SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 7, No. 2, 2024, 156-172
DOI: 10.35377/saucis...1404116

RESEARCH ARTICLE

Prediction of Multivariate Chaotic Time Series using GRU, LSTM and RNN

Gülyeter Öztürk[1], Osman Eldoğan[1]

[1]Department of Mechatronics Engineering, Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, Türkiye

Corresponding author:
Gülyeter Öztürk, Department of Mechatronics Engineering, Faculty of Technology, Sakarya University of Applied Sciences, Sakarya, Türkiye
gulyeterozturk@subu.edu.tr

Article History:
Received: 12.12.2023
Accepted: 22.07.2024
Published Online: 23.08.2024

ABSTRACT
Chaotic systems are identified as nonlinear, deterministic dynamic systems that are exhibit sensitive to initial values. Some chaotic equations modeled from daily events involve time information and generate chaotic time series that are sequential data. Through successful prediction studies conducted on the generated chaotic time series, forecasts can be made about events displaying unpredictable behavior in nature, which have not yet been modeled. This enables preparation for both favorable and unfavorable situations that may arise. In this study, chaotic time series were generated using Lorenz, Chen, and Rikitake multivariate chaotic systems. To enhance prediction accuracy on the generated data, GRU, LSTM and RNN models were trained with different hyperparameters. Subsequently, comprehensive test studies were conducted to evaluate their performance. Predictions were calculated using evaluation metrics, including MSE, RMSE, MAE, MAPE, and R^2. In the experimental study, each chaotic system was trained with different hyperparameter combinations on six network models. The experimental results indicate that the utilized models exhibited greater success in predicting chaotic time series compared to some other models in the literature.

Keywords: Chaotic time series, Multivariate, Time series prediction, GRU, LSTM, RNN

1. Introduction

Dynamical systems that seem complex in our daily lives, yet possess internal order and are sensitive to initial conditions, are defined as chaotic systems, and these systems produce chaotic data. Sequential chaotic data contains time information and is commonly termed chaotic time series. One of the important topics that scientists focus on is conducting prediction studies based on examining past and present values of a system in a time series. Time series prediction studies are applied in real-world domains with chaotic structures, such as traffic flow [1], building energy consumption [2], finance [3], electrical load [4], meteorology [5], earthquake [6], and wind energy [7]. The success achieved in predicting challenging events like this one, along with other seemingly complex phenomena, can render many occurrences in nature foreseeable and controllable.

In studies aimed at predicting time series of linear systems, traditional statistical methods such as Autoregressive Moving Average (ARMA), Autoregressive Moving Average with Exogenous Inputs (ARMAX), and Autoregressive Integrated Moving Averages (ARIMA) were previously employed. However, these methods are found to be insufficient when dealing with nonlinear complex systems [8], [9]. As a response, researchers have explored machine learning, deep learning, and hybrid methods to predict data from nonlinear and complex systems. In their study on chaotic time series prediction using both the noisy and noiseless Lorenz system, Karunasinghe and Liong found that artificial neural network models outperformed local prediction models [10]. Yuxia and Hongtao attempted to predict the time series data of the Lorenz system using a support vector machine with a chaos optimization algorithm. In this study focused on predicting the x state variable of the Lorenz chaotic system, the researchers obtained a root mean square error (RMSE) value of 0.0030335 [11]. The authors, utilizing a NARX neural network in MATLAB, employed 2100 time series data generated from the Lorenz chaotic system for prediction, with RMSE as the evaluation metric [12]. In prediction studies of multivariate chaotic time series, using all variables that constitute the system, rather than employing a single variable, enables obtaining more information about the dynamic system. In their studies, Xiu and Zhang found, in both single and multiple variable analyses, that more accurate predictions were achieved using multiple variables. Addressing the significance of predicting time series data in economics, business, and finance, Siami and Namin demonstrated in their study that the Long Short-Term Memory (LSTM) deep learning model outperforms the ARIMA model [13].

Deep learning models, known for their superior ability to capture nonlinear relationships in large datasets compared to traditional machine learning methods, have gained widespread popularity in time series prediction studies in recent years.

Due to the inherent time relationship in time series data, where the current data point is connected to both past and future data points, the Recurrent Neural Network (RNN) deep learning model, designed with memory capabilities, is commonly utilized to establish and maintain these relationships. RNN and its variations have been predominantly utilized in time series prediction studies across various fields, including finance, energy, solar radiation, and air quality, in different years [2], [3], [14-16]. The frequency of usage of RNN variations in these studies follows the order of LSTM, ELMAN (simple RNN cell) [17], and Gated Recurrent Units (GRU) [18], respectively. Sezer and his colleagues reported that over half of the publications in the field of finance from 2005 to 2019 utilized RNN and its variations for time series prediction. The utilization rates of models in prediction studies in the field of finance are 9.89% for GRU, 29.7% for ELMAN (vanilla RNN), and 60.4% for LSTM [3]. Dudukcu and his team stated in their literature review that Elman RNN, LSTM, Convolutional Neural Networks (CNN), and Temporal Convolutional Networks (TCN) are commonly used in time series prediction studies. They created a dataset for each of the x, y, and z state variables of Lorenz, Rossler, and Lorenz-like chaotic systems. They also possess real-world data in the form of an electrocardiogram dataset from 21 patients. They carried out time series predictions using hyperparameters determined through the grid search method for the entire dataset [9]. Researchers, evaluating the performance of different optimized RNN cell structures on various time series datasets, have introduced a new RNN variation called SLIM, demonstrating cost-effectiveness in terms of both time and computational resources for prediction studies [14]. Chandra and Zhang employed two distinct methods for the cooperative evolution of Elman RNN in predicting the chaotic time series of Mackey-Glass, Lorenz and Sunspot [19]. In the conducted study, the obtained values are as follows: in the Lorenz system, 0.00636 RMSE and 0.000772 normalized mean square error (NMSE); in the Mackey-Glass system, 0.00633 RMSE and 0.000279 NMSE; and in the Sunspot time series, 0.0166 RMSE and 0.00147 NMSE. The authors also demonstrated that their two implemented methods yielded superior results compared to other methods, with the exception of the Evolutionary RNN and Hybrid-NARX-Elman models. In 2020, researchers utilized a hybrid model named Complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN)-LSTM to predict the Lorenz-63 chaotic time series [20]. Utilizing a total of 5,000 data points for training and testing, along with the Adam optimizer and ReLU activation function parameters, they achieved an RMSE of 1.327 and mean absolute error (MAE) of 1.124 with their proposed model. The authors performed predictions using Support Vector Regression, ARIMA, Multilayer Perceptron, and single LSTM models with the same dataset, revealing that their proposed model exhibited superior performance.

Dudukcu and her colleagues employed four different variations of the LSTM model to predict time series data derived from three chaotic systems, which are namely Lorenz, Rossler, and Lorenz-like systems. The study utilized the grid search algorithm for hyperparameter optimization, revealing that the highest success was achieved with an RMSE value of 3.7397×10^{-5} during the validation phase of the Stacked LSTM, and an RMSE value of 0.1558 during the testing phase of the Stacked LSTM [21]. Fu and colleagues developed a hybrid model called DTIGNet, incorporating an improved temporal inception module and GRU for automatic multi-scale feature extraction. They applied DTIGNet to the Mackey-Glass, Rossler, and Lorenz chaotic systems and sunspots time series to assess its efficacy in chaotic time series forecasting. Using metrics such as MAE, RMSE, correlation coefficient (ρ), coefficient of determination (R^2), mean absolute percentage error (MAPE), and SMAPE, the authors stated that their model demonstrated higher accuracy and better performance compared to other estimation methods [22]. In a related study, Cheng and team outperformed LSTM, CNN-LSTM, and TCN models in terms of RMSE, MAE, R^2, and ρ metrics with their designed TCN-CBAM model. They applied this model to the Lorenz system, Chen system, and sunspots dataset, achieving better performance in chaotic time series prediction [23].

When identical values are provided as inputs to mathematical models derived from natural events, the expectation is that the same results will always be produced. Given as an input to the mathematical model of chaotic systems, when the value is changed by one thousandth, a significant change occurs in the result obtained. Therefore, it is crucial to understand the method and precision with which time series are generated from chaotic systems. In the proposed study, the Lorenz system, considered the starting point in terms of chaos theory and utilized in almost all chaotic time series forecasting studies, was employed. Additionally, the Lorenz-like Chen chaotic system and the non-Lorenz-like Rikitake chaotic system were used to enhance the validity of the study. These chaotic systems were solved using the fourth-order Runge-Kutta method with a time step of 0.01. Examining numerous studies on chaotic time series prediction in the literature, it has been observed that only one of the state variables constituting the system is provided as input to the prediction models, and this variable is predicted as output. However, to make accurate predictions about a complex and unpredictable system, it is necessary to consider all variables that influence the system. This approach allows for more accurate prediction or classification results. In this study, all state variables constituting the chaotic systems are given as input to the prediction models, in an attempt to predict a single state variable with better performance. Chaotic systems trained and tested on GRU, RNN, and LSTM models with different hyperparameters were compared using the following evaluation metrics: mean square error (MSE), RMSE, MAE, MAPE, and R^2.

The second section of this study offers a detailed explanation of the deep learning models employed, and information about the datasets obtained from chaotic systems. The third section presents the experimental results and analysis of prediction studies conducted with various hyperparameters. In the final section, the inferences derived from the experiments are stated, and recommendations for future studies are provided.

2. Methodology and Methods

2.1. Prediction Models Used

In this section, we introduce RNN, LSTM, and GRU deep learning models designed to maintain the connection between chaotic time series data, ultimately yielding successful outcomes in predicting future data.

2.1.1. RNN

RNN maintains a connection with future data by retaining information from past steps in its memory. The RNN, which has short-term memory, is successful in remembering the past and predicting the future when given short sequential input data. However, it fails to remember the past and encounters difficulty in predicting future data when given long sequential input data. The reason for this is the emergence of the vanishing gradient problem, where the gradient value diminishes significantly during backpropagation and leads to its disappearance, or the occurrence of the exploding gradient problem, where the gradient value increases excessively during backpropagation, preventing convergence to local minimum errors. To address these fundamental challenges hindering learning in RNN, LSTM and GRU models have been developed for natural language processing, speech recognition, language translation, text generation, and time series classification/prediction tasks involving sequential data. The LSTM and GRU models originate from the recurrent neural network architecture and have demonstrated success in predicting both short-term and long-term time series. The structure of the RNN is shown in Figure 1.

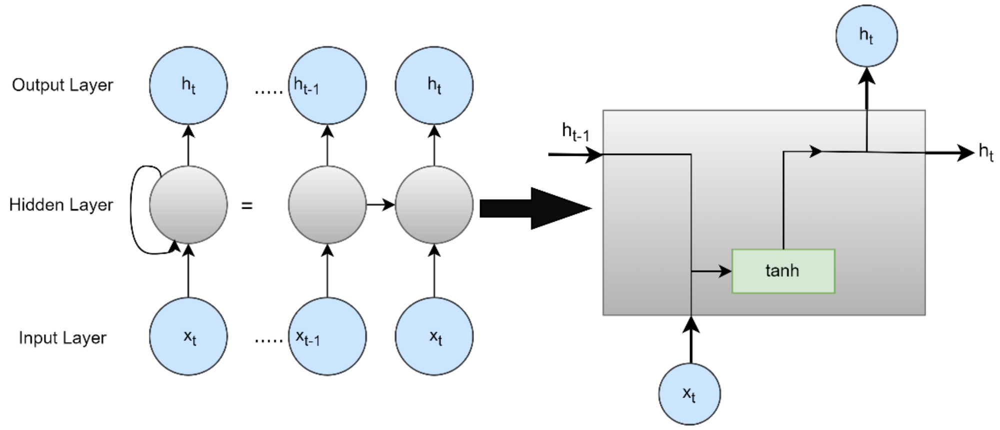

Figure 1. Structure of the RNN

2.1.2. LSTM

Each LSTM cell comprises a cell state, an input gate, a forget gate, and an output gate [24]. Temporal dependencies between dynamic nonlinear data are preserved thanks to the functions performed by the gates that determine which of the information given to the cell should be stored or forgotten. The cell state carries meaningful information received from the gates across the cells. In the LSTM architecture shown in Figure 2, each cell takes the current input (x_t), for time step t, as well as the previous cell state (c_{t-1}), and the previous hidden state (h_{t-1}). It generates a new cell state (c_t) and hidden state (h_t) at the cell output. The forget gate processes the hidden state information (h_{t-1}) from the previous cell and the current information (x_t) through the sigmoid activation function. Information close to 0 is forgotten, while information close to 1 is retained in the cell state. The input gate updates the cell state at the output by processing the previous hidden state information (h_{t-1}) and the current information (x_t) through sigmoid and hyperbolic tangent activation functions. The output gate generates the hidden state for the next LSTM cell. At the output gate, the previous hidden state information (h_{t-1}) and current information (x_t) pass through the sigmoid function, and the information from the cell state passes through the hyperbolic tangent function. By multiplying the results of both functions, the hidden state information of the next cell (h_t) is obtained. Both the hidden state and cell state contain information about previous inputs and are utilized in prediction studies. The operations performed in the LSTM cell are provided in Equation 1.

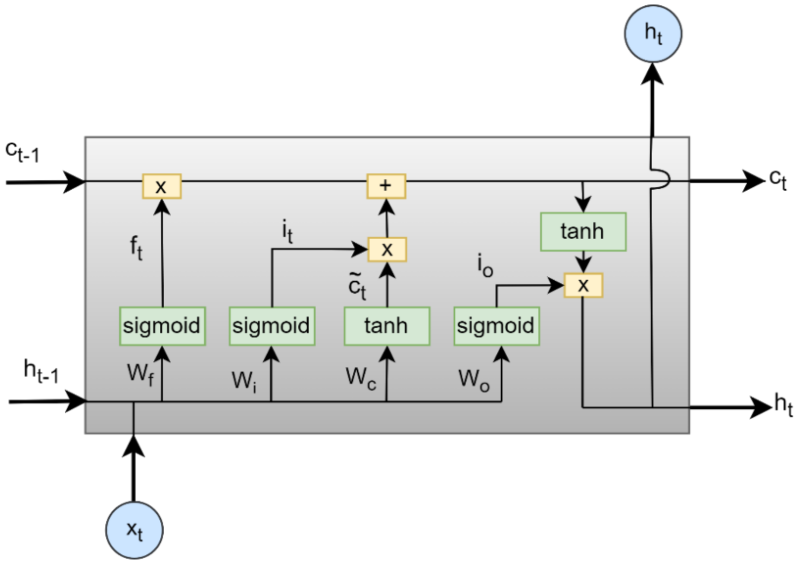

Figure 2. The Internal Structure of an LSTM Cell

$$
\begin{gathered}
\boldsymbol{f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f)} \\
\boldsymbol{i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i)} \\
\boldsymbol{\widetilde{c}_t = tanh\,(W_c.[h_{t-1}, x_t] + b_c} \\
\boldsymbol{c_t = f_t * c_{t-1} + i_t * \widetilde{c}_t} \\
\boldsymbol{o_t = \sigma(W_o.[h_{t-1}, x_t] + b_0)} \\
\boldsymbol{h_t = o_t * tanh(c_t)}
\end{gathered} \quad (1)
$$

Here, x_t denotes the data at time step t; f_t, i_t, o_t denote the forget, input and output gates, respectively, and h_{t-1} refers to the previous cell output. W_f, W_i, W_c, W_o denote the weights, while b_f, b_i, b_c, b_o denote the bias terms and σ denotes the sigmoid function.

2.1.3. GRU

GRU [18] is similar to LSTM and incorporates reset and update gates. With no cell state in its structure, GRU uses only the hidden state information of the previous cell to transfer information, thus reducing the computational cost. In the GRU cell depicted in Figure 3; the update gate determines which information to discard and which new information to retain, while the reset gate determines how much of the past information will be forgotten. When a data is input to the model, the mathematical operations taking place within the cell are provided in Equation 2.

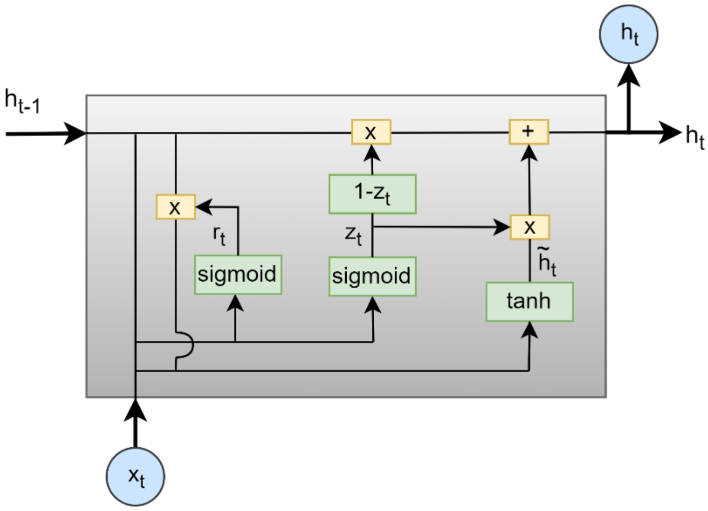

Figure 3. The Internal Structure of a GRU Cell

$$
\begin{gathered}
\boldsymbol{z_t = \sigma(W_z.[h_{t-1}, x_t])} \\
\boldsymbol{r_t = \sigma(W_r.[h_{t-1}, x_t])} \\
\boldsymbol{\widetilde{h_t} = tanh(W.[r_t * h_{t-1}, x_t])} \\
\boldsymbol{h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t}}
\end{gathered} \quad (2)
$$

Here, x_t represents the data at time step t; while r_t, z_t denote the reset and update gates, respectively. Additionally, h_{t-1} refers the output of the previous cell, and W_r, W_z, W denote the weights.

2.2. Chaotic Systems and Datasets

2.2.1. Multivariate Lorenz Chaotic Time Series

The Lorenz chaotic system, widely accepted as the beginning of chaotic systems, was introduced by Edward Lorenz in 1963 and has been employed in modeling atmospheric conditions [25]. Given its widespread use in classification and prediction studies in the literature, we employed the Lorenz chaotic system, described by Equation 3, to validate our proposed methods. The parameters of the system consisting of 3 ordinary differential equations are set as σ = 10, ρ = 28, β = 8/3, and the initial conditions for the state variables are chosen as (x_o, y_o, z_o)=(0,-0.1,9). 5,000 data points were derived from the solution of the chaotic system using the fourth order Runge-Kutta method with a time step value set at 0.001. The models performing predictions were trained with 4,000 data points and tested with 1,000 data points. The obtained data is depicted in Figure 4.

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= x(\rho - z) - y \\ \dot{z} &= xy - \beta z \end{aligned} \tag{3}$$

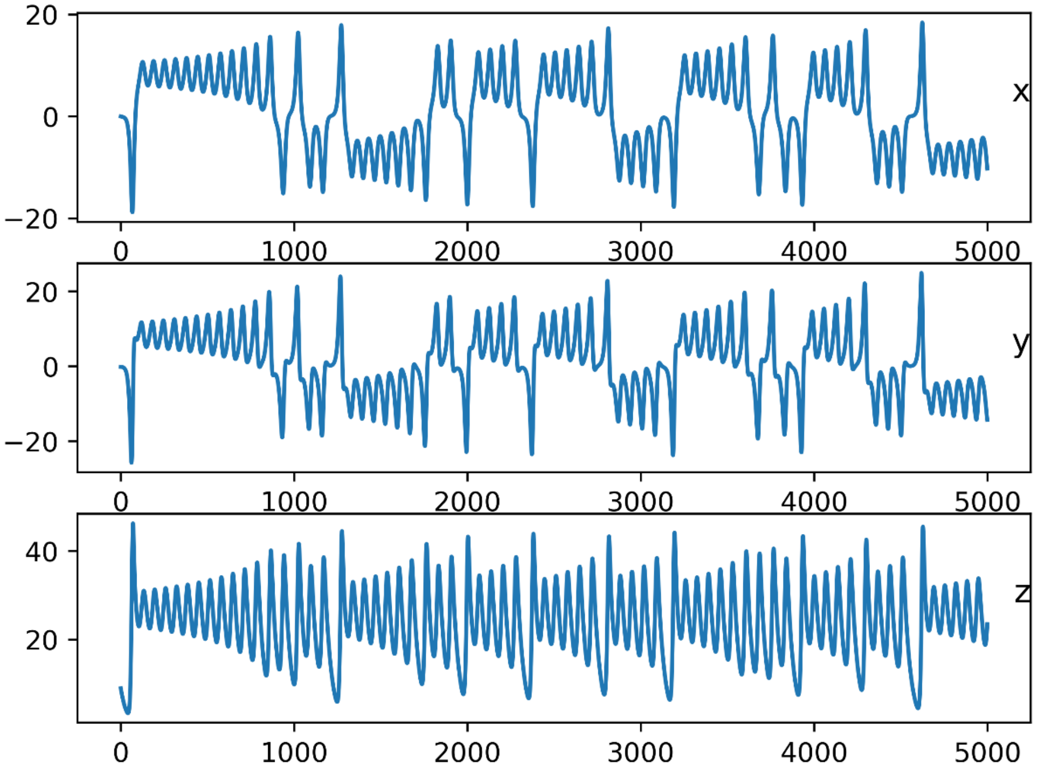

Figure 4. Multivariate Lorenz Chaotic Time Series

2.2.2. Multivariate Chen Chaotic Time Series

The Chen chaotic system [26], consisting of 3 ordinary differential equations and presented to the scientific world by Guanrong Chen and Ueta in 1999, is given in Equation 4.

$$\begin{aligned} \dot{x} &= a(y - x) \\ \dot{y} &= (c - a)x - xz + cy \\ \dot{z} &= xy - bz \end{aligned} \tag{4}$$

The system parameters are defined as a = 35, b = 3, c = 28, with initial values for the state variables set as $x_o = -10$, $y_o = 0$, $z_o = 37$. Employing a time step value of 0.01 and utilizing the fourth order Runge-Kutta method, 5,000 data points have been derived from solving the chaotic system. During the training phase, 4,000 data points were utilized, and during the testing phase, 1,000 data points were used. The data obtained is shown in Figure 5. In this study, the Chen system, which is particularly similar to the Lorenz system, was selected. Thus, differences in prediction results of similar systems in the conducted study will be observed.

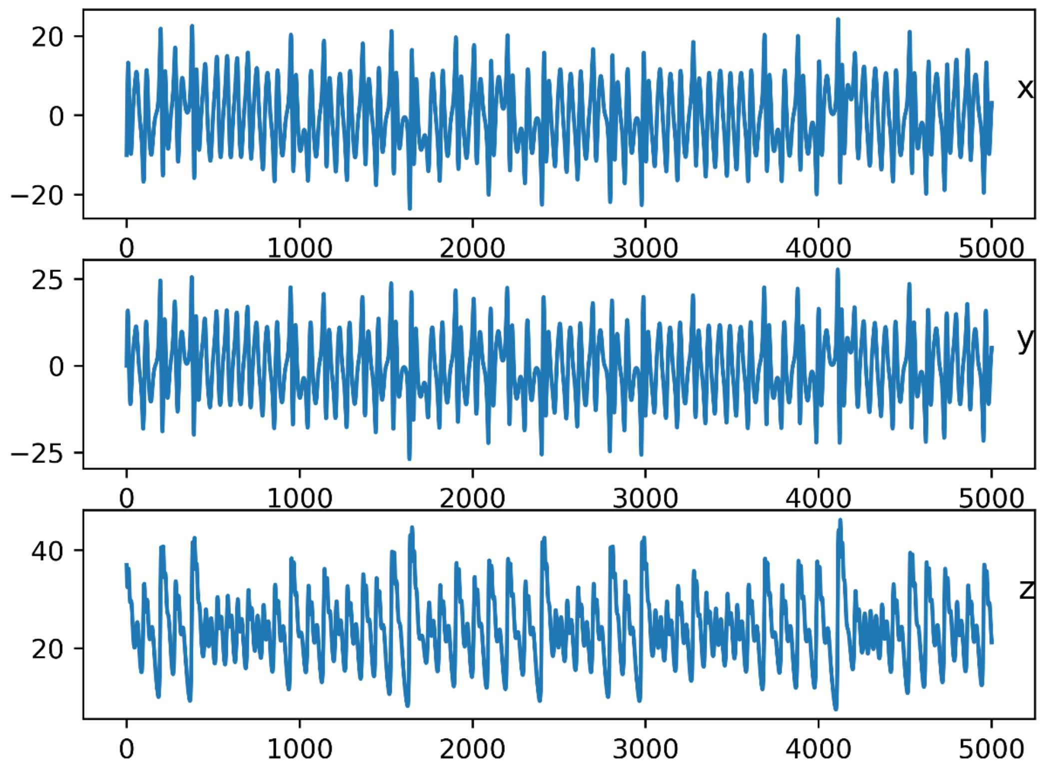

Figure 5. Multivariate Chen Chaotic Time Series

2.2.3. Multivariate Rikitake Chaotic Time Series

The Rikitake system [27], [28], employed in the fields of hydrodynamics and electromagnetic phenomena, elucidates the flow of liquid metal within a magnetic field. The complex behaviors arising from the influence of the magnetic field and the temporal variations imbue this system with chaotic characteristics. The Rikitake system has been utilized to investigate how a system, distinct from the Lorenz and Chen systems, would manifest changes under the same hyperparameter values. The system consists of three ordinary differential equations, which are provided in Equation 5.

$$\begin{aligned} \dot{x} &= -\mu x + yz \\ \dot{y} &= -\mu x + (z - a)x \\ \dot{z} &= 1 - xy \end{aligned} \tag{5}$$

The system parameters are set as μ = 2 and a = 5, with initial values for the state variables defined as $x_o = 3$, $y_o = 1$, $z_o = 6$. Employing a time step value of 0.01 and utilizing the fourth order Runge-Kutta method, a total of 20,000 data points have been obtained from solving the chaotic system. The initial 4,000 data points were discarded, and prediction studies were conducted on the remaining 16,000 data points. The resulting 20,000 data points are depicted in Figure 6.

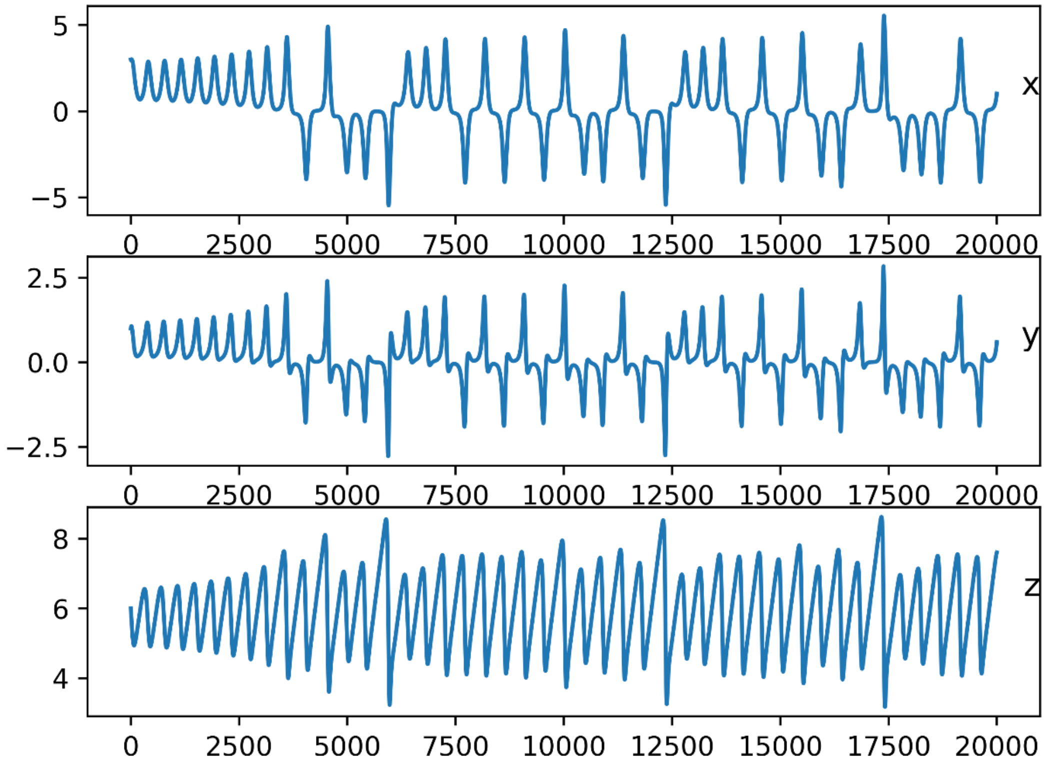

Figure 6. Multivariate Rikitake Chaotic Time Series

2.3 Evaluation Metrics

In this study, metrics such as MSE, RMSE, MAE, MAPE, and R^2 have been employed to measure the error between the predicted and actual values of the chaotic time series. The RMSE provides insights into the generalization capability of prediction models, while MSE, MAE, and MAPE represent the stability of the model. A smaller RMSE signifies enhanced predictive prowess, and lower MAE, MSE, and MAPE values denote greater model stability. R^2 reflects the correlation between the model and the data. Optimal prediction models strive for values close to 0 for RMSE, MSE, MAE, and MAPE and close to 1 for R^2 [29]. The evaluation metrics are as seen in Equation 6.

$$\begin{gathered}
MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \\
RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \\
MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \\
MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \\
R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \qquad \bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}
\end{gathered} \tag{6}$$

Here, n is the total number of data points yielded from the chaotic system, y_i is the actual data values, and $\hat{y}_i$ is the predicted values.

3. Experiment and Result Analysis

The study utilized the Python programming language, along with the Tensorflow, Keras, Scikit-Learn, Pandas, and NumPy libraries, executed within a Jupyter notebook environment. The datasets, derived from multivariate chaotic systems, were partitioned into 80% for training and 20% for testing. The training and testing processes employed the Adaptive Moment Estimation (Adam) optimizer throughout.

The data obtained from multivariate chaotic systems is initially normalized between 0 and 1, and then provided as input to the models in the form of (S, T, F), where S (samples) indicates the number of data rows given to the model, T (time steps) represents how many steps ahead the future data will be predicted, and F (features) denotes the number of state variables included in the data. These two processes are part of the data preprocessing in the chaotic time series prediction flowchart shown in Figure 7. The model layer involves one or two layers of RNN, LSTM or GRU models. The dense layer predicts the y state variable for the Lorenz system and the x state variables for the Chen and Rikitake systems. The estimated variables are later evaluated using assessment metrics.

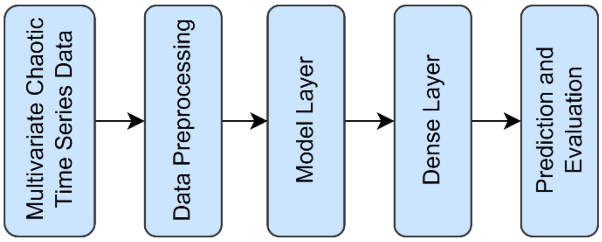

Figure 7. Flowchart of Chaotic Time Series Prediction

Data obtained from chaotic systems were trained on single-layer GRU, LSTM, and RNN models, as well as on two-layer GRU-GRU, LSTM-LSTM, and RNN-RNN models. For the time series forecasting study, it was decided to use the hyperparameters specified in the following tables as a result of the literature research. The models are trained with all hyperparameter combinations given in the tables. In the study aiming to predict the x_t state variable of the Chen chaotic system, each model was trained a total of 960 times (2*4*3*2*4*5) with 2 learning rate parameters, 4 units parameters, 3 loss function parameters, 2 dropout parameters, 4 batch size parameters, and 5 epoch parameters.

The hyperparameters of the models that achieved the best results during the testing phase are presented in Table 1.

Table 1. The Best Performing Hyperparameters in Chen Chaotic Time Series Prediction

<table>
<tr><th rowspan="7">learning_rate = [0.001, 0.01]
units= [32, 50, 64, 128]
loss_function= ['mse','mae','mape']
dropout=[0, 0.1]
batch_size=[16, 32, 64, 128]
epochs=[32, 50, 60, 100, 120]</th><th>Model</th><th>Learning rate</th><th>Unit</th><th>Loss function</th><th>Dropout</th><th>Batch size</th><th>Epoch</th></tr>
<tr><td>GRU-GRU</td><td>0.001</td><td>128</td><td>MSE</td><td>0</td><td>128</td><td>120</td></tr>
<tr><td>LSTM-LSTM</td><td>0.001</td><td>128</td><td>MSE</td><td>0</td><td>16</td><td>32</td></tr>
<tr><td>RNN-RNN</td><td>0.001</td><td>32</td><td>MSE</td><td>0</td><td>128</td><td>120</td></tr>
<tr><td>GRU</td><td>0.001</td><td>128</td><td>MSE</td><td>0</td><td>32</td><td>100</td></tr>
<tr><td>LSTM</td><td>0.001</td><td>128</td><td>MSE</td><td>0</td><td>16</td><td>100</td></tr>
<tr><td>RNN</td><td>0.001</td><td>32</td><td>MAE</td><td>0</td><td>32</td><td>120</td></tr>
</table>

The evaluation metric results for the models in Table 1 are presented in Table 2, revealing that the optimal performance is obtained with the single-layer LSTM model.

Table 2. Metric Values Obtained from the Prediction of the Chen Chaotic Time Series

Model	**MSE**	**RMSE**	**MAE**	**MAPE**	**R^2**
GRU-GRU	0.000189153989333943	0.0137533264824894	0.0097478297875705	0.0041291130350988	0.99999733972308
LSTM-LSTM	0.000357617461120523	0.0189107763225237	0.0130130124454234	0.004393590138745	0.99999497043925
RNN-RNN	0.002138219492340051	0.0462408855055788	0.0298703904947043	0.0097587028593110	0.99996992790902
GRU	0.000145790007131551	0.0120743532800540	0.0092421103157197	0.0035755420327708	0.99999794959761
LSTM	**0.000134953737116325**	**0.0116169590305004**	**0.0078688794263481**	**0.0031135044182560**	**0.99999810199978**
RNN	0.003563302760334845	0.0596934063388482	0.0511561370701601	0.0242245884906309	0.99994988542328

In the prediction study of the y_t state variable of the Lorenz chaotic system, each model was trained with 960 or 480 combinations of selected hyperparameters. The hyperparameters of the models that provided the best performance during the testing phase are indicated in Table 3.

Table 3. The Best Performing Hyperparameters in Lorenz Chaotic Time Series Prediction

	Model	**Learning rate**	**Unit**	**Loss function**	**Dropout**	**Batch size**	**Epoch**
learning_rate = [0.001, 0.01] units= [32, 50, 64, 128] loss_function= ['mse','mae','mape'] dropout=[0, 0.1] batch_size=[16, 32, 64, 128] epochs=[32, 50, 60, 100, 120]	GRU-GRU	0.001	64	MSE	0	64	120
	LSTM-LSTM	0.001	64	MSE	0	128	120
	RNN-RNN	0.001	64	MSE	0.1	128	120
	GRU	0.001	64	MSE	0	16	120
learning_rate = [0.001, 0.01] units= [32, 50, 64, 128] loss_function= ['mse','mae','mape'] **dropout=0** batch_size=[16, 32, 64, 128] epochs=[32, 50, 60, 100, 120]	LSTM	0.001	50	MSE	0	32	60
	RNN	0.001	50	MAE	0	32	120

The evaluation metric results for the models in Table 3 are presented in Table 4, revealing that the best performance is obtained with the two-layer LSTM-LSTM model.

Table 4. Metric Values Obtained from the Prediction of the Lorenz Chaotic Time Series

Model	**MSE**	**RMSE**	**MAE**	**MAPE**	**R^2**
GRU-GRU	0.00156822737135888	0.0396008506393345	0.0292398252888286	0.012008021793599652	0.9999680544908985
LSTM-LSTM	**0.00100249889715198**	**0.0316622629821682**	**0.0207719784188649**	**0.016943825317170078**	**0.9999795786387689**
RNN-RNN	0.00492290097125468	0.0701633876837107	0.0535579514135681	0.0367872331437873	0.99989971825473
GRU	0.00132824040976935	0.03644503271735877	0.02410611054552382	0.0172672800657041	0.999972943135113
LSTM	0.00158062587186254	0.03975708580696701	0.0281864592738778	0.0190784157125894	0.999967801927770
RNN	0.00594852642734398	0.0771266907584137	0.0465172340879340	0.0389858977521359	0.99987882579491

In the prediction of Lorenz and Chen's chaotic time series, good results have generally been obtained during both training and testing phases with learning rate = 0.001 and dropout = 0 values. Considering this, models for predicting the Rikitake chaotic time series were trained and tested with 960, 480, or 240 hyperparameter combinations. In two-layer models, 12,800 data points were used for training, and 3,200 data points were used for testing out of a total of 16,000 data points. For single-layer models, 8,000 data points were used for training and 2,000 data points for testing out of a total of 10,000 data points. The hyperparameters of the models that yielded the best results in predicting the x_t variable of the Rikitake system are provided in Table 5.

Table 5. The Best Performing Hyperparameters in Rikitake Chaotic Time Series Prediction

	Model	**Learning rate**	**Unit**	**Loss function**	**Dropout**	**Batch size**	**Epoch**
learning_rate = [0.001, 0.01] units= [32, 50, 64, 128] loss_function= ['mse','mae','mape'] dropout=[0, 0.1] batch_size=[16, 32, 64, 128] epochs=[32, 50, 60, 100, 120]	GRU-GRU	0.001	64	MSE	0	128	32
learning_rate =0.001 units= [32, 50, 64, 128] loss_function= ['mse','mae','mape'] **dropout=0** batch_size=[16, 32, 64, 128] epochs=[32, 50, 60, 100, 120]	LSTM-LSTM	0.001	32	MSE	0	128	100
	RNN-RNN	0.001	32	MSE	0	128	32
learning_rate = 0.001 units= [32, 50, 64, 128] loss_function= ['mse','mae','mape'] dropout=[0, 0.1] batch_size=[16, 32, 64, 128] epochs=[32, 50, 60, 100, 120]	GRU	0.001	32	MSE	0.1	128	120
	LSTM	0.001	50	MAPE	0	128	120
	RNN	0.001	64	MSE	0.1	128	100

The evaluation metric results for the models in Table 5 are presented in Table 6, revealing that the best performance is obtained with the two-layer GRU-GRU model.

Table 6. Metric Values Obtained from the Prediction of the Rikitake Chaotic Time Series

Model	**MSE**	**RMSE**	**MAE**	**MAPE**	**R^2**
GRU-GRU	**0.000103056930158843**	**0.0101516959252552**	**0.0068777473052769**	**0.0696440854960179**	**0.999964715670350**
LSTM-LSTM	0.000160392630502165	0.0126646212143185	0.0086265765293640	0.0435886229582805	0.999944952715469
RNN-RNN	0.000707852209306246	0.0266054920891579	0.0239891412013945	0.2997457660216611	0.999757647635554
GRU	0.00011047195962303	0.0105105641914709	0.0067783695547486	0.0399761636985075	0.999955300069389
LSTM	0.000117093448747761	0.0108209726340917	0.0065403506694696	0.0111550839614390	0.999952620836528
RNN	0.000150258160515648	0.01225798354198795	0.00868696034951605	0.0397396786910791	0.999939201500799

The actual values of the x_t state variable of the multivariate Chen chaotic system and the predicted values from the LSTM model are shown in Figure 8. The actual values of the y_t state variable of the multivariate Lorenz chaotic system and the predicted values from the LSTM-LSTM model are shown in Figure 9. The actual values of the x_t state variable of the multivariate Rikitake chaotic system and the predicted values from the GRU-GRU model are shown in Figure 10.

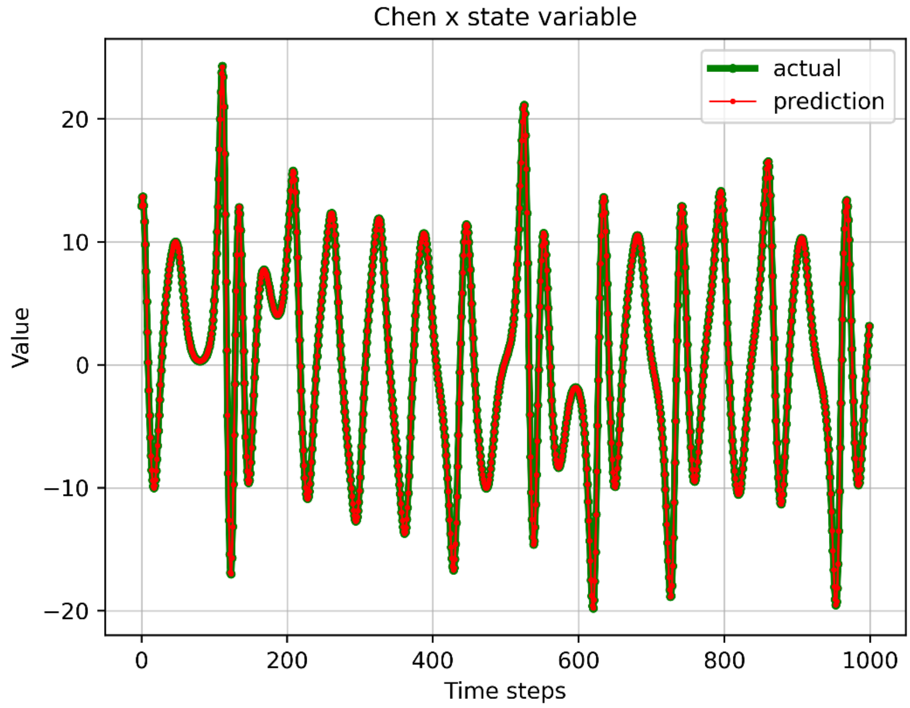

Figure 8. Prediction Result of the $\boldsymbol{x_t}$ State Variable in the Multivariate Chen Chaotic System

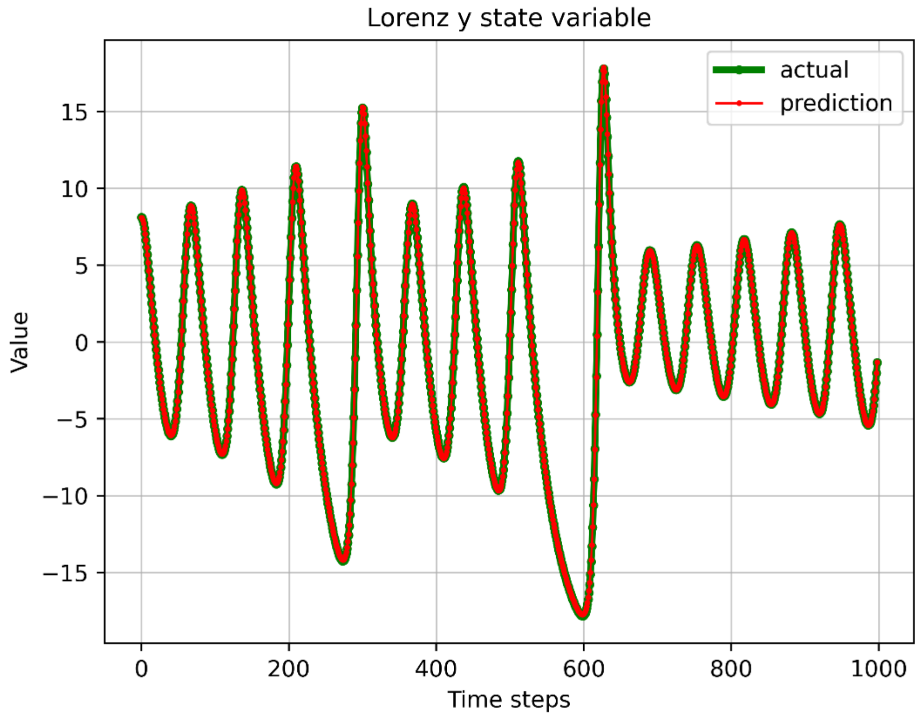

Figure 9. Prediction Result of the $\boldsymbol{y_t}$ State Variable in the Multivariate Lorenz Chaotic System

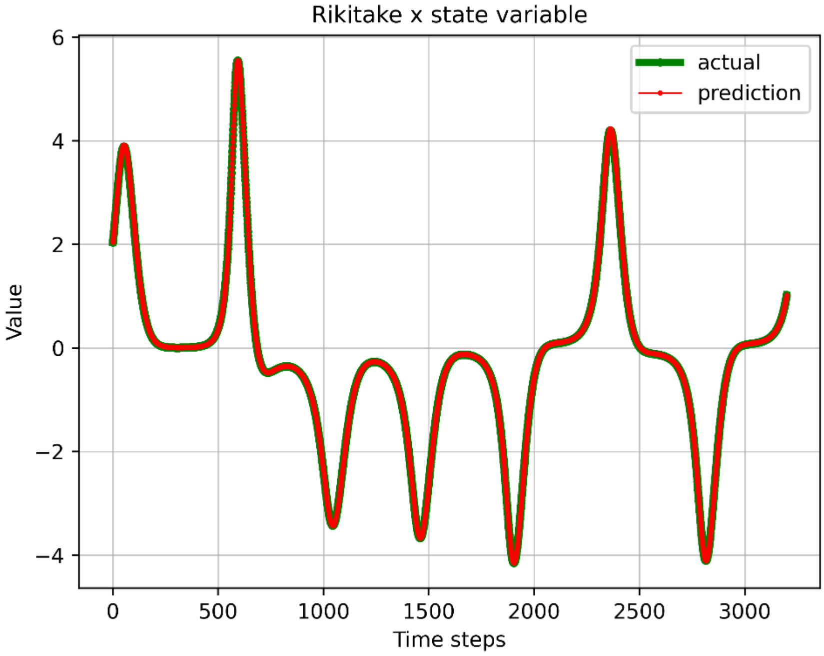

Figure 10. Prediction Result of the x_t State Variable in the Multivariate Rikitake Chaotic System

For each chaotic system, training and testing were conducted on six models using the hyperparameters specified in the above tables. The metric values for the top three combinations yielding the best results in the hyperparameter combination of the model that achieved the best performance for each chaotic system are presented in Figures 11, 12, 13, 14, and 15. For instance, the Lorenz chaotic system was trained and tested with 960 hyperparameter combinations in the GRU model. Among the results of these 960 studies, the three most successful studies are denoted as Lorenz_GRU_1, Lorenz_GRU_2, and Lorenz_GRU_3, respectively.

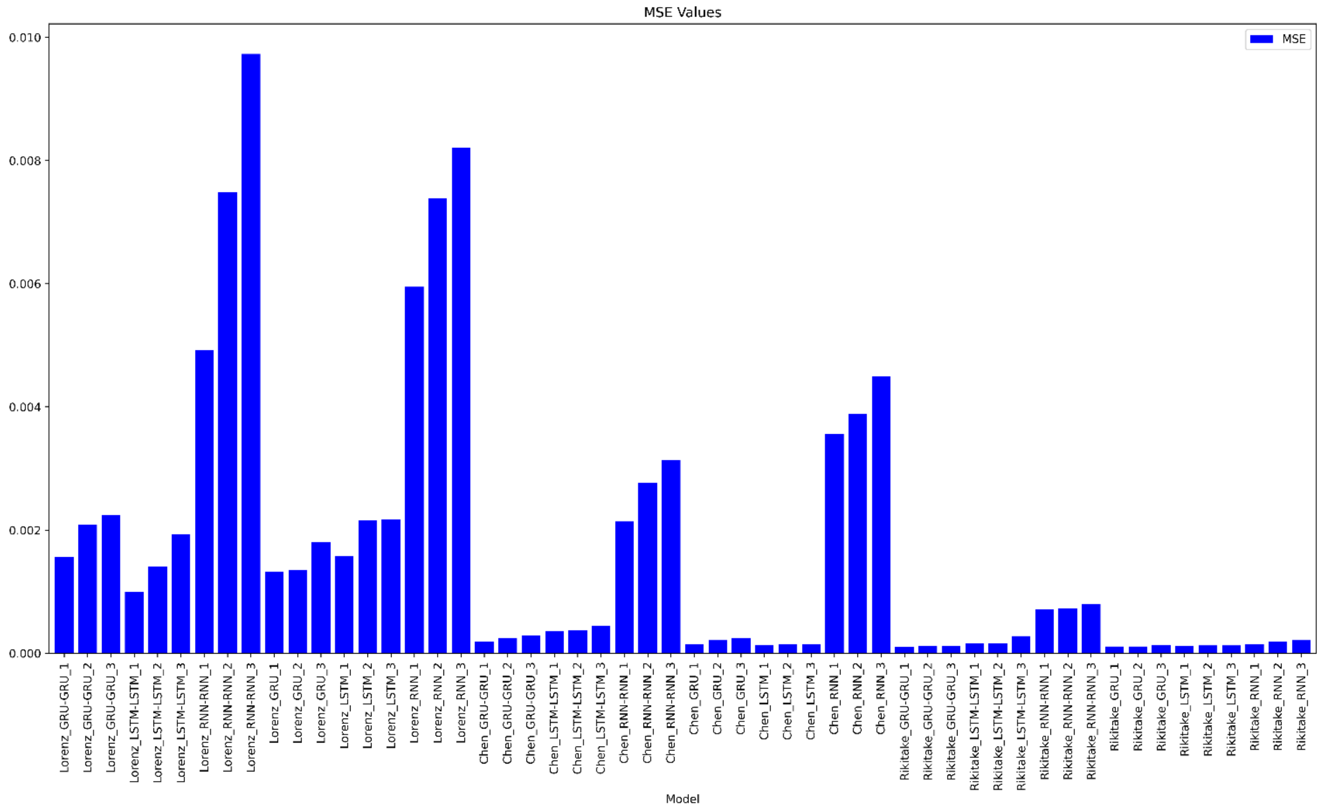

Figure 11. MSE Values of the Models That Achieved the Best Performance in Chaotic Systems

In Figure 11, it appears that models achieved lower MSE values in predicting Chen and Rikitake's chaotic time series. Additionally, RNN models obtained higher MSE values in the prediction of chaotic time series.

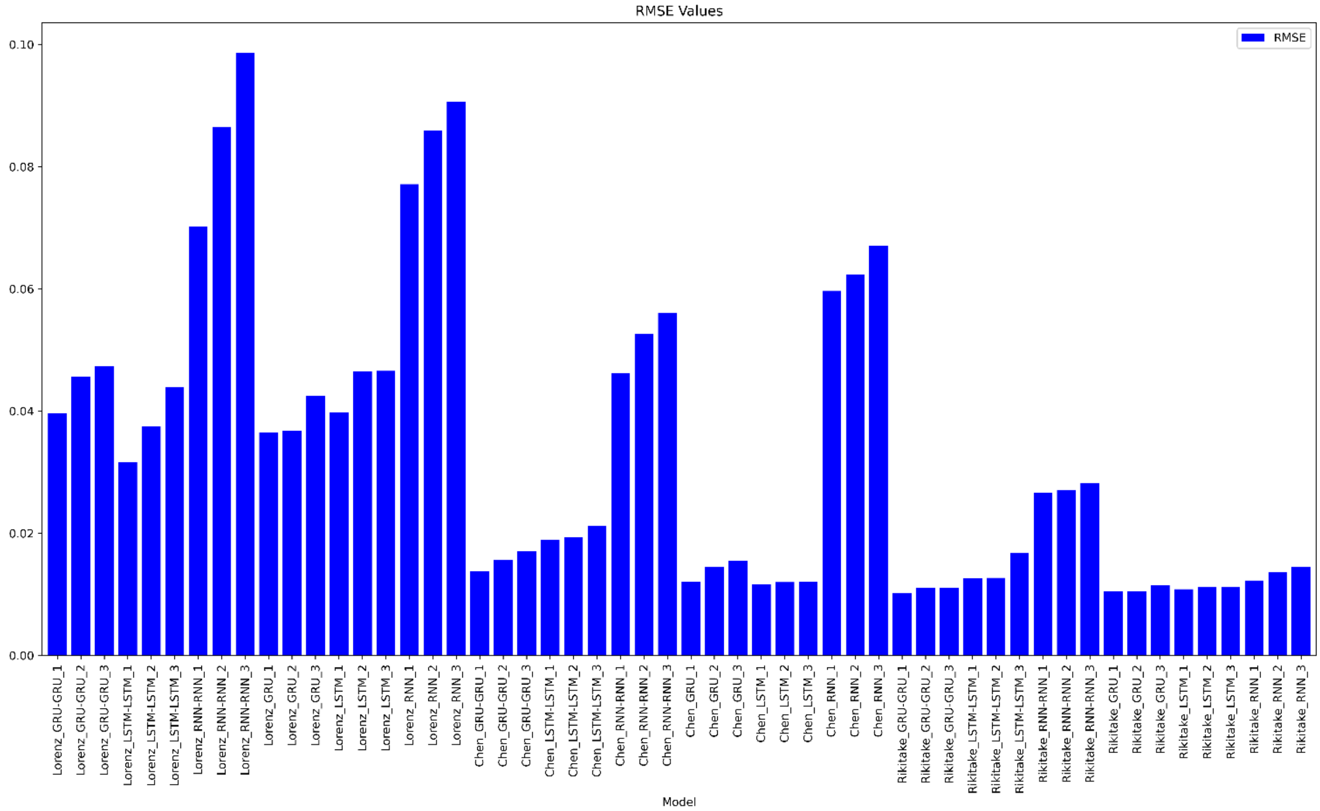

Figure 12. RMSE Values of the Models That Achieved the Best Performance in Chaotic Systems

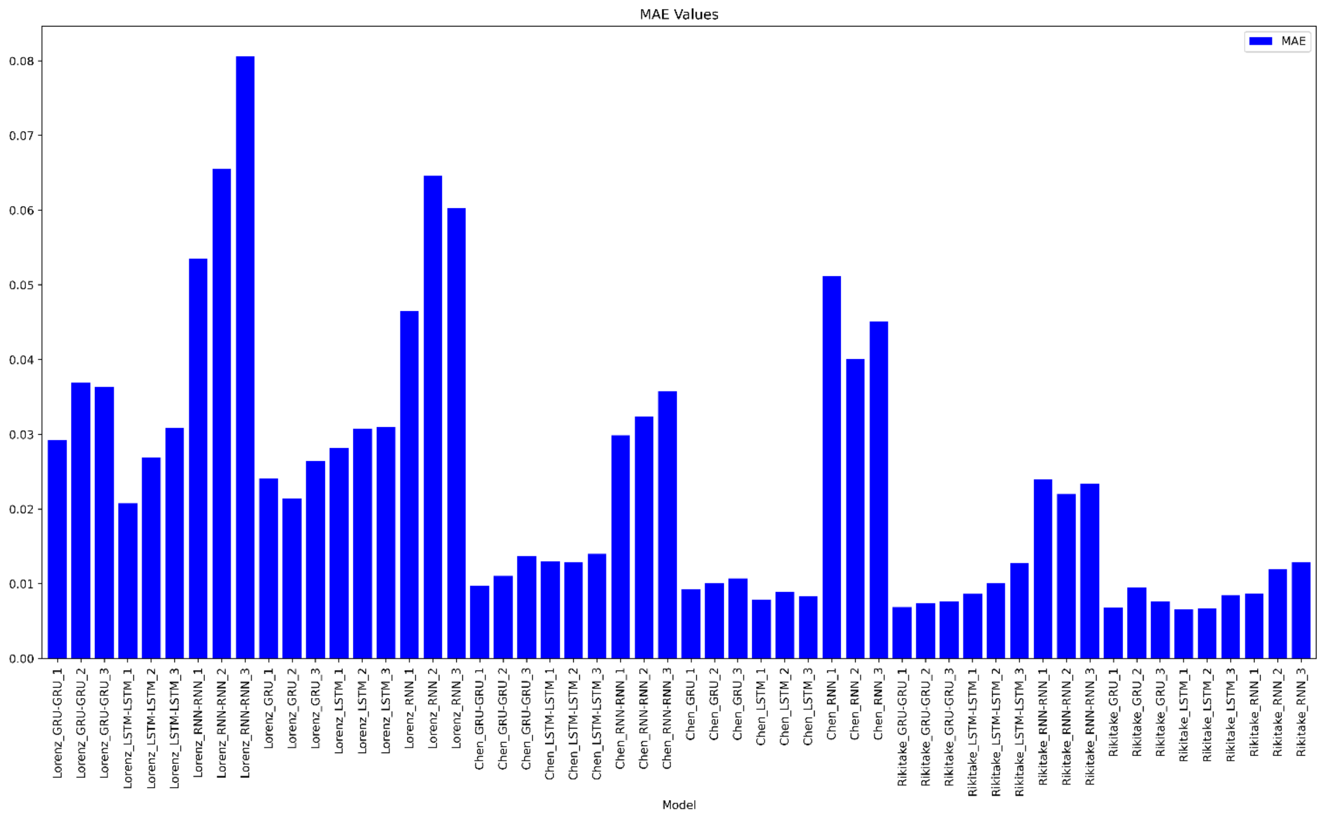

Figure 13. MAE Values of the Models That Achieved the Best Performance in Chaotic Systems

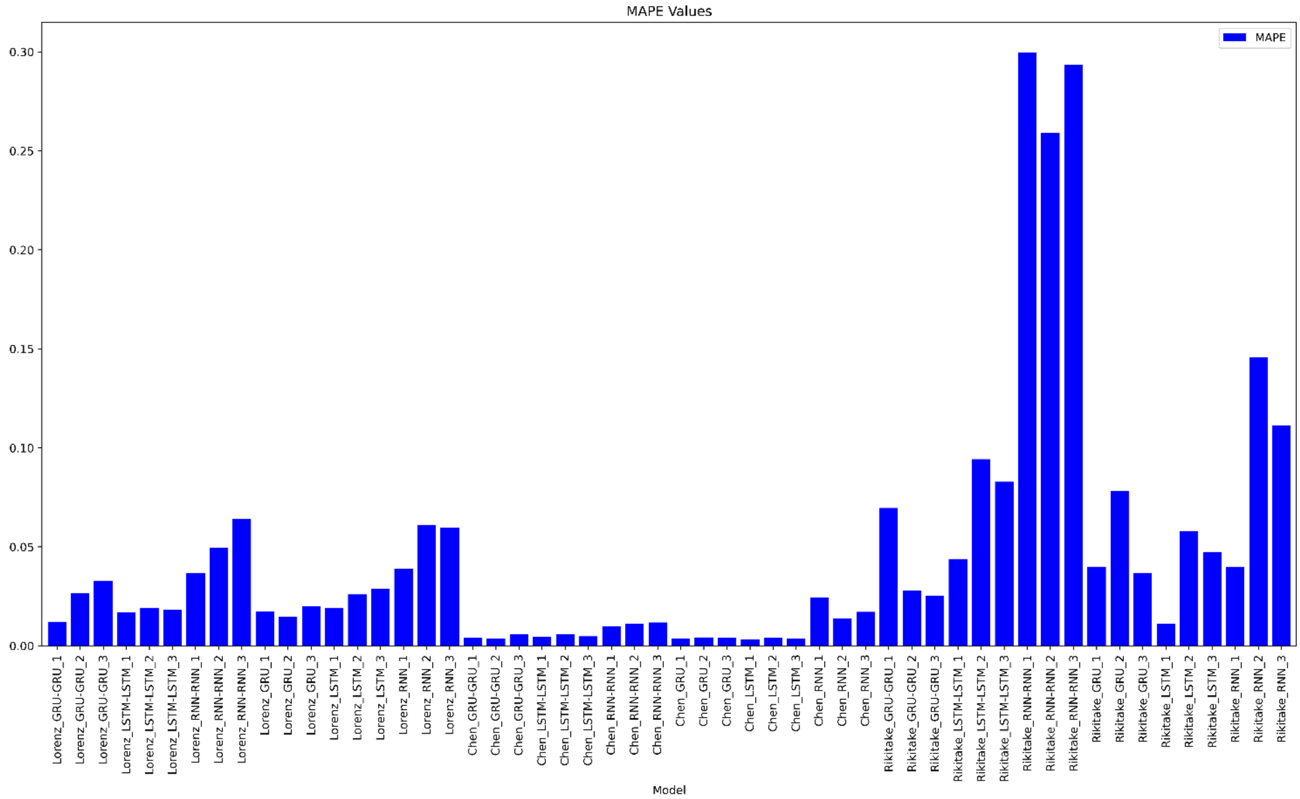

Figure 14. MAPE Values of the Models That Achieved the Best Performance in Chaotic Systems

Upon examining Figures 13 and 14, it is evident that in the prediction results of Lorenz and Chen's chaotic time series, MAPE values are lower than MAE values. This suggests that the model incurs substantial errors concerning the true values; however, when these errors are expressed relative to the entire dataset, they appear relatively smaller. The low MAE values in the prediction of the Rikitake chaotic time series indicate that the predictions are close to the real values, and small errors are made. The fact that the MAPE values are higher than the MAE values indicates that the small errors that occur have a higher effect when compared to the entire data. This indicates the presence of proportional errors in predicting the Rikitake chaotic system. This situation does not imply poor performance in the model's prediction study; rather, it indicates that the data generated from the Rikitake system has a narrower value range than the other two models.

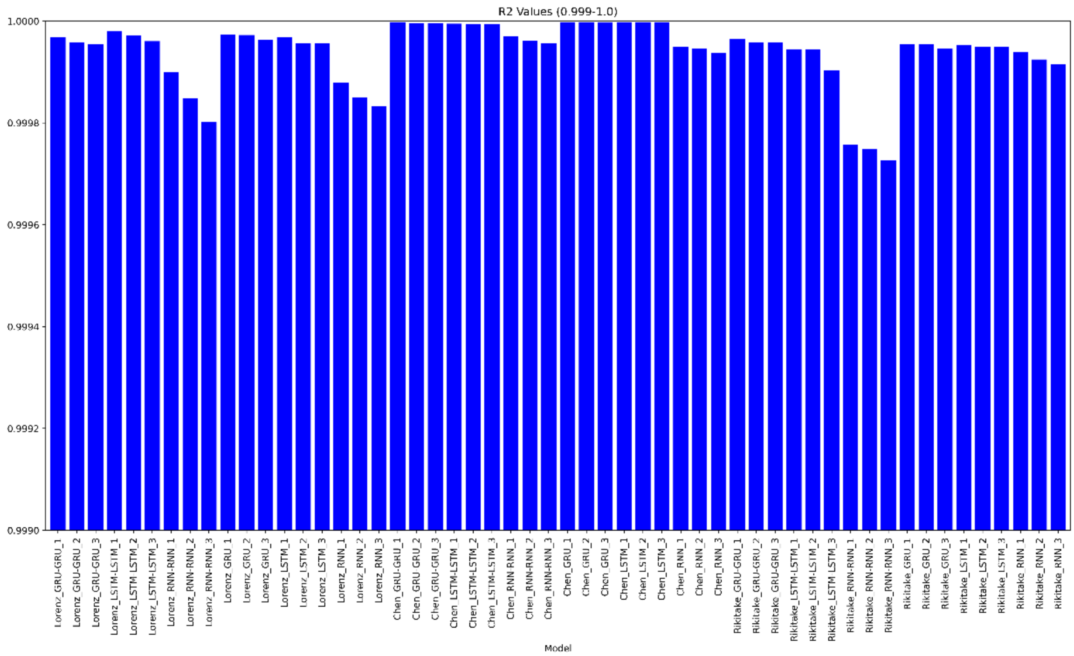

Figure 15. R^2 Values of the Models That Achieved the Best Performance in Chaotic Systems

Table 7 outlines the studies in the literature that utilize the Lorenz, Chen, and Rikitake chaotic systems. It specifies the number of data points generated from these systems, the split ratios of data points used in the models, employed prediction models, and the evaluation metrics used to assess the prediction results. All studies mentioned in Table 7 have attempted to predict only one state variable of the chaotic system by giving one state variable to the prediction models. The studies numbered 20, 22, and 23 created their own hybrid models to achieve good performance in prediction tasks. They demonstrated that their hybrid models yielded more accurate results than other models such as LSTM, GRU, TCN, and CNN. However, in our study, better prediction results were obtained without using any hybrid models. This was achieved by providing the state variables x, y, and z of the system to models like RNN, LSTM, and GRU for predicting a single state variable. The number of data points and the split ratios of the data points also influence the performance of the prediction models. Despite using the same data point split ratio and having a larger number of data points, study number 23 obtained less favorable results compared to our study, as indicated by their evaluation metrics. In Table 7, only study number 9 achieved better prediction results than our proposed study. The researchers utilized 100,000 data points instead of 5,000, split the data into training, testing, and validation sets at different ratios, and used Grid Search for hyperparameter selection to achieve better performance.

Table 7. Models Used in Chaotic Time Series Predictions and the Obtained Metric Values

Reference	Chaotic System	Datasets	Model	Evaluation Metric
Dudukcu et al. [9]	Lorenz $x_0 = 0.9$ $y_0 = 0.9$ $z_0 = 0$ Input: y_t Output: y_t	100,000 data point 40% train 10% validation 50% test	LSTM	RMSE: 0.0042, MAE: 0.0025, R^2: 0.9994
			GRU	RMSE: 0.0045, MAE: 0.0028, R^2: 0.9993
			LSTM-LSTM	RMSE: 0.0041, MAE:0.0025, R^2: 0.9994
			GRU-GRU	RMSE: 0.0038, MAE: 0.0023, R^2: 0.9995
			LSTM-GRU	RMSE: 0.0044, MAE: 0.0027, R^2: 0.9993
			TCN-LSTM	RMSE: 0.0024, MAE: 0.0014, R^2: 0.9998
			TCN-GRU	RMSE: 0.0029, MAE: 0.0017, R^2: 0.9997
Yanan et al. [20]	Lorenz-63 $x_0 = -0.2028$ $y_0 = 3.5418$ $z_0 = 25.0873$	5,000 data point	CEEMDAN-LSTM	RMSE: 1.327, MAE: 1.124, MAPE: 0.119
			LSTM	RMSE: 2.042, MAE: 1.527, MAPE: 0.214
Dudukcu et al. [21]	Lorenz $x_0 = 0.9$ $y_0 = 0.9$ $z_0 = 0$ Input: y_t Output: y_t	150,000 data point 40% train 10% validation 50% test	Vanilla LSTM	RMSE:0.3376
			Stacked LSTM	RMSE:0.3213
			Bidirectional LSTM	RMSE:0.3005
			CNN-LSTM	RMSE:0.3311
Fu et al. [22]	Lorenz $x_0 = 1$ $y_0 = 0$ $z_0 = 1$ Input: x_t Output: x_t	8,000 data point 70% train 5% validation 25% test	DTIGNet GRU LSTM CNN-GRU CNN-LSTM	MAE: 0.022654, RMSE:0.030894 MAPE: 0.02886535, R^2: 0.999983
				MAE: 0.049698, RMSE:0.070488 MAPE: 0.01689394, R^2: 0.999911
				MAE: 0.141781, RMSE:0.212693 MAPE: 0.18966734, R^2: 0.999822
				MAE: 0.046484, RMSE: 0.065715 MAPE: 0.22064176, R^2: 0.999922
				MAE: 0.076838, RMSE: 0.108909 MAPE: 0.23432454, R^2: 0.999787
Cheng et al. [23]	Chen $x_0 = -1$ $y_0 = -1.1$ $z_0 = 0$ Input: x_t Output: x_t	8,000 data point 80% train 20% test	TCN-CBAM TCN CNN-LSTM LSTM	MAE: 0.15410, RMSE: 0.21076, R^2: 0.99938
				MAE:0.26315, RMSE: 0.33381, R^2: 0.99843
				MAE: 0.2706, RMSE: 0.48527, R2: 0.99671
				MAE: 0.42837, RMSE: 0.85431, R^2: 0.98984
	Lorenz $x_0 = 1$ $y_0 = 0$ $z_0 = 1$ Input: x_t Output: x_t	10,000 80% train 20% test	TCN-CBAM TCN CNN-LSTM LSTM	MAE: 0.09998, RMSE: 0.14039, R^2: 0.99969
				MAE: 0.14883, RMSE: 0.18882, R^2: 0.99943
				MAE: 0.12303, RMSE: 0.18913, R^2: 0.99943
				MAE: 0.13647, RMSE: 0.22891, R^2: 0.99917
Our study	Lorenz $x_0 = 0$ $y_0 = -0.1$	5,000 data point	LSTM	MSE: 0.000135 RMSE: 0.01162 MAE: 0.00787

	$z_0 = 9$ Input: x_t , y_t , z_t Output: y_t	80% train 20% test		MAPE: 0.00311 R^2: 0.999998
	Chen $x_0 = -10$ $y_0 = 0$ $z_0 = 37$ Input: x_t , y_t , z_t Output: x_t	5,000 data point 80% train 20% test	LSTM-LSTM	MSE: 0.001002 RMSE: 0.03166 MAE: 0.020772 MAPE: 0.01694 R^2: 0.99998
	Rikitake $x_0 = 3$ $y_0 = 1$ $z_0 = 6$ Input: x_t , y_t , z_t Output: x_t	16,000 data point 80% train 20% test	GRU-GRU	MSE: 0.0001031 RMSE: 0.010152 MAE: 0.006878 MAPE: 0.06964 R^2: 0.999965

4. Conclusion and Future Work

This study assesses the effectiveness of deep learning models in predicting multivariate chaotic time series by focusing on the multivariate Lorenz, Lorenz-like Chen, and non-Lorenz-like Rikitake chaotic systems. The chaotic time series generated from chaotic systems using the fourth order Runge-Kutta method were employed in one and two-layer GRU, LSTM, and RNN models. The prediction performances of deep learning models, trained and tested with different hyperparameter combinations, were compared using evaluation metrics such as MSE, RMSE, MAE, MAPE, and R^2. The experimental study demonstrated that using all state variables composing the chaotic systems, rather than a single state variable, in the prediction models resulted in better performance than similar studies. This approach suggests that a model predicting multivariate chaotic time series can better understand the system dynamics comprehensively, leading to more reliable predictions. In addition, compared to studies in the literature that use hybrid model design to achieve high performance, this study shows that prediction performance is improved with basic LSTM, GRU, and RNN models and appropriate hyperparameters selected for these models, without using hybrid model design.

In future studies, time series predictions can be conducted on more complex chaotic systems and real-world problems using new hyperparameters and models. Additionally, research can be carried out to predict not only the next step but also several steps ahead in time series prediction studies. This would allow for a comprehensive comparison of the most suitable models for such studies.

References

[1] A. L. Mrgole and D. Sever, “Incorporation of Duffing Oscillator and Wigner-Ville Distribution in Traffic Flow Prediction,” *Promet - Traffic&Transportation*, vol. 29, no. 1, pp. 13–22, Feb. 2017, doi: 10.7307/ptt.v29i1.2116.

[2] J. Runge and R. Zmeureanu, “A Review of Deep Learning Techniques for Forecasting Energy Use in Buildings,” *Energies*, vol. 14, no. 3, Art. no. 3, Jan. 2021, doi: 10.3390/en14030608.

[3] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, “Financial Time Series Forecasting with Deep Learning : A Systematic Literature Review: 2005-2019.” arXiv, Nov. 29, 2019. doi: 10.48550/arXiv.1911.13288.

[4] I. Yazici, O. F. Beyca, and D. Delen, “Deep-learning-based short-term electricity load forecasting: A real case application,” *Engineering Applications of Artificial Intelligence*, vol. 109, p. 104645, Mar. 2022, doi: 10.1016/j.engappai.2021.104645.

[5] M. Murat, I. Malinowska, M. Gos, and J. Krzyszczak, “Forecasting daily meteorological time series using ARIMA and regression models,” *International Agrophysics*, vol. 32, no. 2, pp. 253–264, Apr. 2018, doi: 10.1515/intag-2017-0007.

[6] P. Kavianpour, M. Kavianpour, E. Jahani, and A. Ramezani, “A CNN-BiLSTM model with attention mechanism for earthquake prediction,” *J Supercomput*, May 2023, doi: 10.1007/s11227-023-05369-y.

[7] T. Ouyang, H. Huang, Y. He, and Z. Tang, “Chaotic wind power time series prediction via switching data-driven modes,” *Renewable Energy*, vol. 145, pp. 270–281, Jan. 2020, doi: 10.1016/j.renene.2019.06.047.

[8] C. Cheng et al., “Time series forecasting for nonlinear and non-stationary processes: a review and comparative study.” *IIE Transactions*, vol. 47, no. 10, pp. 1053–1071, Oct. 2015, doi: 10.1080/0740817X.2014.999180.

[9] H. V. Dudukcu, M. Taskiran, Z. G. C. Taskiran, and T. Yildirim, “Temporal Convolutional Networks with RNN approach for chaotic time series prediction,” *Applied Soft Computing*, vol. 133, p. 109945, Jan. 2023, doi: 10.1016/j.asoc.2022.109945.

[10] D. S. K. Karunasinghe and S.-Y. Liong, “Chaotic time series prediction with a global model: Artificial neural network,” *Journal of Hydrology*, vol. 323, no. 1, pp. 92–105, May 2006, doi: 10.1016/j.jhydrol.2005.07.048.

[11] H. Yuxia and Z. Hongtao, “Chaos Optimization Method of SVM Parameters Selection for Chaotic Time Series Forecasting,” *Physics Procedia*, vol. 25, pp. 588–594, Jan. 2012, doi: 10.1016/j.phpro.2012.03.130.

[12] Y. Xiu and W. Zhang, "Multivariate Chaotic Time Series Prediction Based on NARX Neural Networks," *in 2017 2nd International Conference on Electrical, Automation and Mechanical Engineering (EAME 2017)*, vol. 86. Paris: Atlantis Press, 2017, pp. 164–167.

[13] S. Siami-Namini and A. S. Namin, "Forecasting Economics and Financial Time Series: ARIMA vs. LSTM." arXiv, Mar. 16, 2018. doi: 10.48550/arXiv.1803.06386.

[14] R. Khaldi, A. E. Afia, R. Chiheb, and S. Tabik, "What is the best RNN-cell structure to forecast each time series behavior?," *Expert Systems with Applications*, vol. 215, p. 119140, Apr. 2023, doi: 10.1016/j.eswa.2022.119140.

[15] G. Alkhayat and R. Mehmood, "A review and taxonomy of wind and solar energy forecasting methods based on deep learning," *Energy and AI*, vol. 4, p. 100060, Jun. 2021, doi: 10.1016/j.egyai.2021.100060.

[16] H. Liu, G. Yan, Z. Duan, and C. Chen, "Intelligent modeling strategies for forecasting air quality time series: A review," *Applied Soft Computing*, vol. 102, p. 106957, Apr. 2021, doi: 10.1016/j.asoc.2020.106957.

[17] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, Apr. 1990, doi: 10.1016/0364-0213(90)90002-E.

[18] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation." arXiv, Sep. 02, 2014. doi: 10.48550/arXiv.1406.1078.

[19] R. Chandra and M. Zhang, "Cooperative coevolution of Elman recurrent neural networks for chaotic time series prediction," *Neurocomputing*, vol. 86, pp. 116–123, Jun. 2012, doi: 10.1016/j.neucom.2012.01.014.

[20] G. Yanan, C. Xiaoqun, L. Bainian, and P. Kecheng, "Chaotic Time Series Prediction Using LSTM with CEEMDAN." *Journal of Physics: Conferences Series*, vol. 1617, no. 1, p. 012094, Aug. 2020, doi: 10.1088/1742-6596/1617/1/012094.

[21] H. V. Dudukcu, M. Taskiran, and Z. G. C. Taskiran, "Comprehensive Comparison of LSTM Variations for the Prediction of Chaotic Time Series," *in 2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, Aug. 2021, pp. 1–5. doi: 10.1109/INISTA52262.2021.9548647.

[22] K. Fu, H. Li, and P. Deng, "Chaotic time series prediction using DTIGNet based on improved temporal-inception and GRU," *Chaos, Solitons & Fractals*, vol. 159, p. 112183, Jun. 2022, doi: 10.1016/j.chaos.2022.112183.

[23] W. Cheng et al., "High-efficiency chaotic time series prediction based on time convolution neural network," *Chaos, Solitons & Fractals*, vol. 152, p. 111304, Nov. 2021, doi: 10.1016/j.chaos.2021.111304.

[24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[25] E. N. Lorenz, "Deterministic Nonperiodic Flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, Mar. 1963, doi: 10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2.

[26] G. Chen and T. Ueta, "Yet another chaotic attractor." *International Journal of Bifurcation and Chaos*, vol. 09, no. 07, pp. 1465–1466, Jul. 1999, doi: 10.1142/S0218127499001024.

[27] K. Ito, "Chaos in the Rikitake two-disc dynamo system," *Earth and Planetary Science Letters*, vol. 51, no. 2, pp. 451–456, Dec. 1980, doi: 10.1016/0012-821X(80)90224-1.

[28] T. Rikitake, "Oscillations of a system of disk dynamos," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 54, no. 1, pp. 89–105, Jan. 1958, doi: 10.1017/S0305004100033223.

[29] L. Wang and L. Dai, "Chaotic Time Series Prediction of Multi-Dimensional Nonlinear System Based on Bidirectional LSTM Model," *Advanced Theory and Simulations*, vol. 6, no. 8, p. 2300148, 2023, doi: 10.1002/adts.202300148.

Author(s) Contributions
Gülyeter Öztürk: Data generation, performing analysis, writing, review, and editing.
Osman Eldoğan: Writing, review, and editing.

Conflict of Interest Notice
Authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval
It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Availability of data and material
Not applicable

Plagiarism Statement
This article has been scanned by iThenticate ™.