) are considered less flexible in capturing the nonlinear and seasonal characteristics of road traffic data, thus deep learning approaches provide a more adaptive and suitable solution for the data characteristics in this study [13].

In this research, researchers want to determine the algorithm that has the lowest prediction error value when applied to time series data related to road density. To measure the model that has been developed, measurements using RMSE are used [14]. The selection of these two algorithms is based on the fact that LSTM and GRU are developments of neural networks for modeling time series data, as has been proven in previous research by Dey P, et al regarding forecasting on time series data objects [10]. Different levels of forecasting success in previous research are acknowledged to be influenced by the complexity of the data, the amount of data, and the characteristics of the object being predicted. Therefore, it is hoped that this research can contribute to helping other researchers in choosing a suitable prediction algorithm for road density objects. This research aims to compare the performance of LSTM and GRU algorithms in predicting road traffic density at the Adipura Intersection, Bandar Lampung City, using time series data. It seeks to determine which model provides the lowest prediction error (RMSE) and to offer a reliable reference for selecting and developing effective machine learning models for traffic prediction tasks, both in terms of resource efficiency and model performance.

MATERIALS AND METHODS

The flow of research carried out in this research can be seen in the flow diagram in Figure 1. The research will start from the problem identification stage, literature study, data collection, data preprocessing, data pre-processing, modeling, testing and performance comparison.

Source: (Research Results, 2025)

Figure 1. Research Flow

Problem identification

The problem identification stage involves a background search to find the root cause of the problem being faced. This research was triggered by the lack of comparative research regarding LSTM and GRU algorithms in predicting road density. Heavy traffic in strategic locations has a serious impact on human life, encouraging the urgency of carrying out this research to understand and overcome the increasing losses.

Study literature

At this stage researchers will look for information related to Long Short Term Memory and Gated Recurrent Unit algorithms from various sources ranging from journals, books and other trusted sources.

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6569

Data collection

The data collection stage is the process where researchers collect the data needed to carry out research. The data used in this research comes from on-site observations from June 1, 2021 to July 31, 2023. The data collected is of the time series type with 8 parameters, namely Data Recap Time, Road, Number of Vehicles, Year, Month, Date, Hour, Day.

Table 1. An Examples of Data

Data Collection Time	2021-11-01 00:00:00	2021-11-01 01:00:00	2021-11-01 02:00:00	2021-11-01 03:00:00
Road	1	1	1	1
Number of Vehicles	12	10	7	5
Year	2023	2023	2023	2023
Month	11	11	11	11
No Date	1	1	1	1
Time	0	1	2	3
Hari	Monday	Monday	Monday	Monday

Source: (Research Results, 2025)

Pre-processing

The data pre-processing stage aims to prepare and process raw data so that it can be used effectively in training learning models according to research needs [15]. This process involves a series of steps to improve data quality, overcome imperfections, and align the data structure with the needs of the analysis model. The following are the objectives of the data pre-processing stages [16]:

1. Cleaning Data
   Eliminate noise, outliers, or irrelevant data to improve data quality. This includes the identification and handling of missing or anomalous values [17].
2. Normalization
   Aligning data scales to have a similar range of values ensures that features have a balanced impact on the model.
3. Data Transformation
   Perform additional transformations or normalization if necessary, such as changing the data distribution.

Modeling

At the modeling stage, researchers will design a learning model using the LSTM and GRU algorithms. The initial step involves building a machine learning model architecture, where training parameters for each algorithm are determined [18]. After the model architecture is complete, the process continues to the training stage by inputting data to train the machine, with the aim of detecting patterns and important relationships between the data to make predictions. Evaluation of machine learning model performance involves analysis of accuracy and root mean square error (RMSE). Researchers monitor learning by checking loss, RMSE, and mean square error at each epoch. The model training process continues until it reaches the desired performance level. Performance tuning involves adjusting architectural parameters, such as varying learning rate or hyperparameter values, to improve validation, reduce loss, and update weights to optimize the loss function during the learning process.

In this study, both GRU and LSTM models were developed using deep architectures with stacked recurrent layers to improve the model's capacity for capturing complex temporal dependencies. Each model used five recurrent layers with a decreasing number of units (150 to 50), designed to gradually distill information across time steps. A tanh activation function was used to ensure non-linearity while maintaining stability in the gradient flow. Dropout layers with a dropout rate of 0.2 were inserted between layers to mitigate overfitting by randomly deactivating neurons during training. The models were trained for a maximum of 50 epochs with an early stopping mechanism (patience of 10, min_delta 0.001) to avoid overfitting and reduce unnecessary training time.

The optimizer used in the final model compilation was **Adam**, as it combines the advantages of momentum and adaptive learning rates, making it more robust for training deep networks on time series data with sparse and noisy gradients. **SGD with momentum** was also tested in preliminary runs to benchmark its performance, but Adam was ultimately favored for its faster convergence and better generalization in this case. A batch size of 120 was selected based on empirical testing to balance between convergence speed and memory efficiency.

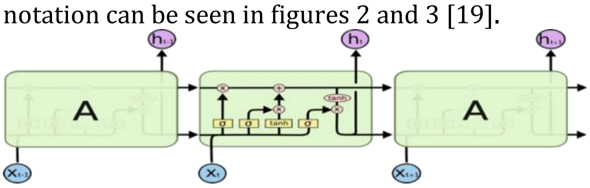
Root Mean Square Error (RMSE) was selected as the primary evaluation metric because it gives higher penalization to larger errors, which is critical in traffic forecasting applications where large deviations can lead to misleading conclusions for traffic management. RMSE also maintains the same unit as the predicted variable, which makes interpretation straightforward for decision-makers.

Long short-term memory

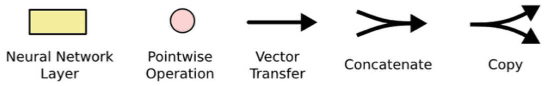
LSTM is a modification of RNN which is equipped with memory and several types of gates, such as input gate, forget gate, and output gate. LSTM's ability to understand data patterns allows it to learn information from more than 1000 previous steps, depending on the complexity of the network.

P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6569

Due to its unique gate structure, LSTM can overcome the long-term dependency problem and process time series data more effectively than conventional RNNs. LSTM architecture and diagram notation can be seen in figures 2 and 3 [19].



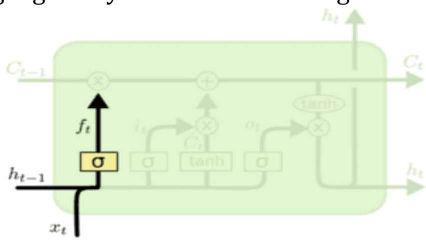
Source: (Research Results, 2025)
Figure 2. LSTM Architecture Flow Diagram



Source: (Research Results, 2025)
Figure 3. LSTM Diagram Notation

In the diagram 3, each line carries a complete vector, from the output of one node to the input of another node. Pink circles depict pointwise operations, such as vector addition, while yellow squares represent the learned layers of the neural network. Converging lines indicate a merge, while branching lines indicate a copy of its content and the copy going to a different location. In this context, each line in the diagram depicts the flow of information, and the color and shape of elements such as pink circles and yellow squares indicate the operations and layers of the neural network involved. A line merge represents combining information from different sources, while a line branch represents a copy of the information that can then be used in a different location. This provides a visual representation of how information flows and is processed in a neural network architecture such as LSTM.

Forget gate in Long Short-Term Memory (LSTM) architectures play a crucial role in determining which information will remain stored and which will be discarded during time series data processing. This process begins by receiving two types of information: hidden state from the previous cell and new information from the current input. These two pieces of information are then combined and processed via the sigmoid function. The sigmoid function produces an output between 0 and 1. A value close to 0 indicates that the information will be ignored or discarded, while a value close to 1 indicates that the information is considered important and needs to be stored in the cell state. In other words, the greater the output of the sigmoid function, the more likely it is that the information remains in the cell's memory. Selecting relevant information to retain or ignore is critical in the context of time series data. Forget gate allows LSTM to adaptively decide whether information in a memory cell needs to be updated or forgotten. This allows LSTM to overcome the vanishing gradient problem and maintain a balance between remembering long-term information and maintaining flexibility in the face of changes in input data. Thus, the forget gate is one of the key components that supports the effectiveness of LSTM in processing and predicting time series data. The forget gate layer can be seen in Figure 4.

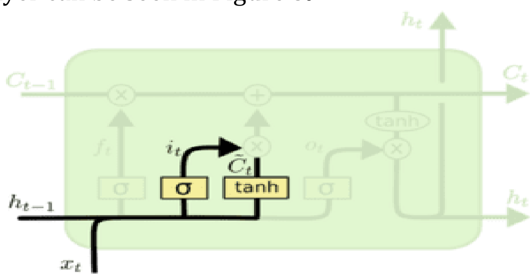


Source: (Research Results, 2025)
Figure 4. Forget Gate Layer

The function formula for forget gate can be seen in formula 1 [20].

$$f_t = \sigma\left(W_f * [\,h_{t-1}, x_t] + b_f\right) \quad (1)$$

Input gate will receive two types of information, namely hidden state from the previous cell and new information from the current input. Next, these two pieces of information will be combined and processed using the sigmoid function and tanh function. The output of the sigmoid function will convert the values into a range of 0 to 1, which determines which information will be updated. A value close to 0 indicates that the information is considered unimportant, while a value close to 1 indicates that the information is considered important. Meanwhile, the output of the tanh function, which is a value of -1 to 1, is used to help cells learn information better. The input gate layer can be seen in Figure 5.



Source: (Research Results, 2025)
Figure 5. Input Gate Layer

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6569

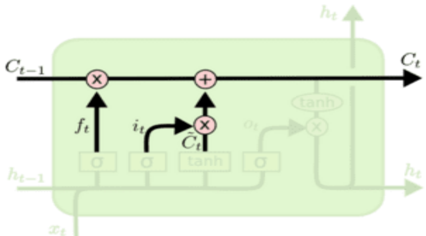
The sigmoid layer will determine which values will be entered into the cell state. The sigmoid layer formula can be seen in formula 2.

$$i_t = \sigma(W_i * [\, h_{t-1}, x_t] + b_i) \qquad (2)$$

The Tanh Layer functions to determine how many new candidates will be included in the cell state. The tanh layer formula can be seen in formula 3.

$$\tilde{c}_t = \tanh\,(W_c * [\, h_{t-1}, x_t] + b_c \qquad (3)$$

Cell State is additional memory owned by the LSTM unit and is absent in the regular RNN unit. Cell State can be thought of as a long straight line at the top of an architectural diagram. With minor linear interactions with other processes, Cell State allows information to flow without change. This is what makes LSTM effective in dealing with vanishing gradient problems. The cell state layer can be seen in Figure 6.



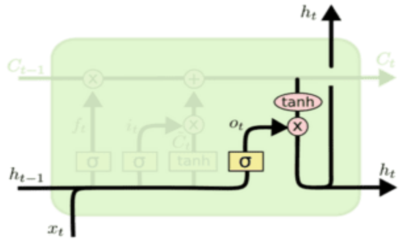
Source: (Research Results, 2025)
Figure 6. Cell State Layer

After the input gate process is complete and we know which information we want to enter the cell state using Formula 4.

$$c_t = f_t \; * c_{t-1} + i_t * \tilde{c}_t \qquad (4)$$

f_t * C_{t-1} will delete the value in the cell state while i_t * $\tilde{c}_t$ will insert new information into the cell state.

Output gate plays a role in determining the hidden state that will be sent to the next cell [21]. To do this, the gate output receives two types of information: hidden state from the previous cell and new information from the current input. Next, the two pieces of information are combined and processed using the sigmoid function. The new cell state is then processed via the tanh function. The output of the tanh function is multiplied by the output of the sigmoid function to get the information that will be stored in the new hidden state. This new hidden state and cell state will then be passed to the next cell. The output gate uses a Sigmoid layer to determine how much of the cell state will flow to the output. After that, the information from the cell state goes through a tanh process, changing the value to a range of -1 to 1. The result is then multiplied by the output from the Sigmoid layer. This final value determines the extent to which information will be passed to the output; if it is close to 1, information will be passed, but if it is close to 0, information will be suppressed and will not enter the output. The gate layer output can be seen in Figure 7.will delete the value in the cell state while i_t * c ̃_t will insert new information into the cell state.



Source: (Research Results, 2025)
Figure 7. Output Gate Layer

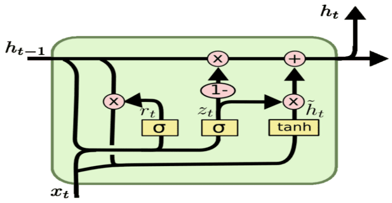
$$o_t = \sigma(W_o * [\, h_{t-1}, x_t] + b_o) \qquad (5)$$

$$h_t = o_t * \tanh(c_t) \qquad (6)$$

Gated recurrent unit

Gated Recurrent Unit (GRU) is an innovative architecture in recurrent neural networks that is intended to overcome the vanishing gradient problem that often occurs in sequence data. Along with its similarity to Long Short-Term Memory (LSTM), GRU also utilizes a gated unit that is capable of selectively updating or ignoring information through two main components, namely the update gate and reset gate. While GRU has similar functionality to LSTM, GRU architecture tends to be simpler with a lower number of parameters. These advantages make it a faster choice in the training process and more resistant to overfitting problems. The advantage of GRU lies in its more compact structure, resulting in more efficient use of computing resources compared to LSTM. With a simpler architecture, GRU offers a more effective and resource-saving solution for handling sequential data. The comparison between GRU and LSTM can be seen clearly in Figure X, which visualizes the structure and relationships between units in the GRU architecture. Thus, GRU becomes an attractive option in facing the vanishing gradient challenge, proving itself as an efficient and effective alternative in processing sequential information. An image of the GRU architecture can be seen in Figure 8.

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6569



Source: (Research Results, 2025)
Gated Recurrent Unit Architecture

1. Reset Gate

The reset gate in the GRU has a crucial role in controlling the amount of information to be deleted or retained from the hidden state in the previous timestep. The function is explained through Formula 7, which uses input from h_(t-1) and x_t to produce the gate reset vector value. With this mechanism, GRU can selectively regulate the extent to which historical information will be retained in current data processing. In other words, the reset gate gives the GRU the flexibility to adjust the rate of deletion or retention of information, creating adaptive intelligence that helps in handling sequential data effectively. The formula for resetting the gate can be seen in formula 7.

$$r_t = \sigma(W_r * x_t + u_r * h_{t-1} + b_r) \tag{7}$$

W and u are the weight matrices used in the activation calculation, while b is the bias associated with the matrix.

2. Update Gate

The update gate function in recurrent network architectures, such as the Gated Recurrent Unit (GRU), is very crucial because it acts as a control gate that determines whether new incoming information will be stored or discarded. With this capability, update gates ensure selectivity in information handling, ensuring only relevant and significant data is passed to the next stage. The formula for updating the gate can be seen in formula 8.

$$z_t = \sigma(W_z * x_t + u_z * h_{t-1} + b_z) \tag{8}$$

After passing the reset gate and update gate stages, the information in the neuron will be used to update the current hidden state (h_t) value by calculating a new candidate value for the hidden state $(\hat{h}_t)$ using Formula 9.

$$\hat{h}_t = tanh(W_h * x_t + u_h(r_t * h_{t-1}) + b_h) \tag{9}$$

Then the current hidden state h_t value is obtained using Formula 10.

$$h_t = (1 - z_t) * h_{t-1} + (z_t * \hat{h}_t) \tag{10}$$

Data training and testing

At this stage the data will be divided into 2, namely training data of 52425 data and testing data of 5242 data. The figures above were obtained by dividing 90% of the dataset for training and the remaining 10% for testing.

RESULTS AND DISCUSSION

Pre-Processing

The Normalize function applies Z-score normalization to standardize the data with formula 11.

$$z = \frac{(X-\mu)}{\sigma} \tag{11}$$

where X is the original value, μ is the mean, and σ is the standard deviation. This function returns the normalized data along with the mean and standard deviation. The Difference function calculates the difference between the current value and the previous value based on a certain interval, based on formula 12.

$$D_t = X_t - X_{t-interval}$$
(12)

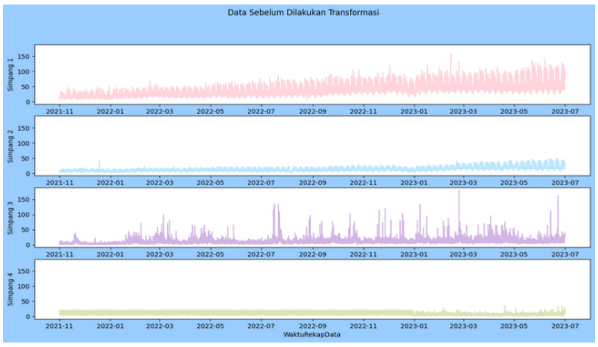
Used in time series analysis to remove trends. This function produces a list of differences that can help in making the data stationary. The plot visualization results of the data before transformation can be seen in Figure 10.

Data stasioner is required in some time series analysis methods, especially in models such as ARIMA (AutoRegressive Integrated Moving Average). After normalization, differentiation is carried out on the normalized data using the Differentiation function. Differentiation is carried out at certain intervals:

1. 24*7 for df_N1, meaning for differentiation 1 week.
2. 24 for df_N2, meaning for differentiation of 1 day or 24 hours.
3. 1 for df_N3 and df_N4, meaning for 1 hour differentiation.

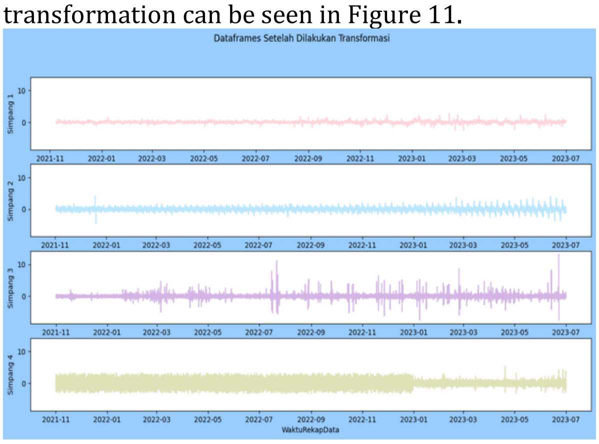
Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)
VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6569



Source: (Research Results, 2025)
Figure 10. Data Visualization Before Transformation

The plot visualization results of the data after transformation can be seen in Figure 11.



Source: (Research Results, 2025)
Data Visualization After Transformation

Augmented Dicket-Fuller Validation

From Figure 11 the data looks linear. To ensure that the data is stationary, an augmented dicket-fuller test is carried out. The Augmented Dickey-Fuller test is used to determine whether a time series can be considered stationary or not [22][23]. Stationary_check function: This function accepts a time series DataFrame (df) and uses ADF test to test stationarity. ADF test produces several values, such as ADF statistics (check[0]), p-value (check[1]), and critical values (check[4]). The function prints ADF statistics, p-value, and critical values. Based on the ADF statistical values, it is compared with critical values to decide whether the time series is stationary or not. If the ADF statistical value is greater than the critical value at a certain confidence level (for example, 1%), then the time series is considered non-stationary. Conversely, if the ADF statistical value is smaller, then the time series is considered stationary.

```
FUNCTION Stationary_check(dataframe)
   PRINT "ADF Statistic:", check[0]
   PRINT "p-value:", check[1]
   PRINT "Critical Values:"
   FOR key, value IN check[4]
      PRINT key, value (format tiga desimal)
   IF check[0] > check[4]["1%"]
      PRINT "Data Time Series is Non-Stationary"
   ELSE
      PRINT "Data Time Series is Stationary"

BEGIN
   List_df_ND ← daftar data differencing dari df_N1, df_N2, df_N3, df_N4

   PRINT "The Result"
   FOR each dataframe IN List_df_ND
      PRINT new line
      CALL Stationary_check(dataframe)
END
```

The results obtained are as follows.

```
Checking the transformed series for stationarity:
ADF Statistic: -15.23022618473042
p-value: 5.334088126188849e-28
Critical Values:
        1%: -3.431
        5%: -2.862
        10%: -2.567
Data Time Series is Stationary
```

GRU Model

The GRU model is built using the Keras Sequential API. Several GRU layers are stacked sequentially. Each GRU layer has 150 units, uses hyperbolic tangent (tanh) activation, and returns sequences (return_sequences=True). Dropout with a rate of 0.2 is applied after each GRU layer to prevent overfitting. Finally, a Dense layer with one unit is used as the output layer. The model was compiled using the Stochastic Gradient Descent (SGD) optimizer with momentum 0.9. The error function used is mean squared error (MSE). The model is trained using training data (X_Train, y_Train) with validation data (X_Test, y_Test). Training is carried out for 50 epochs with a batch size of 120. The EarlyStopping callback is used to stop training early if there is no significant increase in performance.

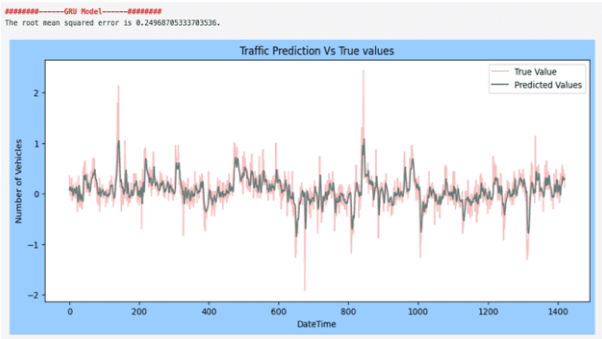
LSTM Models

The LSTM model is built using the Keras Sequential API. The LSTM layers are stacked sequentially. Each LSTM layer has 150 units, uses hyperbolic tangent (tanh) activation, and returns sequences (return_sequences=True). Dropout with a rate of 0.2 is applied after each LSTM layer to prevent overfitting. Finally, a Dense layer with one unit is used as the output layer. The model was compiled using the Stochastic Gradient Descent (SGD) optimizer with a momentum of 0.9 and a possible learning rate adjusted using lr_schedule. The loss function used is mean squared error (MSE).

The model is trained using training data (X_Train, y_Train) with validation data (X_Test, y_Test). Training is carried out for 50 epochs with a batch size of 120. The EarlyStopping callback is used to stop training early if there is no significant increase in performance.

Road 1 (Jl. Raden Intan)

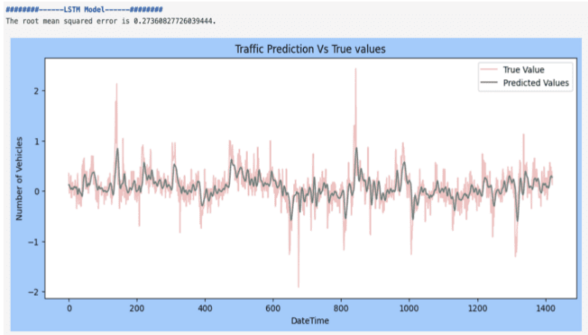
A comparison graph between predictions and actual values for road section 1 using the GRU model with SGD optimizer can be seen in Figure 12 dan GRU Model with Adam optimizer can be seen in Figure 13. From the prediction assessment using GRU and SGD, the RMSE value is 0.25, while the RMSE value with the Adam optimizer is 0.23.



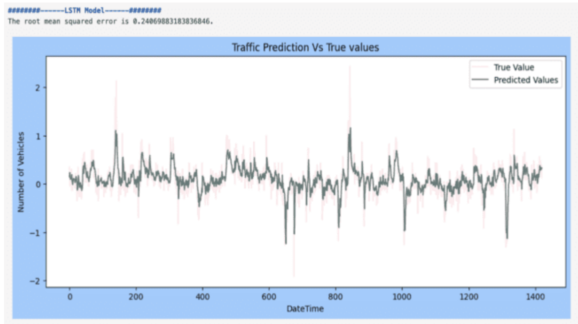
Source: (Research Results, 2025)
Graph of Comparison of Predicted and Actual Values on Road 1 Using the GRU Model with SGD Optimizers

Source: (Research Results, 2025)
Figure 13. Graph of Comparison of Predicted and Actual Values on Road 1 Using the GRU Model with Adam Optimizers

The comparison graph between the prediction and the actual value on road section 1 using the LSTM model with SGD optimizer can be seen in Figure 14 dan and LSTM Model with Adam optimizer can be seen in Figure 15. From the prediction assessment using LSTM, the RMSE value is 0.27, while the RMSE value with the Adam optimizer is 0.24.



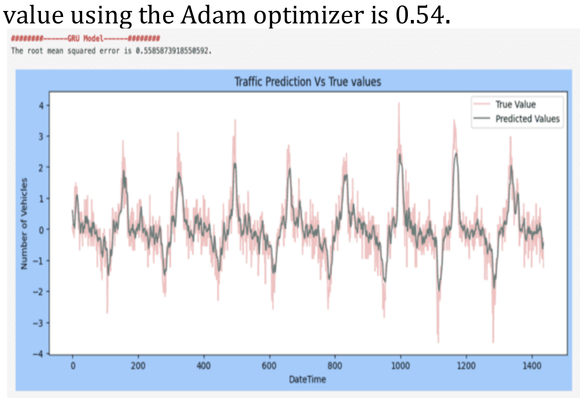
Source: (Research Results, 2025)
Graph Comparison of Predicted and Actual Values on Road 1 Using LSTM Model with SGD Optimizer



Source: (Research Results, 2025)
Figure 15. Graph Comparison of Predicted and Actual Values on Road 1 Using LSTM Model with Adam Optimizer

Road 2 Jl. Jendral Ahmad Yani

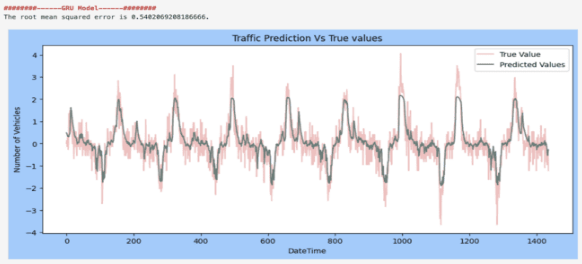
The comparison graph between the prediction and the actual value on road section 2 using the GRU model with SGD optimizer can be seen in Figure 16 and and GRU Model with Adam optimizer can be seen in Figure 17. From the prediction assessment using GRU, the RMSE value is 0.55, while the RMSE value using the Adam optimizer is 0.54.



Source: (Research Results, 2025)
Graph of Comparison of Predicted and Actual Values on Road 2 Using the GRU Optimizers

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

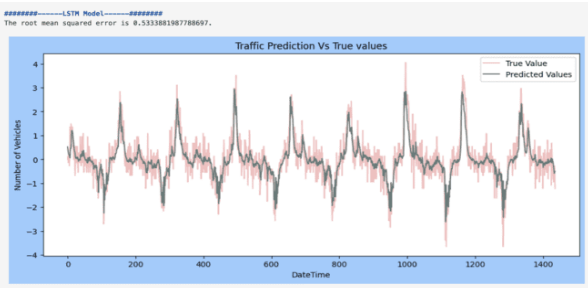
JITK (JURNAL ILMU PENGETAHUAN DAN TEKNOLOGI KOMPUTER)
VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6569



Source: (Research Results, 2025)
Graph of Comparison of Predicted and Actual Values on Road 2 Using the GRU Model with Adam Optimizers

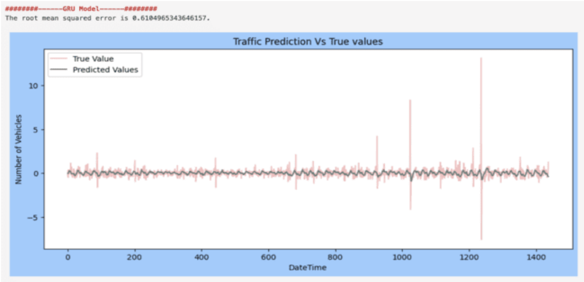
The comparison graph between the prediction and the actual value on road section 2 with the GRU model with SGD optimizer can be seen in Figure 18 and LSTM Model with Adam optimizer can be seen in Figure 19. From the prediction assessment using LSTM, the RMSE value is 0.59, while the RMSE value with the Adam optimizer is 0.53.

Source: (Research Results, 2025)
Graph of Comparison of Predictions and Actual Values on Road 2 Using the LSTM Model with SGD Optimizers

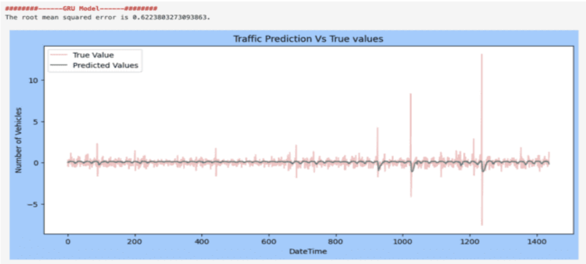


Source: (Research Results, 2025)
Graph of Comparison of Predictions and Actual Values on Road 2 Using the LSTM Model with Adam Optimizers

Road 3 Jl. Diponegoro
The comparison graph between the predicted and actual values on road section 3 using the GRU model with the SGD optimizer is presented in Figure 20, while that using the GRU model with the Adam optimizer is presented in Figure 21. Based on the prediction assessment using the GRU model, the RMSE value is 0.55, whereas the RMSE value with the Adam optimizer is 0.62.

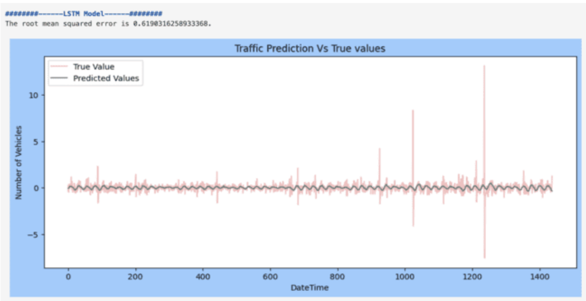


Source: (Research Results, 2025)
Comparison graph of predicted and actual values on Road 3 using the GRU model with SGD optimizers



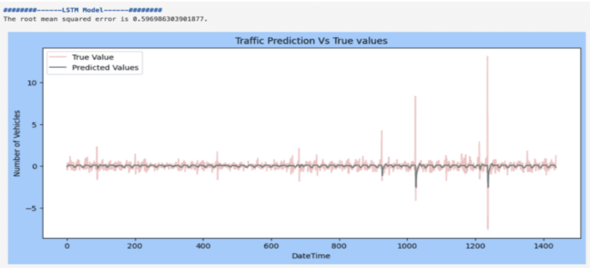
Source: (Research Results, 2025)
Comparison graph of predicted and actual values on Road 3 using the GRU model with Adam optimizers

The comparison graph between the predicted and actual values for road section 3 using the LSTM model with the SGD optimizer is presented in Figure 22, while the corresponding graph using the Adam optimizer is shown in Figure 23. The RMSE value obtained with the Adam optimizer is 0.59.



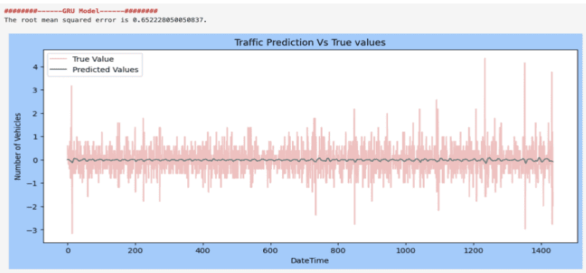
Source: (Research Results, 2025)
Figure 22. Comparison graph of predicted and actual values on Road 3 using the LSTM model with SGD Optimizers

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480 /jitk.v10i4.6569



Source: (Research Results, 2025)
Figure 23. Comparison graph of predicted and actual values on Road 3 using the LSTM model with Adam optimizers

Road 4 Jl. Sudirman
The comparison graph between the predicted and actual values for road section 4 using the GRU model with the SGD optimizer is **presented in** Figure 24, while the graph using the GRU model with the Adam optimizer is **shown in** Figure 25. Based on the prediction assessment using the GRU model, the RMSE value is 0.65, whereas the RMSE value obtained with the Adam optimizer is 0.59.

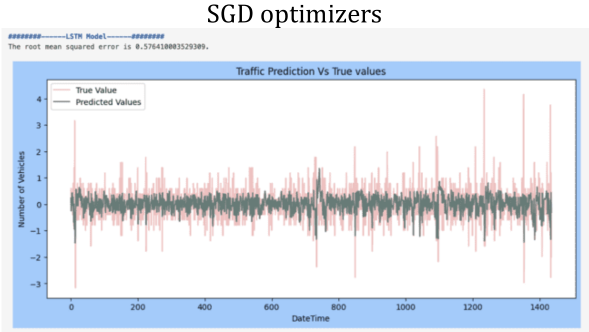


Source: (Research Results, 2025)
Figure 24. Graph of Comparison of Predictions and Actual Values on Road 4 Using the GRU Model with SGD Optimizers

Source: (Research Results, 2025)
Figure 25. Graph of Comparison of Predictions and Actual Values on Road 4 Using the GRU Model with Adam Optimizers

The comparison graph between the predicted and actual values for road section 4 using the LSTM model with the SGD optimizer is **presented in** Figure 26, while the graph using the LSTM model with the Adam optimizer is **shown in** Figure 27. Based on the prediction assessment using the LSTM model, the RMSE value is 0.65 with the SGD optimizer and 0.57 with the Adam optimizer.

Source: (Research Results, 2025)
Figure 26. Comparison graph of predicted and actual values on Road 4 using the LSTM model with SGD optimizers



Source: (Research Results, 2025)
Figure 27. Comparison graph of predicted and actual values on Road 4 using the LSTM model with Adam optimizers

Results of Analysis

In this research, traffic was identified on 4 road sections at Simpang Tugu Adipura, Bandar Lampung City. The first road section is Jl. Raden Intan, road two is Jl. General Ahmad Yani, road section three is Jl. Diponegoro, and road section four is Jl. Sudirman. These four road sections have different characteristics. Jl. General Sudirman and Jl. Diponegoro has 4 divided 2-way lanes, while Jl. Raden Intan and Jl. Ahmad Yani has 3 undivided 1-way lanes. The highest traffic is on the Jl. Raden Intan, this is because this road has many trading areas and money services spread along the road. Apart from that, this road also accommodates movement towards the Tugu Adipura Roundabout. Vehicles using this road can go to residential areas, trade and service areas, central office areas, educational areas, as well as Pesawaran Regency. Jl. General Ahmad Yani has the second highest traffic. Jl. General Ahmad Yani accommodated the

Accredited Rank 2 (Sinta 2) based on the Decree of the Dirjen Penguatan RisBang Kemenristekdikti No.225/E/KPT/2022, December 07, 2022. Published by LPPM Universitas Nusa Mandiri

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6569

movement to Tanjung Karang District. The high traffic on these two road sections means that the lanes are made in one direction. Next Jl. Jenderal Sudirman, has the third highest traffic. Based on its land use, this road accommodates movement to the office center and Pesawaran Regency. Meanwhile Jl. Jend Sudirman has the least traffic. This is because this road only accommodates movement to areas that are not diverse.

From the comparison of RMSE values for the four road sections, it was found that the lowest RMSE value occurred on road section 1 using the GRU model using Adam Optimizer, with an RMSE value of 0.23. The number of vehicles on road section one has increased more rapidly compared to roads two, three and four. This is because the movement of various kinds of activities/activities is concentrated through this road. Traffic on road section one has stronger weekly and daily seasonality compared to other road sections at the Tugu Adipura intersection because this road accommodates work and school activities on weekdays and shopping and entertainment activities on weekends. These various activities influence traffic trends because each activity is carried out at different hours. Meanwhile, other road sections have a more significant and linear trend, this is because there are not many activities accommodated by these three roads.

The superior performance of the GRU model, particularly on Road 1 (Jl. Raden Intan), can be attributed to its simpler architecture which facilitates faster convergence and reduces the risk of overfitting when dealing with highly dynamic time series. GRU's gating mechanism, which consists only of an update gate and reset gate, enables it to selectively retain or discard information with lower computational cost, making it more responsive to short-term fluctuations commonly observed in roads with high temporal variability. This result is consistent with previous findings by Dey et al. [10], who showed that GRU outperformed LSTM in forecasting time series with high-frequency fluctuations due to its ability to capture local temporal structures more efficiently. In contrast, Karyadi and Santoso [9] observed better performance of LSTM in air quality prediction, where long-term dependencies were more dominant and smoother temporal patterns existed. In this study, the GRU model demonstrated an advantage in cases where traffic exhibited strong seasonal patterns with frequent peaks and troughs—such as on Road 1, which is influenced by weekday commuting and weekend leisure activities. The results also indicate that LSTM models, although more robust in some scenarios, were slightly less responsive to rapid changes in traffic volume. These findings emphasize the importance of matching model architecture to the characteristics of the data, and demonstrate that GRU can serve as an effective alternative in urban traffic forecasting, particularly when high-frequency seasonality is present.

CONCLUSION

The test results show that the LSTM model produced slightly lower RMSE values than the GRU model on most road sections: 0.24 for LSTM and 0.27 for GRU on Road 1, 0.558 for LSTM and 0.589 for GRU on Road 2, 0.610 for LSTM and 0.691 for GRU on Road 3, and 0.652 for LSTM and 0.654 for GRU on Road 4. Although LSTM demonstrated a marginally better performance overall, the GRU model achieved the best result specifically on Road 1 when trained with the Adam optimizer, attaining an RMSE of 0.23—the lowest among all models and configurations. This result highlights GRU's strength in capturing traffic dynamics on roads with high-frequency seasonal patterns and strong short-term fluctuations, such as Road 1 (Jl. Raden Intan), which is influenced by diverse urban activities.

GRU is also favored for its simpler architecture and faster training time, making it well-suited for real-time applications and resource-constrained environments. While LSTM can deliver slightly better accuracy in some instances, GRU's computational efficiency and responsiveness provide distinct practical benefits, especially when traffic patterns change rapidly within short time frames. Future work can build upon this study by integrating external variables such as weather conditions, public holidays, and real-time road events to improve predictive performance. Additional validation across various intersections or cities could enhance generalizability. Further exploration of hybrid deep learning models—such as LSTM-CNN or attention-based architectures—may yield better pattern recognition. Testing alternative optimizers like RMSProp or AdaGrad could also contribute to more effective convergence. Ultimately, deploying these models in real-time traffic systems may enable early warning mechanisms and support data-driven decisions in urban transport management.

REFERENCE

[1] P. Sun, Q. Yu, and K. You, "Intelligent traffic management strategy for traffic congestion in underground loop," *Tunn. Undergr. Sp. Technol.*, vol. 143, no. March 2023, p.

105509, 2024, doi: 10.1016/j.tust.2023.105509.

[2] M. Sasidharan, M. E. Torbaghan, Y. Fathy, C. D. F. Rogers, N. Metje, and J. Schooling, "Designing user-centric transport strategies for urban road space redistribution," *Commun. Transp. Res.*, vol. 3, no. June 2023, p. 100109, 2023, doi: 10.1016/j.commtr.2023.100109.

[3] Q. L. Jing, H. Z. Liu, W. Q. Yu, and X. He, "The Impact of Public Transportation on Carbon Emissions—From the Perspective of Energy Consumption," *Sustain.*, vol. 14, no. 10, pp. 1–18, 2022, doi: 10.3390/su14106248.

[4] R. Tian, C. Wang, J. Hu, and Z. Ma, "Multi-scale spatial-temporal aware transformer for traffic prediction," *Inf. Sci. (Ny).*, vol. 648, no. February, p. 119557, 2023, doi: 10.1016/j.ins.2023.119557.

[5] S. Feng *et al.*, "A macro–micro spatio-temporal neural network for traffic prediction," *Transp. Res. Part C Emerg. Technol.*, vol. 156, no. February, p. 104331, 2023, doi: 10.1016/j.trc.2023.104331.

[6] R. T. Theodora, "Kajian Teknis Terhadap Kelayakan Bundaran di Kota Bandarlampung," Universitas Lampung, 2022.

[7] T. A. Prasetyo *et al.*, "Evaluating the efficacy of univariate LSTM approach for COVID-19 data prediction in Indonesia," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 34, no. 2, pp. 1353–1366, 2024, doi: 10.11591/ijeecs.v34.i2.pp1353-1366.

[8] R. Cahuantzi, X. Chen, and S. Güttel, "A Comparison of LSTM and GRU Networks for Learning Symbolic Sequences," *Lect. Notes Networks Syst.*, vol. 739 LNNS, pp. 771–785, 2023, doi: 10.1007/978-3-031-37963-5_53.

[9] Y. Karyadi, "Prediksi Kualitas Udara Dengan Metoda LSTM, Bidirectional LSTM, dan GRU," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 1, pp. 671–684, 2022, doi: 10.35957/jatisi.v9i1.1588.

[10] P. Dey *et al.*, "Comparative analysis of recurrent neural networks in stock price prediction for different frequency domains," *Algorithms*, vol. 14, no. 8, pp. 1–20, 2021, doi: 10.3390/a14080251.

[11] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," *Chaos, Solitons and Fractals*, vol. 140, p. 110212, 2020, doi: 10.1016/j.chaos.2020.110212.

[12] A. Nilsen, "Perbandingan Model RNN, Model LSTM, dan Model GRU dalam Memprediksi Harga Saham-Saham LQ45," *J. Stat. dan Apl.*, vol. 6, no. 1, pp. 137–147, 2022, doi: 10.21009/jsa.06113.

[13] K. D. Prasetyo, R. Wijaya, and G. S. Wulandari, "Comparative Analysis of ARIMA and LSTM Models for Predicting Physical Fatigue in Bandung Workers," *J. Media Inform. Budidarma*, vol. 8, no. 1, p. 528, 2024, doi: 10.30865/mib.v8i1.7282.

[14] X. Wu, L. Chen, J. Zhao, M. He, and X. Han, "CNN-GRU-Attention Neural Networks for Carbon Emission Prediction of Transportation in Jiangsu Province," *Sustain.*, vol. 16, no. 19, pp. 1–21, 2024, doi: 10.3390/su16198553.

[15] W. Yulita, M. C. Untoro, M. Praseptiawan, I. F. Ashari, A. Afriansyah, and A. N. Bin Che Pee, "Automatic Scoring Using Term Frequency Inverse Document Frequency Document Frequency and Cosine Similarity," *Sci. J. Informatics*, vol. 10, no. 2, pp. 93–104, 2023, doi: 10.15294/sji.v10i2.42209.

[16] I. F. Ashari, E. D. Nugroho, R. Baraku, I. N. Yanda, and R. Liwardana, "Analysis of Elbow , Silhouette , Davies-Bouldin , Calinski-Harabasz , and Rand-Index Evaluation on K-Means Algorithm for Classifying Flood-Affected Areas in Jakarta," vol. 7, no. 1, pp. 95–103, 2023.

[17] I. F. Ashari, "Analysis Sentiments In Facebook Down Case Using Vader And Naive Bayes Classification Method," *Multitek Indones. J. Ilm.*, vol. 16, no. 2, pp. 75–89, 2022.

[18] Muttaqin, Yuswardi, A. Maulidinnawati, A. Parewe, I. F. Ashari, and M. Munsarif, *Pengantar Sistem Cerdas*. 2023.

[19] H. Niu, Z. Zhang, Y. Xiao, M. Luo, and Y. Chen, "A Study of Carbon Emission Efficiency in Chinese Provinces Based on a Three-Stage SBM-Undesirable Model and an LSTM Model," *Int. J. Environ. Res. Public Health*, vol. 19, no. 9, 2022, doi: 10.3390/ijerph19095395.

[20] A. M. Nassef, A. G. Olabi, H. Rezk, and M. A. Abdelkareem, "Application of Artificial Intelligence to Predict CO2 Emissions: Critical Step towards Sustainable Environment," *Sustain.*, vol. 15, no. 9, 2023, doi: 10.3390/su15097648.

[21] W. Hastomo, N. Aini, A. S. B. Karno, and ..., "Machine Learning Methods for Predicting

VOL. 10. NO. 4 MAY 2025
P-ISSN: 2685-8223 | E-ISSN: 2527-4864
DOI: 10.33480/jitk.v10i4.6569

Manure Management Emissions," *J. …*, vol. 11, no. 2, pp. 131–139, 2022, [Online]. Available: http://download.garuda.kemdikbud.go.id/article.php?article=2807435&val=24806&title=Metode Pembelajaran Mesin untuk Memprediksi Emisi Manure Management.

[22] R. R. Elhakim, "Prediksi Nilai Tukar Rupiah Ke Dollar As Menggunakan Metode Arima," *MATHunesa J. Ilm. Mat.*, vol. 8, no. 2, pp. 145–150, 2020, doi: 10.26740/mathunesa.v8n2.p145-150.

[23] Y. Yundari, R. M. Syahfutri, N. M. Huda, S. A. Antaristi, and R. Jonathan, "Pemodelan Autoregresif dengan Error Berkorelasi Waktu untuk Data Covid-19 Kasus Pasien Terkonfirmasi di Kalimantan Barat," pp. 199–204, 2021, doi: 10.26418/pipt.2021.37.