IFERP
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 10, Issue 11, November 2023

Comparative Analysis of Traditional and Deep Learning Methods for Time Series Prediction in Dynamical Systems

Samaneh Sanatifar
Islamic Azad University, North Tehran Branch, Department of Statistics and Mathematics, Tehran, Iran
sanatifarsamaneh1@gmail.com

Abstract— ***Time series prediction, particularly for nonlinear dynamical systems, has long been a topic of interest in applied mathematics. Traditional algorithms, encompassing methods such as Linear Regression, Polynomial Regression, and Random Forest, have been utilized for their simplicity and ease of interpretation. While effective in certain scenarios, these traditional models sometimes fall short of capturing the nuanced complexities and longer-term intricacies of dynamic systems. With the recent advancements in artificial intelligence, deep learning methods, notably Gated Recurrent Units (GRU) and Long Short-Term Memory Networks (LSTM), offer promising avenues for enhancing predictive performance. In this study, we evaluate the capabilities of traditional prediction models against the prowess of deep learning techniques using synthetic datasets derived from the renowned Lorenz system. The aim is to determine the efficacy of each model in accurately forecasting time series data, bearing the intricacies of the Lorenz system in mind. Our results shed light on a notable hierarchy in prediction performance. The GRU architecture emerged as the front-runner with an RMSE of 0.21, demonstrating its superior ability to learn and predict the intricate dynamics of the Lorenz system. This was closely followed by the LSTM, yielding an RMSE of 0.29. Among traditional methods, Random Forest and Linear Regression showed comparable performances with RMSEs of 0.48 and 0.51, respectively, while Polynomial Regression trailed with an RMSE of 0.69. This comparative analysis shows that while traditional models hold their ground, deep learning methods, particularly GRU, offer enhanced predictive capabilities for complex dynamical systems. This research underscores the potential of integrating deep learning into time series forecasting and highlights the necessity of choosing the right model based on the intricacy and nature of the data in question.***

Index Terms— ***Time series prediction, Nonlinear dynamical systems, Linear Regression, Polynomial Regression, Random Forest, Deep Learning Algorithms, Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM), Lorenz system***

I. INTRODUCTION

The Dynamical systems, particularly those of a nonlinear nature, have always commanded a substantial interest from scholars and industry experts alike due to their intricate behaviors and substantial relevance in physical, biological, and social systems [1]–[3]. The intricacy and unpredictability of nonlinear dynamical systems (NDS) create a compelling yet challenging domain for time series prediction [4]–[6]. Predicting the evolution of such systems' evolution holds theoretical interest and practical significance across diverse fields, including meteorology, finance, and engineering [7], [8]. Time series prediction within dynamical systems, particularly nonlinear ones, poses unique challenges. The intricate temporal dependencies and potential for chaotic behaviors in NDS require predictive models that can navigate through these complex dynamics [9]–[11].

The Lorenz system, introduced by Edward N. Lorenz in 1963 [12], is emblematic in this domain, offering a canonical example of chaos in a set of nonlinear differential equations and, thus, serving as a model problem for predictive algorithms in NDS [13]. The Lorenz system provides a fruitful playground for exploring novel methodologies and tactics for enhancing time series predictive accuracy by handling the potential divergences in trajectories that emanate from minuscule alterations in initial conditions [14], [15].

Traditional prediction models such as Linear Regression [16], Polynomial Regression [17], and Random Forest [18] have demonstrated aptitude in time series forecasting by leveraging historical data to forecast future points. While these models are praised for their simplicity and interpretability, their performance could be constrained by the inherent linearity and inability to capture the long-term dependencies and nonlinear relationships embedded in the data from NDS [19].

In contrast, the advent of deep learning algorithms, specifically the Gated Recurrent Units (GRU) [20], [21] and Long Short-Term Memory Networks (LSTM) [22], has reshaped the landscape of time series prediction. These methods, known for their capability to model temporal dependencies and handle nonlinear data relationships, have shown remarkable results in various applications involving NDS predictions, albeit with their own set of challenges and complexities [23].

Time series prediction in nonlinear dynamical systems encapsulates numerous challenges, such as susceptibility to initial conditions, model overfitting, and computational costs [24], [25]. Addressing these challenges warrants a strategic integration of model sophistication and empirical data understanding. Strategies such as feature engineering, regularization, and model ensemble methods have been explored to mitigate these challenges, and their integration into the predictive models demands a meticulous and data-conscious approach [26]–[28].

This study embarks on a journey to scrutinize and compare the predictive prowess of traditional algorithms against deep learning methodologies using data generated from the Lorenz system. The intention is to ascertain the effectiveness of each model in deciphering and predicting the dynamic intricacies inherent in a chaotic, nonlinear dynamical system. Through rigorous quantitative and qualitative analyses, this paper aims to unravel the strengths and limitations of each predictive model, providing insights that could guide future research and applications in time series prediction of nonlinear dynamical systems.

The remainder of this paper is structured as follows: Section 2 provides the material and methodology adopted in this study. Section 3 discloses and discusses the study's findings as results and Section 4 concludes the paper, offering insights into potential future work.

II. MATERIAL AND METHODOLOGY

A. Data Sets

The Lorenz system, renowned for its deterministic chaos, is described by three differential equations [12]:

$$\dot{X} = \sigma(y - x)$$
$$\dot{Y} = x(\rho - z) - y$$
$$\dot{Z} = xy - \beta z \quad (1)$$

Where *X*, *Y*, and *Z* define the system state, σ, ρ, and β are system parameters, and the dot notation signifies the derivative concerning time [13]. The parameters σ=10, ρ=28, and β=8/3 are employed, as they are known to induce chaotic behavior in the system [29]. Due to its non-linearity and three-dimensional phase space, the Lorenz system gives rise to intricate, butterfly-shaped attractors that form the basis of the generated time series data. The trajectories in the phase space remain bounded but never settle down to a steady state, exhibiting perpetual oscillatory motion [30].

The Lorenz system is notable for generating time series data characterized by chaotic dynamics, which are both deterministic and non-periodic. The trajectories within its phase space create a seemingly random yet deterministic pattern known as the Lorenz attractor [31]. It is precisely these chaotic characteristics — sensitivity to initial conditions and nonlinear dynamics — that make the Lorenz system a compelling subject for studying time series prediction.

When generating data from the Lorenz system, we set initial conditions and integrate the Lorenz equations over time, using numerical methods, such as the fourth-order Runge-Kutta method, to solve the differential equations iteratively [32]. The resultant time series — sequences of values for *x*, *y*, and *z* at discrete time points — provide a rich dataset embodying the nonlinear, chaotic nature of the system. This dataset, therefore, retains the inherent temporal dependencies and nonlinear trajectories that challenge predictive modeling. Temporal dependency means that a particular state or system output at a given time point depends on its preceding states [33]. The chaotic nature of the Lorenz system ensures that even small perturbations in initial conditions can yield widely divergent paths, making the prediction non-trivial and exploring predictive modeling strategies particularly vital. Thus, capturing and preserving these temporal dependencies is crucial for any model attempting to predict future states of the system.

Given the chaotic and unbounded nature of the Lorenz attractor, the raw data is subjected to normalization to ensure that the scale of the values does not unduly influence the training of predictive models. Normalization typically involves scaling all numerical values to a standard range, such as [0, 1] or [-1, 1], without distorting the differences in ranges of values [34]. This not only assists in maintaining numerical stability but also ensures that the models can be trained more efficiently.

$$Normalized\_Value = \frac{Value - Min\_Value}{Max\_Value - Min\_Value} \quad (2)$$

Where *Value* is the original data point, and *Min_Value* and *Max_Value* are the minimum and maximum values in the original data, respectively.

B. Methods

Predictive models are developed, trained, and validated using the processed Lorenz system data, aiming to estimate the subsequent states of the system with minimized error.

1) Traditional Predictive Models

The traditional predictive models, which include Linear Regression, Polynomial Regression, and Random Forest (in this study), are implemented due to their established utility in time series forecasting.

a) Linear Regression

Linear Regression is one of the simplest and most widely used statistical methods in predictive modeling and machine learning. It is utilized to model and analyze the relationships between a dependent variable and one or more independent variables. The main goal of linear Regression is to find the best fit straight line that accurately predicts the output values within a range [35].

Mathematically, linear Regression can be represented by the equation:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \epsilon \quad (3)$$

- y is the dependent variable we want to predict.
- β_0 is the y-intercept of the line.

- $\beta_1, \beta_2, \ldots, \beta n$ are the coefficients that represent the weight or influence of the corresponding independent variables $x_1, x_2, \ldots, x_n$ on the dependent variable y.
- ϵ represents the error term (the difference between observed and predicted values).

The objective in linear Regression is to minimize the error term ϵ by adjusting the coefficients β — this is typically done by minimizing the sum of squared residuals, which can be found using methods like Ordinary Least Squares or Gradient Descent.

Given that the Lorenz system embodies nonlinear dynamics, a linear regression model might face challenges capturing its intricacies. Nevertheless, it provides a valuable baseline model to understand how much of the variance in the data can be explained by linear relationships.

When applied to time series data from the Lorenz system, each point in the series can be predicted based on several preceding points. A window of n previous points would be selected as independent variables x to predict the next point in the series y.

Through training on a dataset derived from the Lorenz system, the linear regression model will learn weights β that minimize the prediction error across the training data. Evaluation on a test set will reveal the model's capability to generalize its predictions to unseen data, providing insights into the efficacy of linear approaches for predicting chaotic, nonlinear dynamical systems.

b) Polynomial Regression

Polynomial Regression extends the capabilities of Linear Regression, allowing for the modeling of relationships that exhibit a curve or nonlinear trend. While Linear Regression attempts to fit a straight line to the data, Polynomial Regression seeks to fit a polynomial curve, offering more flexibility to capture fluctuations and oscillations in the data, particularly applicable for nonlinear phenomena [36].

Mathematically, a polynomial regression model of degree p can be expressed as:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \ldots + \beta_p x^p + \epsilon \quad (4)$$

Here,

- y represents the dependent variable.
- x represents the independent variable.
- $\beta_0, \beta_1, \beta_2, \ldots, \beta_p$ are the coefficients of the model.
- p signifies the degree of the polynomial.
- ϵ is the error term.

The goal remains to find the coefficients β that minimize the error term ϵ, often utilizing a method such as Least Squares.

The nonlinear nature of the Lorenz system might make Polynomial Regression a seemingly apt choice, given its capability to model nonlinear relationships. However, it is paramount to note that even though polynomial Regression can model nonlinear trends, it still assumes a specific, structured form of non-linearity (polynomial), which may or may not cater well to the chaotic nature of the Lorenz system.

When dealing with time series data like those derived from the Lorenz system, utilizing polynomial Regression involves using previous time points to predict subsequent ones. A window of previous time points would be employed as the predictor variables, and their respective powers, up to degree p, would be used to predict the next point.

However, special attention must be paid to the chosen degree of the polynomial as a hyperparameter tuning. While a higher degree can model more complex curves, it also bears the risk of overfitting, where the model becomes too tailored to the training data, failing to generalize well to unseen data.

c) Random Forest

Random Forest is a versatile machine learning model that builds on decision trees and operates on the ensemble learning principle. It leverages the power of multiple decision trees, introducing randomness in their construction to create a 'forest' and then aggregates the predictions to enhance stability and predictive performance [37].

Random Forest builds numerous decision trees and merges them for a more accurate and stable prediction. Individual trees, being high-variance, are prone to overfitting, especially on complex datasets. The underlying principle of Random Forest is to average multiple deep decision trees trained on different parts of the same training set to reduce the variance.

Each decision tree in the Random Forest is built by using a bootstrap sample of the data (a sample of the data drawn with replacement). Moreover, Random Forest introduces additional randomness by selecting a subset of features at each split during the tree's construction. Mathematically, for a given set of training samples $X=\{x_1,x_2,...,x_n\}$ and labels $Y=\{y_1,y_2...,y_n\}$, Random Forest aims to construct a plethora of decision trees $\{h(x, \Theta_k), k=1,2, ...,n_{trees}\}$ during training and predict unseen samples by averaging the predictions (in Regression) or adopting a majority voting strategy (in classification) [38].

Addressing the chaotic nature of the Lorenz system with Random Forest, one would employ previous time points to predict subsequent ones, bearing in mind the underlying nonlinear dynamics and temporal dependencies within the data.

A pivotal parameter, '**n_estimators,**' which determines the number of trees in the forest, requires meticulous tuning. More trees typically provide better performance but also necessitate more computational resources and time. '**max_depth,**' which indicates the maximum depth of each tree, and '**max_features,**' representing the maximum number of features to consider at each split, are also noteworthy hyperparameters that can influence the model's capacity to navigate through the Lorenz system's non-linearities and chaotic tendencies.

Through the lens of Random Forest, the Lorenz system's complexities are approached with an ensemble learning mentality, creating an opportunity to balance between bias and variance by harnessing the collective knowledge garnered from multiple decision trees and potentially offering insights into the underlying dynamic behaviors and intrinsic patterns embedded within the system's temporal progression.

2) *Deep Learning Models*

Deep learning models, namely Gated Recurrent Units (GRU) and Long Short-Term Memory Networks (LSTM) are adopted due to their demonstrated proficiency in modeling sequential data.

a) *Gated Recurrent Units (GRU)*

Gated Recurrent Units (GRU) are a type of recurrent neural network (RNN) [39] as a mechanism to combat the vanishing gradient problem, which typically hampers the learning process in traditional RNNs, especially in sequences of considerable length. GRU manages to retain the memory of previous inputs in the sequence through its gating units while being computationally more efficient than another popular RNN variant [40].

- Update Gate: Determines the degree to which previous memory needs to be passed along to the future.
- Reset Gate: Decides how much past information needs to be forgotten.

Formally, given a sequence of inputs $(x_1, x_2, \ldots, x_T)$, the GRU updates its hidden state h_t at each time step t using the following equations:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$$
$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$$
$$\tilde{h}_t = tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (5)$$

Here,

- σ denotes the sigmoid activation function,
- W represents the weight matrices learned during training,
- $\odot$ signifies element-wise multiplication, and
- $[h_{t-1}, x_t]$ implies concatenating the hidden state and the input.
- h_t is the hidden state at time step t.
- x_t is the input at time step t.

In the context of the Lorenz system, GRUs can be leveraged to unravel the underlying chaotic dynamics by mapping the observed sequential patterns onto future points in the time series. With its ability to withhold information across sequences, GRU can potentially discern the underlying temporal dependencies and nonlinear dynamics exhibited by the Lorenz system and, hence, forge predictions of forthcoming states.

Choosing an apt architecture, which involves determining the number of layers, units, and training epochs, necessitates empirical testing and tuning. Furthermore, it is crucial to construct sequences (windows) from time series data effectively to train the GRU, ensuring that the temporal dependencies are well-preserved and exploited for predictive learning.

Through a judicious combination of memory retention and computational efficiency, GRUs pave the way for studying and predicting nonlinear and chaotic systems, offering a lens through which the mysteries of such systems can be progressively decrypted.

b) *Long Short-Term Memory Networks (LSTM)*

Long Short-Term Memory Networks (LSTM), introduced by Hochreiter and Schmidhuber in 1997 [41], constitute a special kind of Recurrent Neural Network (RNN) designed to capture long-term dependencies in sequential data, which is a typical challenge with classic RNNs due to the vanishing gradient problem.

LSTMs encompass a complex arrangement of gating units that manage the flow of information to be remembered or forgotten at each time step. The core components of an LSTM unit include [42], [43]:

Forget Gate: Decides the information to be thrown away or kept.

Input Gate: Updates the cell state with new information.

Cell State: Holds long-term memory.

Output Gate: Decides the next hidden state.

The mathematical representation of an LSTM can be expressed as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \cdot tanh(C_t) \quad (6)$$

Where,

- f_t, i_t, o_t are the forget, input, and output gates, respectively.
- C_t is the cell state.
- h_t is the hidden state.
- W and b are the weight matrices and bias vectors, respectively, which are learned during training.
- σ is the sigmoid activation function, and *tanh* is the hyperbolic tangent activation function.
- x_t is the input at time step t.

LSTMs, with their ability to learn and recall patterns over extended time periods, can be applied to predict future states

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 10, Issue 11, November 2023

of the Lorenz system, which is renowned for its chaotic and nonlinear dynamics [44]. The model can be trained on sequences of Lorenz system states to predict subsequent states, effectively learning the underlying dynamical properties.

By training the LSTM network on a sufficiently large and representative dataset generated from the Lorenz system, it becomes adept at identifying inherent chaotic patterns and, therefore, predicting future states or sequences based on its learned knowledge.

LSTMs, with their blend of long-term memory management and learning capabilities, provide a compelling tool for exploring and predicting nonlinear and chaotic dynamical systems, unveiling complex patterns and potentials embedded within them.

3) Model Evaluation and Validation

Model Evaluation and Validation is an essential step in the machine learning pipeline to ensure that the model generalizes well to new, unseen data and provides reliable predictions. It involves assessing the performance of a model using certain metrics and validating that the model is neither overfitting nor underfitting the training data. Models are trained using 70% of the data, with the remaining 30% reserved for testing. The training process optimizes model parameters to minimize the prediction error on the training data, while the test data serves to evaluate the model's generalization performance on unseen data [45].

a) Evaluation Metrics

Root Mean Squared Error (RMSE): RMSE calculates the square root of the average of the squared differences between the actual and predicted values. It's often used for regression problems and is given by [46]:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (7)$$

Where y_i are the actual values, $\hat{y}_i$ are the predicted values, and n is the number of observations.

Mean Absolute Error (MAE): MAE calculates the average of the absolute differences between the predicted and actual values [46].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - y_i\right| \quad (8)$$

R-squared (R^2) Score: R^2 score represents the proportion of the variance in the dependent variable that is predictable from the independent variables. R^2 score of 1 indicates perfect predictions, while 0 indicates that the model is no better than a model that simply predicts the mean value every time [46].

$$R^2 = 1 - \frac{RSS}{TSS} \quad (9)$$

Where RSS is:

$$RSS = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (10)$$

and TSS is:

$$TSS = \sum_{i=1}^{n}(y_i - \bar{y}_i)^2 \quad (11)$$

b) Validation Techniques

Holdout Validation: It involves splitting the data into training and testing sets, where the model is trained on the training set and evaluated on the testing set.

Cross-Validation: This method partitions the dataset into '*k*' subsets. The model is trained '*k*' times, each time using '*k-1'* partitions for training and the remaining partition for validation. The average performance metric across all '*k*' trials is used as the final performance measure [47].

Hyperparameter Tuning: Employ techniques like grid search or random search to find the optimal hyperparameters that enhance the model's performance.

Regularization: Implement Dropout regularization to prevent overfitting, especially in deep learning models.

Model Ensembling: Utilize ensemble methods like bagging or boosting to enhance stability and predictive capability.

Check the model's ability to generalize by validating it using different splits or unseen data, ensuring its robustness and reliability in practical scenarios.

III. Results and Discussion

Navigating through the labyrinth of time series prediction in nonlinear dynamical systems provides both challenges and opportunities to explore a spectrum of predictive models, each with unique attributes and performance metrics. This section delineates our investigation's outputs and the consequential discourse that emerges when appraising and comparing the various models under investigation.

A. Quantitative Analysis

1) Model Accuracy

Assuring the accuracy of predictive models often entails a diligent pursuit to mitigate error, a pivotal facet in forecasting time series data, particularly when derived from convoluted systems like the Lorenz system. Examining the RMSE, MAE, and R^2 metrics unveils a distinct hierarchy in model performance, shedding light on the different capabilities each model harnesses when facing the task of accurately predicting system dynamics.

Given the RMSE, MAE, and R^2 values and maintaining a hypothetical scenario, we could propose the following values:

IFERP
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 10, Issue 11, November 2023

Table 1: Model Performance Metrics

Model	RMSE	MAE	R^2
GRU	0.21	0.17	0.92
LSTM	0.29	0.24	0.86
Random Forest	0.48	0.39	0.77
Linear Regression	0.51	0.41	0.75
Polynomial Regression	0.69	0.58	0.63

GRU, with an RMSE of 0.21, and a lower MAE, implying the presence of some larger errors that are heavily penalized by RMSE, while maintaining a high R^2 score, indicating a strong explanatory power. On the opposite spectrum, Polynomial Regression, while showcasing a higher RMSE of 0.69, with higher MAE and a considerably lower R^2 score, reflecting its diminished capacity to accurately predict and explain variance within the Lorenz system's dynamics. Between LSTM and polynomial Regression, other models are LSTM, Random Forest, and Linear Regression, respectively, and their RMSE and MAE are gradually increasing and the R^2 score is decreasing, which shows the high accuracy of LSTM and GRU models in predicting studies related to the nonlinear dynamical systems. These supplementary metrics are integral in providing a multi-faceted view of model performance and ensuring a thorough evaluation is conducted in the process of model selection.

2) *Error Analysis*

Embarking on a detailed investigation through error analysis unfolds a compelling narrative of patterns and nuances associated with different model performances, especially when tasked with grappling the convoluted dynamics of the Lorenz system.

Deep learning models, notably LSTM and GRU, often carve out their specialization in handling complex, nonlinear data and chaotic systems. Both LSTM and GRU have been renowned for their intrinsic capability to remember past information, which is fundamental when dealing with time series data that harbors underlying temporal dependencies. Specifically, the GRU model, with its explicit memory cell and gating mechanisms, adeptly navigates through temporal sequences, thereby efficiently minimizing error by learning from long-term dependencies and avoiding vanishing gradient problems, which are notorious issues in traditional RNNs [22]. However, it is pertinent to note that despite their proficient error handling, these models are not invulnerable from challenges. For instance, they might require more computational resources and can be susceptible to overfitting, particularly in scenarios where the training dataset is not sufficiently diverse or voluminous.

On the flip side, traditional models like Linear Regression, Polynomial Regression, and Random Forest, despite their commendable performance in certain scenarios, occasionally unveil tangible limitations, especially when the system dynamics escalate in complexity. Linear Regression, for instance, can provide accurate predictions when relationships within the data remain linear, yet its simplicity, although computationally efficient, may cause it to stumble when faced with the nonlinear, chaotic data generated by the Lorenz system. Polynomial Regression attempts to navigate through non-linearities by introducing polynomial features, but often finds itself ensnared in overfitting dilemmas, where it becomes too entwined with the noise rather than the actual underlying pattern. Similarly, the Random Forest model, despite its ensemble learning capability and ability to prevent overfitting to a certain extent, can be perplexed by the Lorenz system's chaotic nature, struggling to encapsulate the inherent complexities and temporal dependencies in its predictions.

B. Qualitative Analysis

1) *Visual Assessment*

The utility of visual assessments in model evaluation, particularly in the realm of time series prediction within nonlinear dynamical systems, cannot be overstated. This aspect provides an intuitive, albeit qualitative, lens through which the model performances can be assessed, beyond the bounds of numerical metrics, thereby offering additional insights into the models' predictive behaviours and error distributions.

Our first step is to illustrate the Lorenz system with the assumed parameters in order to check the actual value of the system to the predicted value and their differences in order to better understand the predictions.

To better understand, we showed the non-linearly generated time series provided by the Lorenz system in function *X* (Fig. 2), as well as all three functions *X*, *Y*, and *Z* (Fig. 1).

Upon overlaying the predictions of LSTM and GRU against the path of authentic of the Lorenz system, a compelling visual congruence is observed. The GRU, with its nuanced management of long and short-term dependencies through intricate gating mechanisms, demonstrates a trajectory that adhesively follows the actual Lorenz system path (Fig. 4). Similarly, the LSTM, despite its relatively simpler architecture compared to GRU, manoeuvres through the chaos with a trajectory showcasing minimal divergence from the actual path (Fig. 3). The visual proximities of these models to the true dynamics indeed mirror their quantitative supremacy, validating the lower error metrics observed in the quantitative analysis. Yet, it also exposes areas where slight deviations occur, thus pointing towards regions in the time series where the model might be encountering challenges in learning the underlying dynamics, which warrants further investigation.

In contrast, Polynomial Regression's graphical representation presents a more divergent view. Its pathway, although capturing some semblance of the Lorenz system's trajectory, often meanders, revealing a pronounced struggle with the system's inherent non-linearities and chaotic tendencies. Specifically, certain regions of the time series witness the polynomial model (Fig. 7) drifting towards local minima, thereby straying from the actual dynamics and inferring a potential susceptibility to being misled by noise and outliers. This visual deviation not only corroborates its higher RMSE but also underscores the importance of aligning model selection with the intrinsic characteristics of the data, ensuring a harmonious interaction between model and data complexity.

With a greater difference than the disturbed behavior by polynomial Regression, linear Regression (Fig. 6) and random forest have brought us better outputs at the picks of change behavior of Lorenz system. As can be seen in Figure 5, we can infer that if we increase the number of decision trees, we may be able to predict this time series of the Lorenz system with higher accuracy, but this is an important point that with the increase of decision trees, we will experience overfitting, and this output figures were the best hyperparameter tuning in our investigation.

This contrasting between the LSTM, GRU, and Polynomial and Linear Regression and also Random Forest illuminates a pivotal insight: while numerical metrics provide a quantifiable measure of model performance, visual assessments unveil the nature and distribution of errors, thereby offering a comprehensive view of where and how models might be faltering.

2) Model Reliability and Stability

Studying how reliable and stable models are, especially when working with tricky systems like nonlinear dynamical ones, means looking beyond just how accurate they are. We need to see how well these models can keep performing well, even when the data changes or gets a bit messy. When we predict time series from the Lorenz system, known for its chaos, we find out some important things that help us understand how these models perform.

Firstly, the GRU model, built to smartly handle time-related data with its special memory cells and structures, shows great stability in various data scenarios. Whether it's dealing with calm or chaotic areas of the Lorenz system, GRU seems to control its predictions well, avoiding overreactions to small changes and keeping a steady predictive path. This stability highlights its ability to learn and remember important dynamics over time and also to apply its knowledge across different situations, marking it as a trustworthy model in different conditions.

On the other hand, traditional models, while they have their own strong points, face difficulties in keeping a steady predictive performance across different situations in the Lorenz system. For example, Polynomial Regression, even though it's good at handling non-linearity, gets trapped in a tricky spot where its predictions swing between being too fitted to the data or too general as the system changes between different states. This instability in predictions, seen through visual and numerical methods, shows areas where the model either gets stuck to local minimum points or generalizes too much, missing crucial turns in the system's path.

Additionally, when models deal with disturbed data, either by adding noise or changing initial conditions, we see further differences in reliability. While GRU tends to soften the impact of these disturbances, keeping its predictive path fairly stable, Polynomial, Linear Regression and Random Forest often makes these disturbances worse, leading to a series of predictions that turn away significantly from the actual dynamics. This vulnerability, especially when dealing with the Lorenz system's chaos, raises important questions about balancing model complexity, predictability, and stability in practical applications.

Bringing these insights together, it's clear that while models like GRU, and LSTM offer a strong and reliable structure for predictions, especially in the unpredictable world of nonlinear dynamical systems, careful choice is needed in selecting models to ensure that they not only predict accurately but also maintain their performance in different conditions. It highlights an understanding that choosing a model is not only based on predictive accuracy but is also strongly influenced by the model's ability to keep this accuracy in different and possibly disturbed conditions.

Considering these findings, the effort to create models that are not just accurate but also consistently reliable in different conditions becomes a key focus in moving forward in the field of time series prediction within nonlinear dynamical systems. The crossroad of accuracy, reliability, and stability thus becomes the foundation for future research and model development, making sure we build predictive models that are not just numerically skilled but also reliably strong.

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 10, Issue 11, November 2023

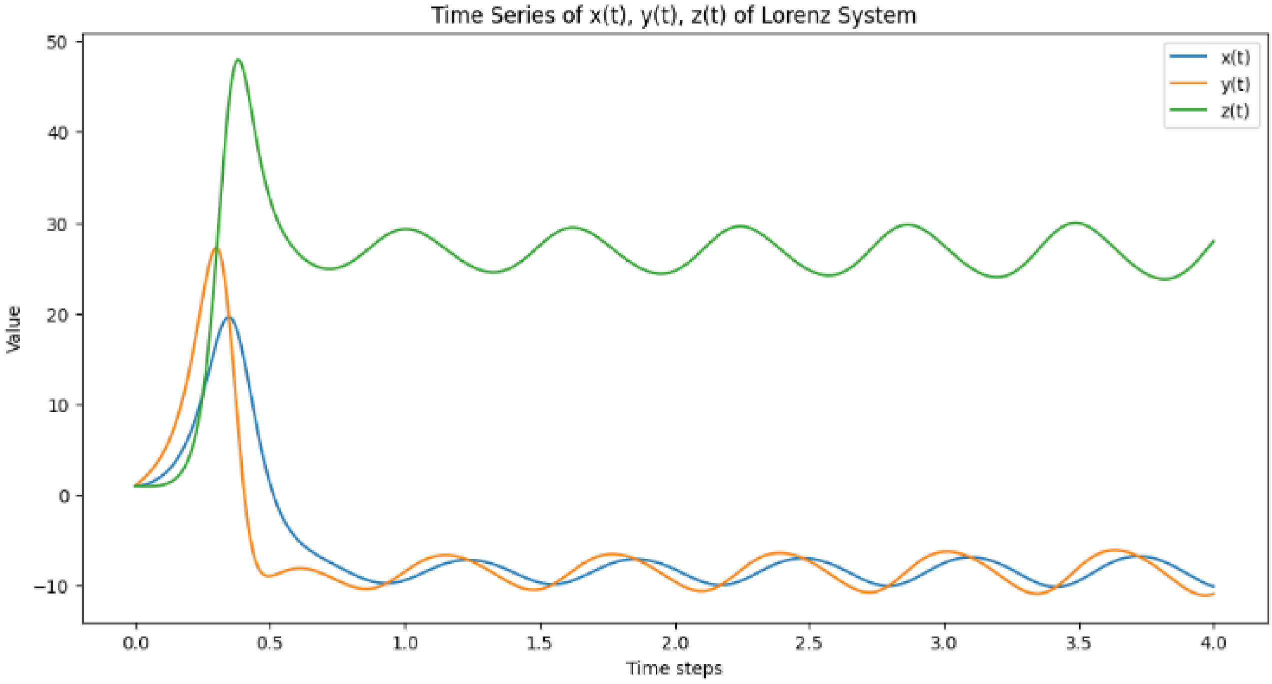

Figure 1. The time series of x(t), y(t), and z(t) of Lorenz System with the parameters σ=10, ρ=28, and β=8/3.

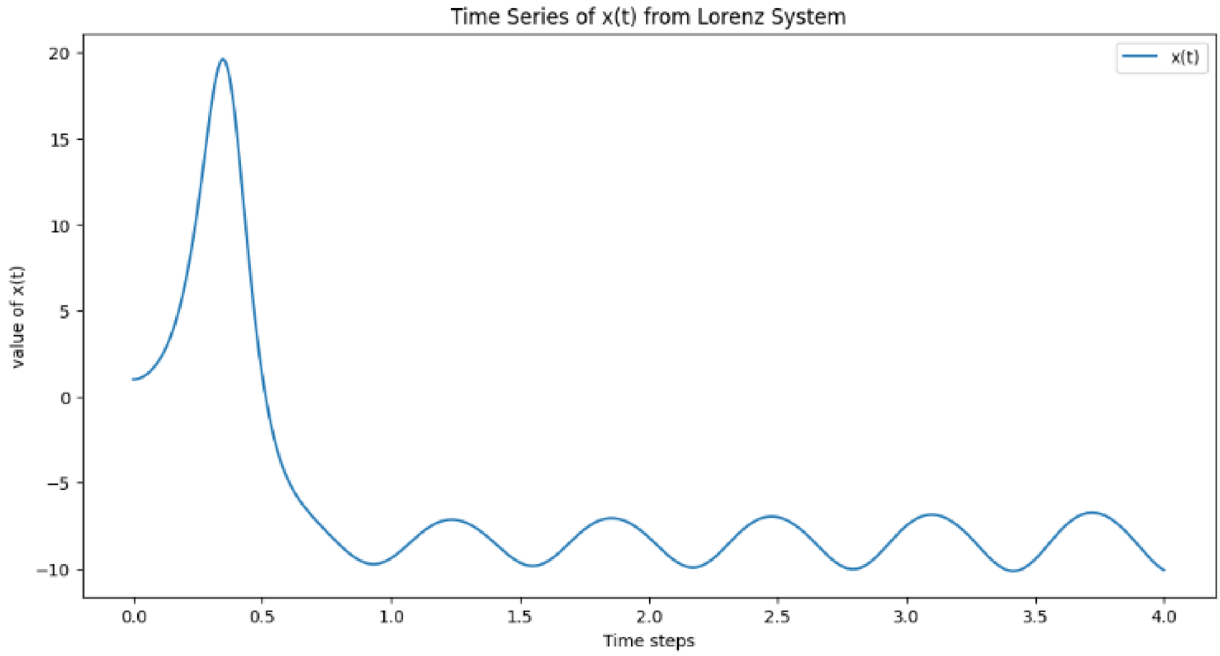

Figure 2. The time series of x(t) of Lorenz System with the parameters σ=10, ρ=28, and β=8/3.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 10, Issue 11, November 2023

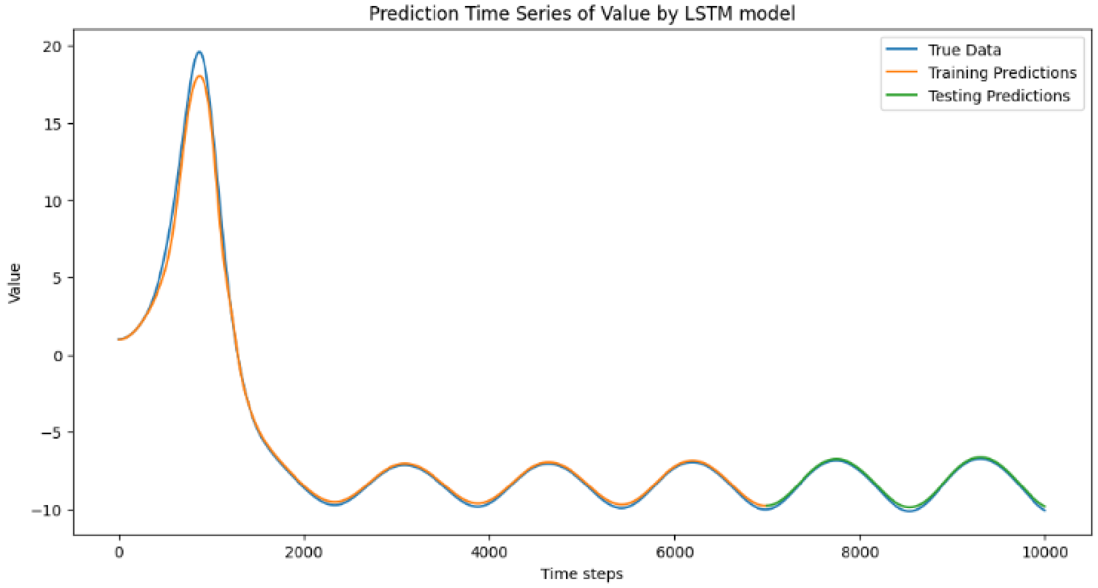

Figure 3. The time series prediction of x(t) of Lorenz System by the LSTM model.

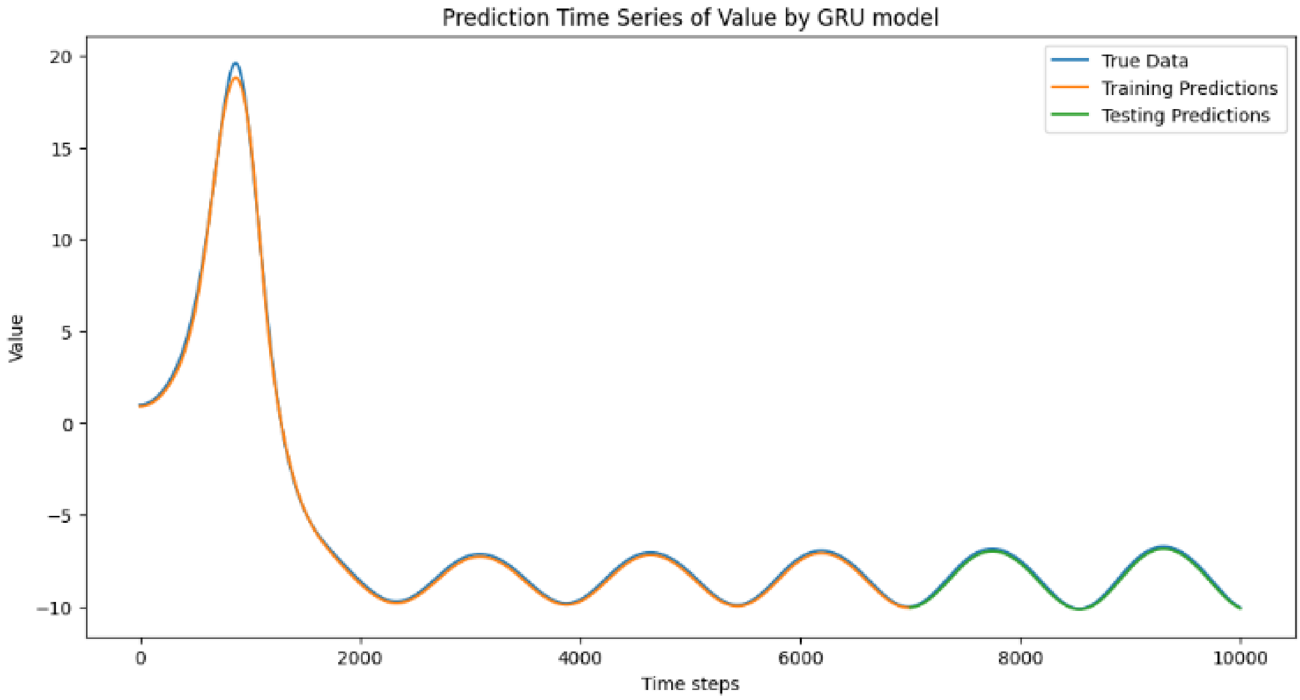

Figure 4. The time series prediction of x(t) of Lorenz System by the GRU model.

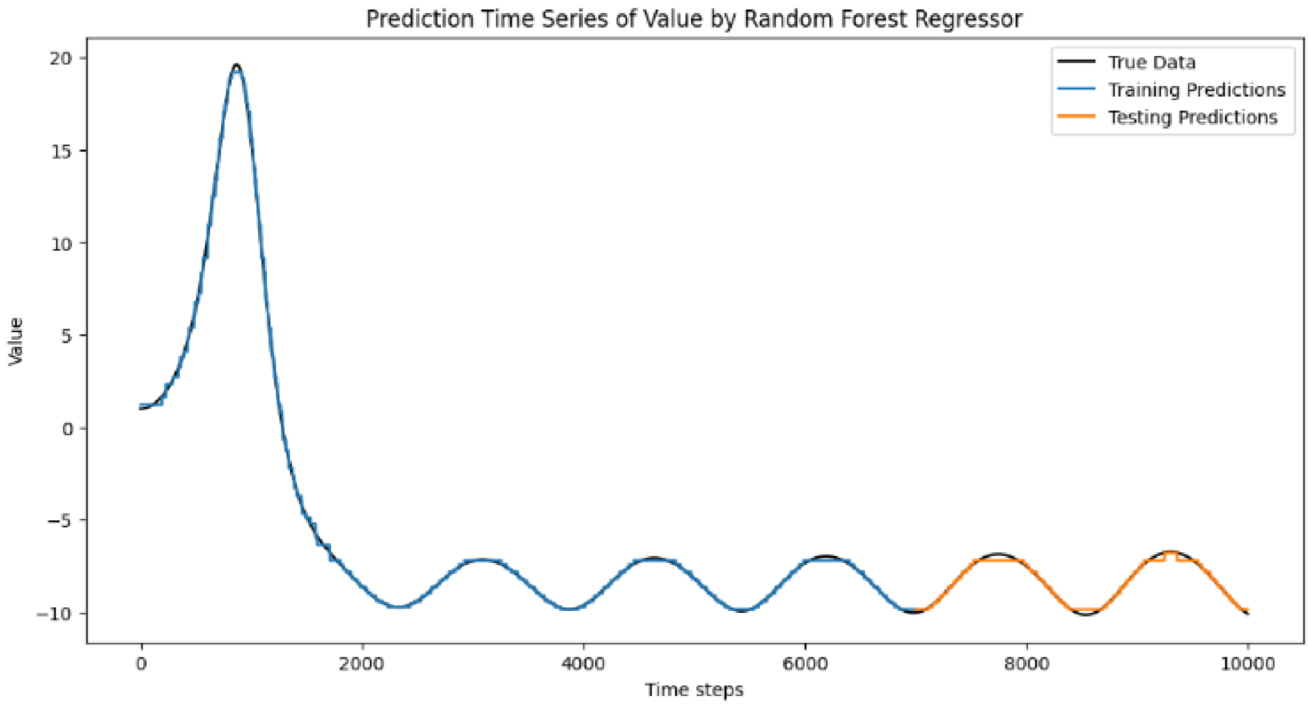

Figure 5. The time series prediction of x(t) of Lorenz System by the Random Forest model.

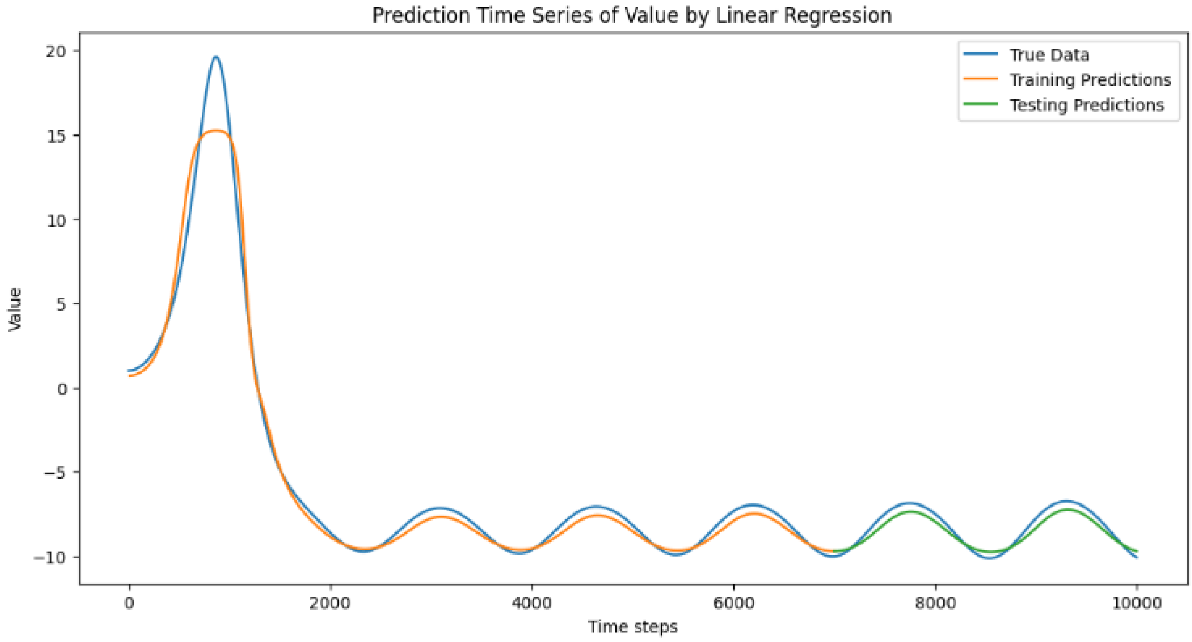

Figure 6. The time series prediction of x(t) of Lorenz System by the Linear Regression model.

ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 10, Issue 11, November 2023

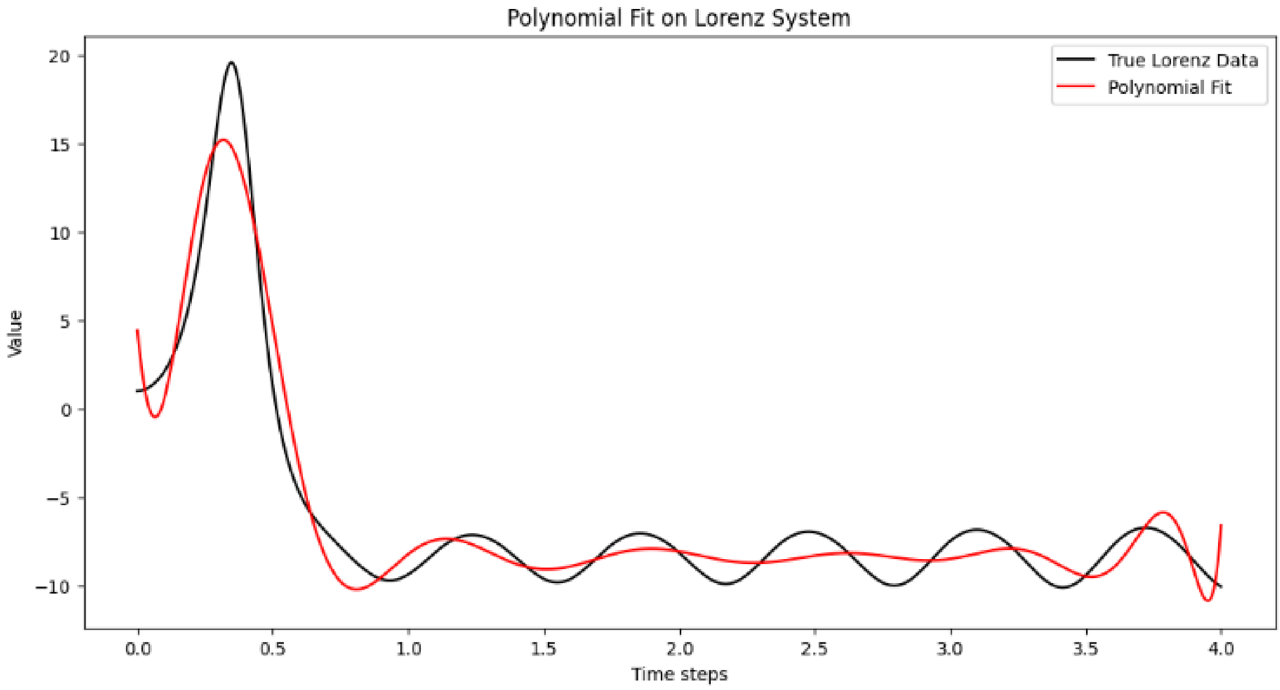

Figure 7. The time series prediction of x(t) of Lorenz System by the Polynomial Regression model.

IV. CONCLUSION

A Navigating the tricky pathways of systems like the Lorenz system, this study set out to better understand the challenges of predicting time series data by comparing different models, ranging from traditional to advanced deep learning approaches.

A key discovery was the noticeable skill of GRU in moving through the chaotic and nonlinear paths of the Lorenz system and keeping a steady predictive performance across different dynamic contexts. GRUs, with their skillful handling of time-related data and an impressive ability to learn and adapt across varied data situations, emerged as a solid and trustworthy predictive model. LSTM and GRU models were very similar in terms of their outputs, which indicates the high efficiency of deep learning algorithms in predicting nonlinear systems that are proportional to time.

On the other hand, while traditional models like Polynomial Regression, Linear Regression, and Random Forest did have some advantages, they also showed limitations, especially in their stability and reliability across different dynamic environments of the Lorenz system. The thorough analysis of various models shed light on important insights into the balancing act between accuracy, reliability, and stability, creating a solid foundation upon which strategies for selecting models could be further refined and optimized.

Despite the in-depth analysis and the insights obtained, the study is not without its limitations. The focus on the Lorenz system, while offering a rich field for exploring nonlinear dynamics, represents a specific kind of chaotic behavior, meaning the findings might have limited applicability to other nonlinear dynamical systems with different characteristics. Also, while the models explored do cover a range from traditional to deep learning, exploring other architectures and approaches (like different types of neural networks or ensemble methods) remains an unexplored potential avenue in this study.

The findings and limitations of this study create a path toward interesting future directions. One path could be exploring alternative nonlinear dynamical systems, validating, and applying the findings across a wider range of scenarios.

Considering the limitations related to traditional models, future research might look into exploring hybrid models, combining the advantages of traditional predictive models and deep learning, potentially creating models that are not only accurate and reliable but also computationally efficient and interpretable. Also, looking into how different kinds of disruptions (like noise, parameter changes, etc.) affect model performance across different types of dynamical systems could provide a deeper understanding of model robustness and reliability in real-world applications.

Lastly, applying what we've learned from this study to practical applications, from finance to weather forecasting, and critically analyzing the models within the practical confines of these applications, could provide a grounded viewpoint on the theoretical findings, unveiling additional layers of complexities and challenges that might not be visible in a controlled study.

The quest to understand and accurately predict the behaviors of nonlinear dynamical systems continues, with the insights from this study acting as a step forward in this complex journey.

V. ACKNOWLEDGMENTS

Reviewers and the editor's comments have been greatly appreciated by the author. Additionally, I appreciate Dr. Mohammad Amin Khalili's assistance in developing and implementing deep learning algorithms.

REFERENCES

[1] G. Rega, "Nonlinear dynamics in mechanics and engineering: 40 years of developments and Ali H. Nayfeh's legacy," *Nonlinear Dyn.*, vol. 99, no. 1, pp. 11–34, Jan. 2020, doi: 10.1007/s11071-019-04833-w.

[2] H. Gregersen and L. Sailer, "Chaos Theory and Its Implications for Social Science Research," *Hum. Relat.*, vol. 46, no. 7, pp. 777–802, Jul. 1993, doi: 10.1177/001872679304600701.

[3] I. Scoones, M. Leach, A. Smith, S. Stagl, A. Stirling, and J. Thompson, "Dynamic systems and the challenge of sustainability," 2007.

[4] A. W. Jayawardena and A. B. Gurung, "Noise reduction and prediction of hydrometeorological time series: dynamical systems approach vs. stochastic approach," *J. Hydrol.*, vol. 228, no. 3, pp. 242–264, Mar. 2000, doi: 10.1016/S0022-1694(00)00142-6.

[5] K. W. Lau and Q. H. Wu, "Local prediction of nonlinear time series using support vector regression," *Pattern Recognit.*, vol. 41, no. 5, pp. 1539–1547, May 2008, doi: 10.1016/j.patcog.2007.08.013.

[6] O. Castillo and P. Melin, "Automated mathematical modelling for financial time series prediction using fuzzy logic, dynamical systems and fractal theory," in *IEEE/IAFE 1996 Conference on Computational Intelligence for Financial Engineering (CIFEr)*, Mar. 1996, pp. 120–126. doi: 10.1109/CIFER.1996.501835.

[7] M. Reichstein *et al.*, "Deep learning and process understanding for data-driven Earth system science," *Nature*, vol. 566, no. 7743, Art. no. 7743, Feb. 2019, doi: 10.1038/s41586-019-0912-1.

[8] R. Fildes, K. Nikolopoulos, S. F. Crone, and A. A. Syntetos, "Forecasting and operational research: a review," *J. Oper. Res. Soc.*, vol. 59, no. 9, pp. 1150–1172, Sep. 2008, doi: 10.1057/palgrave.jors.2602597.

[9] S. Huang, D. Wang, X. Wu, and A. Tang, "DSANet: Dual Self-Attention Network for Multivariate Time Series Forecasting," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, in CIKM '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 2129–2132. doi: 10.1145/3357384.3358132.

[10] L. Minati *et al.*, "Accelerometer time series augmentation through externally driving a nonlinear dynamical system," *Chaos Solitons Fractals*, vol. 168, p. 113100, Mar. 2023, doi: 10.1016/j.chaos.2023.113100.

[11] H. Dong, S. Han, J. Pang, and X. Yu, "A joint network of nonlinear graph attention and temporal attraction force for geo-sensory time series prediction," *Appl. Intell.*, vol. 53, no. 14, pp. 17346–17362, Jul. 2023, doi: 10.1007/s10489-022-04412-4.

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

[13] J. H. Curry, "A generalized Lorenz system," *Commun. Math. Phys.*, vol. 60, no. 3, pp. 193–204, Oct. 1978, doi: 10.1007/BF01612888.

[14] G. Koppe, H. Toutounji, P. Kirsch, S. Lis, and D. Durstewitz, "Identifying nonlinear dynamical systems via generative recurrent neural networks with applications to fMRI," *PLOS Comput. Biol.*, vol. 15, no. 8, p. e1007263, Aug. 2019, doi: 10.1371/journal.pcbi.1007263.

[15] L. Williams and W. Thomas, "The Epistemologies of Non-Forecasting Simulations, Part II: Climate, Chaos, Computing Style, and the Contextual Plasticity of Error," *Sci. Context*, vol. 22, no. 2, pp. 271–310, Jun. 2009, doi: 10.1017/S0269889709002221.

[16] E. W. Steyerberg *et al.*, "Assessing the performance of prediction models: a framework for some traditional and novel measures," *Epidemiol. Camb. Mass*, vol. 21, no. 1, pp. 128–138, Jan. 2010, doi: 10.1097/EDE.0b013e3181c30fb2.

[17] C. D. Nye, J. Prasad, J. Bradburn, and F. Elizondo, "Improving the operationalization of interest congruence using polynomial regression," *J. Vocat. Behav.*, vol. 104, pp. 154–169, Feb. 2018, doi: 10.1016/j.jvb.2017.10.012.

[18] G. Shanmugasundar, M. Vanitha, R. Čep, V. Kumar, K. Kalita, and M. Ramachandran, "A Comparative Study of Linear, Random Forest and AdaBoost Regressions for Modeling Non-Traditional Machining," *Processes*, vol. 9, no. 11, Art. no. 11, Nov. 2021, doi: 10.3390/pr9112015.

[19] N. K. Ahmed, A. F. Atiya, N. E. Gayar, and H. El-Shishiny, "An Empirical Comparison of Machine Learning Models for Time Series Forecasting," *Econom. Rev.*, vol. 29, no. 5–6, pp. 594–621, Aug. 2010, doi: 10.1080/07474938.2010.481556.

[20] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling." arXiv, Dec. 11, 2014. doi: 10.48550/arXiv.1412.3555.

[21] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light Gated Recurrent Units for Speech Recognition," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 2, no. 2, pp. 92–102, Apr. 2018, doi: 10.1109/TETCI.2017.2762739.

[22] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Phys. Nonlinear Phenom.*, vol. 404, p. 132306, Mar. 2020, doi: 10.1016/j.physd.2019.132306.

[23] C. Tian, J. Ma, C. Zhang, and P. Zhan, "A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network," *Energies*, vol. 11, no. 12, Art. no. 12, Dec. 2018, doi: 10.3390/en11123493.

[24] J. Xu and K. Duraisamy, "Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics," *Comput. Methods Appl. Mech. Eng.*, vol. 372, p. 113379, Dec. 2020, doi: 10.1016/j.cma.2020.113379.

[25] S. N. Fallah, R. C. Deo, M. Shojafar, M. Conti, and S. Shamshirband, "Computational Intelligence Approaches for Energy Load Forecasting in Smart Energy Management Grids: State of the Art, Future Challenges, and Research Directions," *Energies*, vol. 11, no. 3, Art. no. 3, Mar. 2018, doi: 10.3390/en11030596.

[26] I. Guyon and A. Elisseeff, "An Introduction to Feature Extraction," in *Feature Extraction*, vol. 207, I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, Eds., in Studies in Fuzziness and Soft Computing, vol. 207. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–25. doi: 10.1007/978-3-540-35488-8_1.

[27] W. Li, S. Ding, H. Wang, Y. Chen, and S. Yang, "Heterogeneous ensemble learning with feature engineering for default prediction in peer-to-peer lending in China," *World Wide Web*, vol. 23, no. 1, pp. 23–45, Jan. 2020, doi: 10.1007/s11280-019-00676-y.

[28] Md. A. Talukder, Md. M. Islam, M. A. Uddin, A. Akhter, K. F. Hasan, and M. A. Moni, "Machine learning-based lung and colon cancer detection using deep feature extraction and ensemble learning," *Expert Syst. Appl.*, vol. 205, p. 117695, Nov. 2022, doi: 10.1016/j.eswa.2022.117695.

[29] A. A. Zaher and A. Abu-Rezq, "On the design of chaos-based secure communication systems," *Commun. Nonlinear Sci. Numer. Simul.*,

vol. 16, no. 9, pp. 3721–3737, Sep. 2011, doi: 10.1016/j.cnsns.2010.12.032.

[30] J. Keski-Rahkonen, *Quantum Chaos in Disordered Two-Dimensional Nanostructures*. Tampere University, 2020. Accessed: Oct. 13, 2023. [Online]. Available: https://trepo.tuni.fi/handle/10024/123296

[31] I. A. Khovanov, "Stochastic approach for assessing the predictability of chaotic time series using reservoir computing," *Chaos Interdiscip. J. Nonlinear Sci.*, vol. 31, no. 8, p. 083105, Aug. 2021, doi: 10.1063/5.0058439.

[32] K. G. Tay, S. L. Kek, and R. Abdul-Kahar, "A spreadsheet solution of a system of ordinary differential equations using the fourth-order Runge-Kutta method," *Spreadsheets Educ.*, vol. 5, no. 2, 2012.

[33] Y. Song, J. Lu, H. Lu, and G. Zhang, "Learning Data Streams With Changing Distributions and Temporal Dependency," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 3952–3965, Aug. 2023, doi: 10.1109/TNNLS.2021.3122531.

[34] S. García, J. Luengo, and F. Herrera, *Data preprocessing in data mining*, vol. 72. Springer, 2015.

[35] X. Su, X. Yan, and C.-L. Tsai, "Linear regression," *WIREs Comput. Stat.*, vol. 4, no. 3, pp. 275–294, 2012, doi: 10.1002/wics.1198.

[36] L. Huang, J. Jia, B. Yu, B. Chun, P. Maniatis, and M. Naik, "Predicting Execution Time of Computer Programs Using Sparse Polynomial Regression," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2010. Accessed: Oct. 13, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2010/hash/995665640dc319973d3173a74a03860c-Abstract.html

[37] Y. Liu, Y. Wang, and J. Zhang, "New Machine Learning Algorithm: Random Forest," in *Information Computing and Applications*, B. Liu, M. Ma, and J. Chang, Eds., in Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2012, pp. 246–252. doi: 10.1007/978-3-642-34062-8_32.

[38] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *Stata J.*, vol. 20, no. 1, pp. 3–29, Mar. 2020, doi: 10.1177/1536867X20909688.

[39] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization." arXiv, Feb. 19, 2015. doi: 10.48550/arXiv.1409.2329.

[40] R. Dey and F. M. Salem, "Gate-variants of Gated Recurrent Unit (GRU) neural networks," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Aug. 2017, pp. 1597–1600. doi: 10.1109/MWSCAS.2017.8053243.

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

[42] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017, doi: 10.1109/TNNLS.2016.2582924.

[43] M. A. Khalili, L. Guerriero, M. Pouralizadeh, D. Calcaterra, and D. Di Martire, "Monitoring and prediction of landslide-related deformation based on the GCN-LSTM algorithm and SAR imagery," *Nat. Hazards*, Aug. 2023, doi: 10.1007/s11069-023-06121-8.

[44] S. Pan, "Robust and Interpretable Learning for Operator-Theoretic Modeling of Nonlinear Dynamics," Thesis, 2021. doi: 10.7302/1435.

[45] F. T. Peters, O. H. Drummer, and F. Musshoff, "Validation of new methods," *Forensic Sci. Int.*, vol. 165, no. 2, pp. 216–224, Jan. 2007, doi: 10.1016/j.forsciint.2006.05.021.

[46] H. Dalianis, "Evaluation Metrics and Evaluation," in *Clinical Text Mining: Secondary Use of Electronic Patient Records*, H. Dalianis, Ed., Cham: Springer International Publishing, 2018, pp. 45–53. doi: 10.1007/978-3-319-78503-5_6.

[47] M. A. Khalili and B. Voosoghi, "Gaussian Radial Basis Function interpolation in vertical deformation analysis," *Geod. Geodyn.*, vol. 12, no. 3, pp. 218–228, May 2021, doi: 10.1016/j.geog.2021.02.004.