Journal of NAMP 69, 1 (2025) 139-152

The Nigerian Association of Mathematical Physics

Journal homepage: https://nampjournals.org.ng

MODELING AND PREDICTION OF FRACTIONAL-ORDER CHAOTIC LORENZ SYSTEM USING RNN AND LSTM NETWORKS.

A. E. Adeniji[1], K. S. Oyeleke[2], K.S. Ojo[3,] and A. M. Lasisi[4]

[1]*Department of Physical Sciences, Bells University of Technology, Ogun state, Nigeria.*

[2]*Department of Physical Sciences, Lagos State University of Science and Technology, Ikorodu, Lagos, Nigeria*

[3]*Department of Physics, University of Lagos, Akoka, Lagos, Nigeria*

[4]*Department of Physics, Ajayi Crowther University, Oyo, Oyo State.*

ARTICLE INFO

Article history:
Received xxxxx
Revised xxxxx
Accepted xxxxx
Available online xxxxx

Keywords:
Complexities, Fractional-order systems, Initial condition, Long Short Term Memory (LSTM), Lorenz System, Recurrent Neural Networks.

ABSTRACT

The complexities inherent underlying in the chaotic systems have made long term prediction impossible due to their sensitivity dependence on initial condition. There is need to employ machine learning to detect intricacies as they can capture patterns in complex system and also, extract fractional order behaviour from data. In this study, the comparison between the performance of Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks in forecasting fractional-order Lorenz chaotic time series data was investigated. The results show that training and test data for LSTM networks have lower Root Mean Squared Error (RMSE) values than the RNN values, indicating superior generalization to unseen data. By effectively modeling long-term dependencies of the chaotic system, LSTM enhances prediction accuracy and performance compared to traditional RNNs. Accordingly, these findings imply that LSTM networks are more capable of modeling fractional-order dynamics, chaotic systems, thus being more valuable in applications.

.

1. Introduction

Traditional integer-order calculus is extended to arbitrary orders in fractional-order calculus (FOC) [1]. FOC offers a novel method for simulating and analyzing physical processes by providing more adjustable parameters. By using fractional derivatives and integrals, it is especially helpful for characterizing systems with memory effects or long-range interactions, which are prevalent in a variety of physical, biological, and engineering contexts.

*Corresponding author: A. E. Adeniji

E-mail address: aeadeniji@bellsuniversity.edu.ng

https://doi.org/10.60787/jnamp.vol69no1.487

The dynamic characteristics of semiconductors that conventional integer-order models are unable to capture are described by fractional models in electronics [2]. Because of their powerful modeling capabilities, a wide range of fractional models have been created and extensively used in recent decades in domains including electrical engineering [3], signal processing [4], neural networks [5], and others.

An intriguing subfield of chaos theory that applies the classical theory of chaos to systems with fractional-order differential equations is that of fractional-order chaotic systems. These systems have complicated behaviors with erratic, aperiodic motion, yet because of the non-integer order of differentiation, they have some special characteristics. Due to the severe dependence of fractional-order chaotic systems on initial conditions, even slight modifications to the initial state can result in trajectories that diverge significantly over time. They also exhibit characteristics such as the lack of periodic orbits, topological mixing, and unpredictable behavior. Applications in practical fields require an understanding and the ability to govern fractional-order chaotic systems. In order to secure communication or manipulate chaotic dynamics for desired results, methods such as chaos management and synchronization are employed to stabilize chaotic behavior or synchronize chaotic systems.

One distinct type of nonlinear system that is extremely unpredictable is a chaotic system. Even more sophisticated in their behavior, fractional-order chaotic systems are crucial to the encryption and decoding of secure communications [6, 7]. Since Leon Chua initially presented the well-known Chua's circuit [8], the application of chaotic systems has gained significant attention. In accordance with these concepts, Pham et al., 2017 [9] created a three-dimensional fractional-order chaotic system without equilibrium, while [10] created an electronic circuit to produce a 4-D fractional-order chaotic system. The problem of approximating fractional-order systems using low-order rational functions has been tackled through optimization strategies [11-13]. However, fractional-order chaotic systems are difficult to realize in engineering applications due to their complexity and unpredictability. Specifically, fractional-order circuit units, which are composed of numerous electrical components, are also a source of complexity. In fact, there are two primary causes of uncertainty: (1) the chaotic system's extreme unpredictability and non-linearity, and (2) the discrepancies between nominal and actual values seen in electrical circuit components [14].

As for potential pairings between machine learning and fractional dynamics, we discover that, as demonstrated in [15] or [16] machine learning can be utilized to extract fractional order behavior from data. In general, the least number of publications were discovered in this category. This could be because fractional order dynamics modeling necessitates a deep understanding of machine learning and mathematics.

Research on employing artificial neural networks (ANN) to forecast and describe chaotic systems began about ten years ago, with the work of [17]. The authors conducted a study in 2009 to investigate the feasibility of using artificial neural networks (ANN) for the purpose of forecasting the outputs of nonlinear dynamic systems. They trained the artificial neural networks (ANN) with a Nonlinear Autoregressive Moving Averages model with Exogenous input (NARMAX) using the Lorenz System, which produced a chaotic data set. Phase diagrams, statistical studies, and

Lyapunov exponents were utilized to assess the neural network output and contrast it with the real Lorenz System. This paper was substantially motivated by their efforts.

In this research, a chaotic input signal generated from fractional order chaotic system was used to compare the one-step-ahead predictions provided by traditional Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks and also to evaluate both networks' modeling and prediction abilities.

2. Fractional Order Chaotic Lorentz System

The Lorenz fractional order system is given by the nonlinear differential equation below

$$\frac{d^\alpha x}{dt^\alpha} = a(y - x) \quad (1)$$

$$\frac{d^\alpha y}{dt^\alpha} = (c - a)x - xy + cy \quad (2)$$

$$\frac{d^\alpha z}{dt^\alpha} = xy - bz \quad (3)$$

where a is the Pranti number, c is the Rayleigh number, b is the magnitude of the system, x is convection overturning, y is the horizontal temperature difference, and z is the vertical temperature difference. The above equation is fractional order form of integer form of the Lorenz [18, 19]. This system of equations was created by Lorenz to show that weather unpredictability is not caused by random terms of unknown origin, but rather by the properties of the Navier-Stokes equation solutions. His fundamental claim was that there might be an attractive and invariant set in a deterministic system, where the dynamics are still constrained but also linearly unstable [20]. When such strange attractors exist, trajectories are chaotic and appear random [20]. The two surfaces of a strange attractor only appear to merge because the uniqueness theorem suggests that trajectories cannot overlap or merge. Lorenz came to the conclusion that this seeming merger takes place on a "infinite complex of surfaces". These days, we call this "infinite complex of surfaces" a fractal. Additionally, two trajectories that start relatively near to one another will soon split and take very distinct routes because of the attractor's sensitive dependence on initial conditions. This has important practical ramifications because long-term prediction is very difficult due to the system's rapid amplification of tiny uncertainty. To have a more complex dynamic, the fractional order form of the Lorenz system was realized, and the detail of the dynamical behaviour can be seen in [21]. The phase space portrait and the corresponding trajectory are shown in the Figures 1 and 2 below.

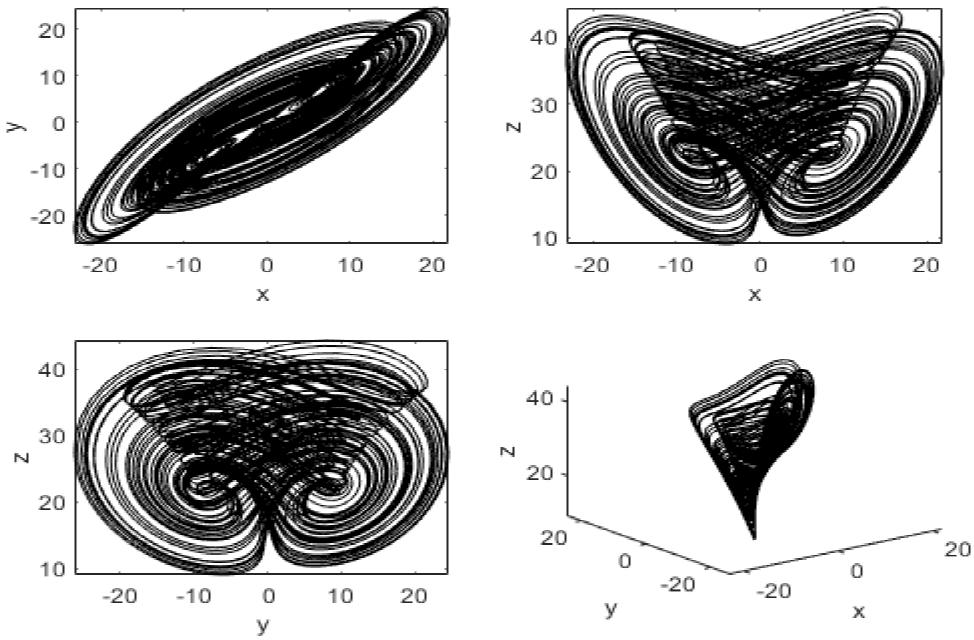

Figure 1: Phase portrait of fractional order chaotic Lorenz system at $a = 35, b = 3, c = 8/3$ with initial conditions $-10, -8,\ 37$ and $\alpha = 0.95$

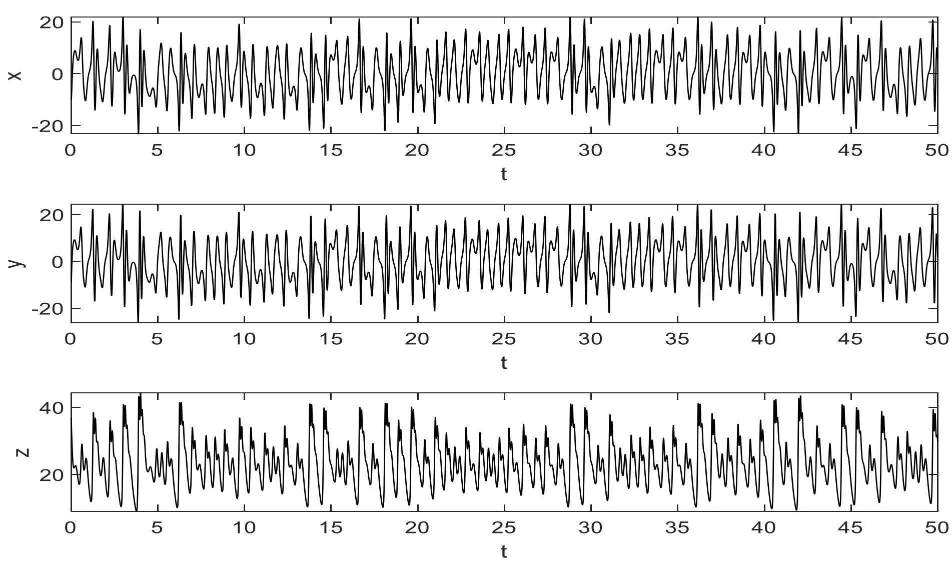

Figure 2: Time series of the state variable of fractional order chaotic Lorenz system at $a = 35, b = 3, c = 8/3$ with initial conditions $-10, -8,\ 37$ and $\alpha = 0.95$

3. Long Short-Term Memory Units (LSTMs)

LSTMs [22] are a type of long short-term memory unit that was created especially to handle the vanishing gradient problem. LSTMs allow RNNs to learn dependencies over considerably longer sequences, well beyond 1000-time steps, by preserving a more consistent error signal [23]. Gated cells, which store extra information outside of the normal neural network flow, are integrated to do this [23, 24]. Specialized gates govern the functions of an LSTM. These gates include an input gate I_t for adding new data to the cell, an output gate O_t for retrieving cell entries, and a forget-gate F_t for resetting cell contents (figure 3). Equations 13, 14, and 15 provide specific details on the calculations that control these gates.

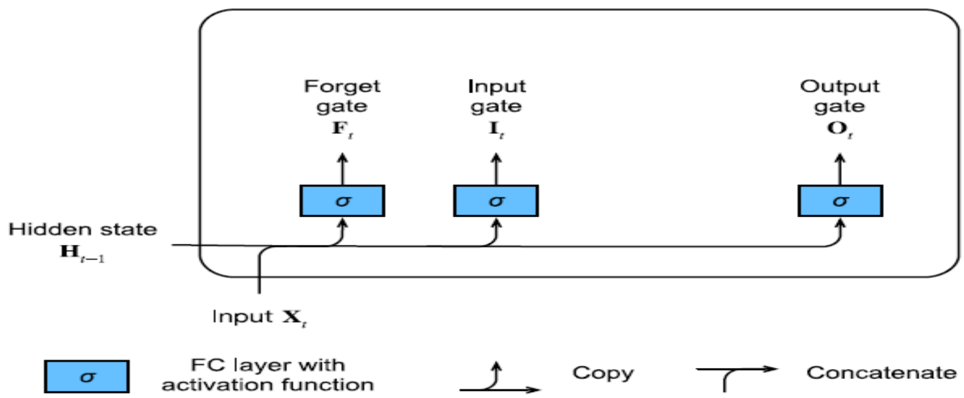

Figure 3: Calculation of input, forget, and output gates in an LSTM [25].

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o) \quad (13)$$
$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i) \quad (14)$$
$$F_t = \sigma\left(X_t W_{xf} + H_{t-1} W_{hf} + b_f\right) \quad (15)$$

Where $W_{xi}, W_{xf}, W_{xo} \in \mathbb{R}^{d\times h}$ and $W_{hi}, W_{hf}, W_{ho} \in \mathbb{R}^{h\times h}$ are weight matrices while b_i, b_f, $b_o \in \mathbb{R}^{1\times h}$ are their respective biases. The sigmoid activation function σ is used to transform the output $\in$ (0, 1) which each result in a vector with entries $\in$ (0, 1).

Next, we introduce a candidate memory cell $\tilde{C}_t \in \mathbb{R}^{n\times h}$, which performs computations similarly to the previously mentioned gates but uses a tanh activation function to produce an output in the range $(-1, 1)$. This memory cell also has its own set of weights $w_{xc} \in \mathbb{R}^{d\times h}$, $w_{hc} \in \mathbb{R}^{h\times h}$, and biases $b_c \in \mathbb{R}^{`1\times h}$(figure 4). The corresponding computation is outlined in Equation 16.

$$\tilde{C}_t = tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c) \quad (16)$$

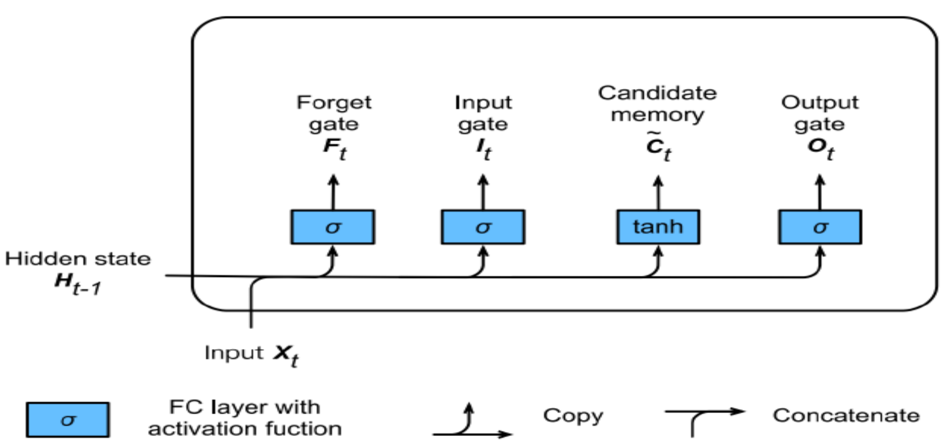

Figure 4: Computation of candidate memory cells in LSTM [25].

To integrate various components, we introduce the previous memory content $C_{t-1} \epsilon\ \mathbb{R}^{n\times h}$, which, in combination with the gates, determines how much of the previous memory content to retain in order to update to the new memory content C_t (figure 5). This process is described in Equation 17, where ⊙ represents element-wise multiplication.

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t \qquad (17)$$

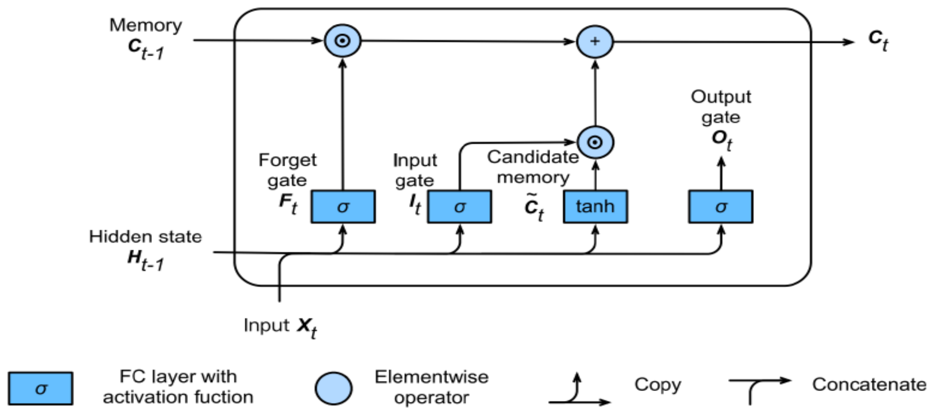

Figure 5: Computing the memory cell internal state in an LSTM model [25].

The final step is to incorporate the computation of the hidden states $H_t\ \epsilon\ \mathbb{R}^{n\times h}$ into the framework, as shown in Equation 18.

$$H_t = O_t \odot tanh(C_t) \qquad (18)$$

With the tanh function we ensure that each element of H_t is $\epsilon\ (-1, 1)$. The full LSTM framework can be seen in Figure 6.

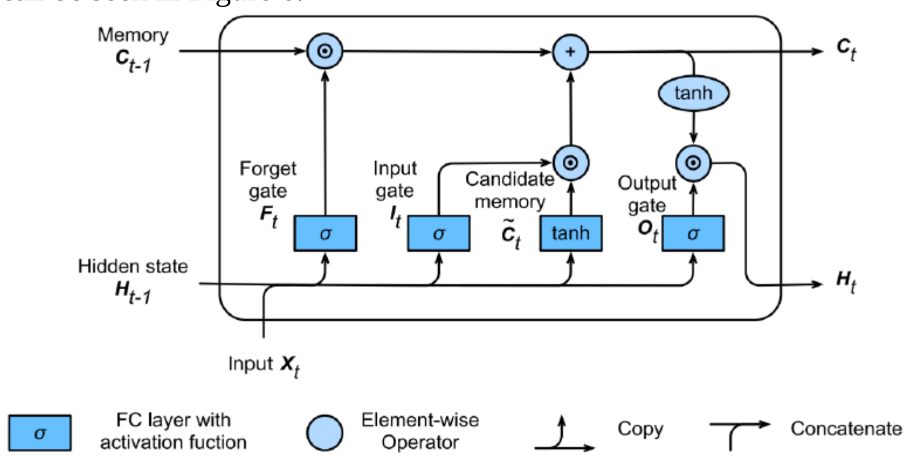

Figure 6: Computation of hidden state in an LSTM [25].

The Root Mean Squared Error (RMSE) is a commonly used loss function for regression tasks, including those involving Long Short-Term Memory (LSTM) networks. The RMSE measures the square root of the average of the squared differences between the predicted and true values.

Results and Discussion

Equations (1), (2), and (3) yielded data from solving the fractional order Lorenz system, which we used to train two models: an LSTM network and a conventional RNN. These models were trained to forecast the state of the fractional order Lorenz System one step ahead of time. The simulated data is structured so that neural network models can utilize 30% of the data for testing, while 70% of the data for training [26-29].

To train the model, two fundamental components must be established: a training dataset and a static architecture. The testing dataset, like the training set, consists of three sequences. We need to reshape the data into this required configuration. Both the training and testing inputs are in the form of 10 size sequences with 3,297 and 6,696 time steps respectively in (x, y, z) format.

Recurrent neural networks (RNNs) are a type of neural network that uses loops to preserve information throughout the network while receiving sequential input by processing each element one at a time. These networks are trained using specialized weight adjustment methods, like Backpropagation Through Time, which employs a gradient descent-based learning strategy [30]. The gradient problem associated with RNN during propagation can either increase or decrease exponentially as the number of time steps increases. which can result in the gradients either vanishing or exploding. This limits the network's capacity to efficiently learn over time the connection between previous and subsequent states in the system [30, 31].

The vanishing and exploding gradient problem can be effectively resolved by using LSTM neural networks [32]. By storing data for extended periods of time, these customized RNNs are able to learn long-term dependencies. LSTM networks use a gated cell mechanism that retains the error and allows it to be backpropagated over time and between layers, enabling the network to learn over numerous time steps [33, 34]. While maintaining historical system state information, time series sequences can be processed and predicted by LSTM networks. As the system's complexity rises, they perform better than normal RNNs at learning long-term dependencies [35]. The fractional order of Lorenz chaotic time series plots, which are displayed in Figure 7, provides insight into the data's complexity as well as the LSTM's capacity for learning and generalization. It draws attention to the complex interactions between machine learning and chaotic dynamics, guiding model development and evaluation of models. Using fractional-order Lorenz chaotic time series is important because it improves the LSTM network's capacity to represent dynamic, complex systems, which improves generalization and predictive performance in chaotic settings. The sensitive dependence on initial conditions that characterizes chaotic systems is depicted in the plot of the fractional order Lorenz system. Even slight variations in initial conditions can lead to vastly different trajectories [19], emphasizing the complexity that LSTMs must learn to model. The plot can reveal various complexities inherent in the chaotic system. A rich, convoluted trajectory suggests that the LSTM needs to extract meaningful features over time from the sequential data [36]. The ability of LSTMs to maintain memory over longer sequences is crucial for understanding these complex patterns [22].

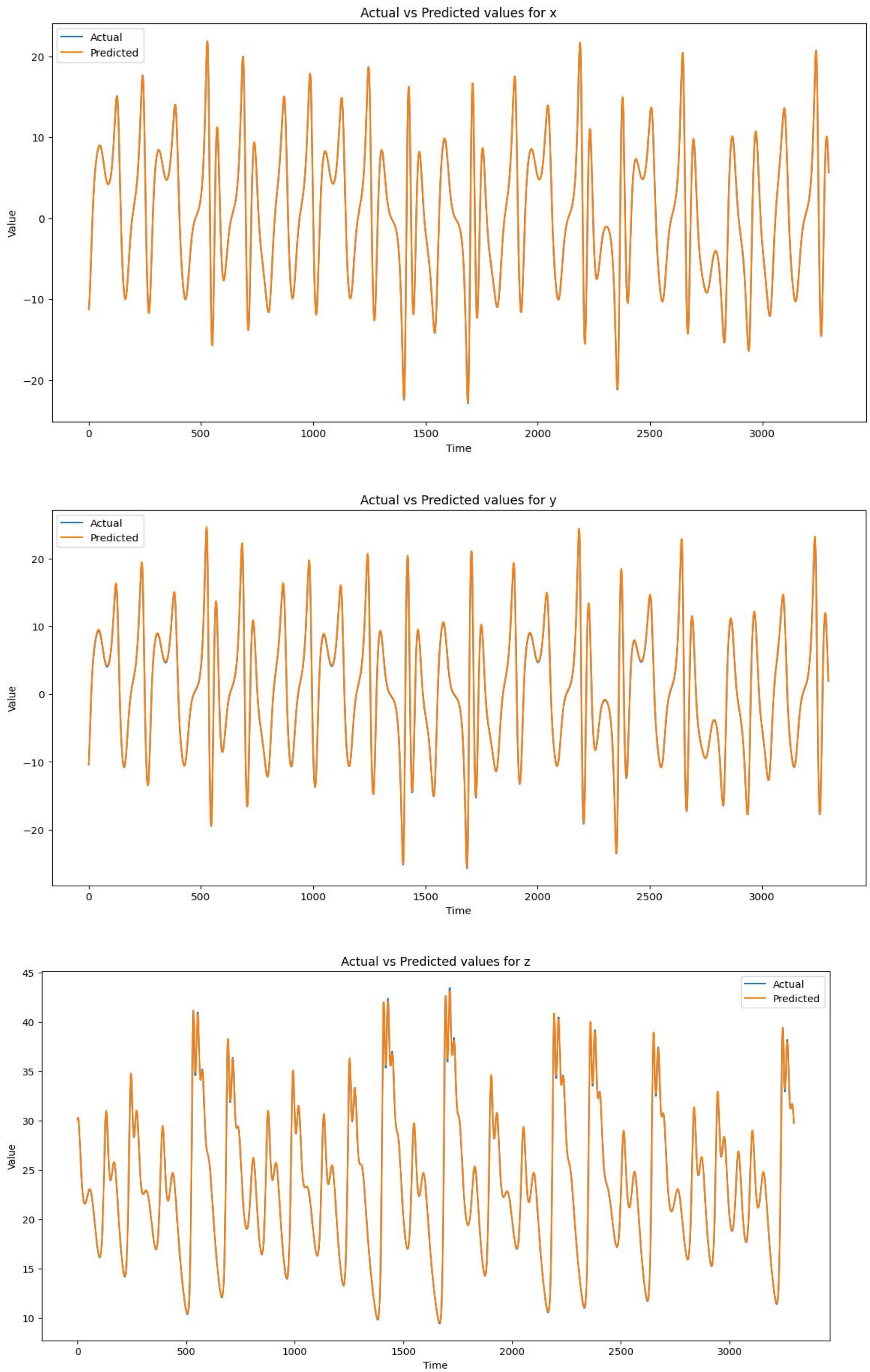

Figure 7: the fractional order Lorenz chaotic time series used for training and testing of a LSTM model.

Figure 8 depicts the Log loss over epoch for LSTM Model. Because the LSTM model has not yet mastered the intricate and chaotic temporal relationships included in the fractional order time series, it is noted that the log loss is relatively significant during the early training phases. The explanation for this is because chaotic systems, particularly those with fractional orders, display intricate dynamics, such as sensitivity to initial conditions, extended memory, and non-linearity [37]. Because of its recurrent structure, the LSTM requires time to capture these features. The LSTM model is successfully learning from the training data while still generalizing to the validation set when the epoch increases and the training loss and validation loss decrease jointly. This indicates that the model is functioning successfully [38]. It is also observed that beyond 25 epochs, there is erratic behaviour of the training loss and validation loss which indicate that the model is struggling to learn from the fractional chaotic data [39, 40]. It suggests that the model may start to learn the training data and being sensitive to particular details that do not hold on the validation set. This can occur in chaotic systems where slight perturbances to the input can results in significant changes in the output [19].

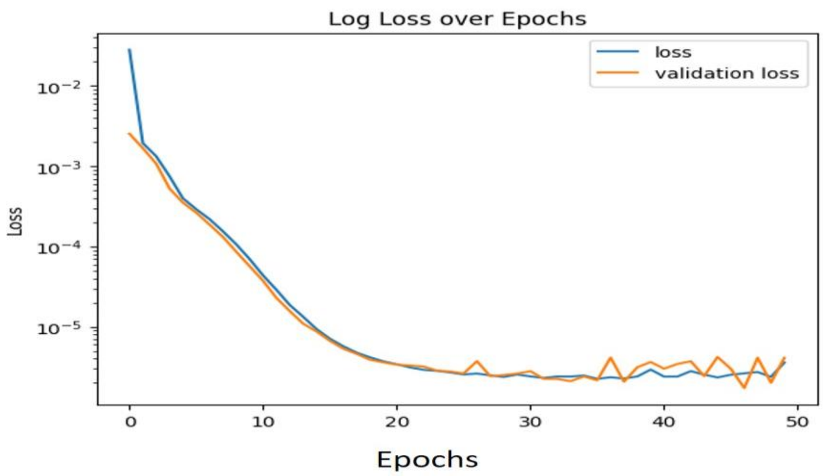

Figure 8: Log loss over epoch for LSTM Model in predicting fractional order of Lorenz chaotic system.

It is obvious from figure 9 that the training and testing losses drop at the same time, suggesting that the model is learning from the training data and generalizing well to the testing data. The model is capturing the chaotic temporal patterns without overfitting, as evidenced by the diminishing training and testing losses over epochs [39]. The RNN gradually enhances its predictions on the training and test sets by capturing temporal dependencies in the fractional chaotic time series. The series' chaotic character and the RNN's memory capacity are both effectively captured. This is the outcome of appropriate generalization and training. This is as a result of the proper training and generalization. The chaotic time series is complex, but the model manages to learn the long-term dependencies through its recurrent structure.

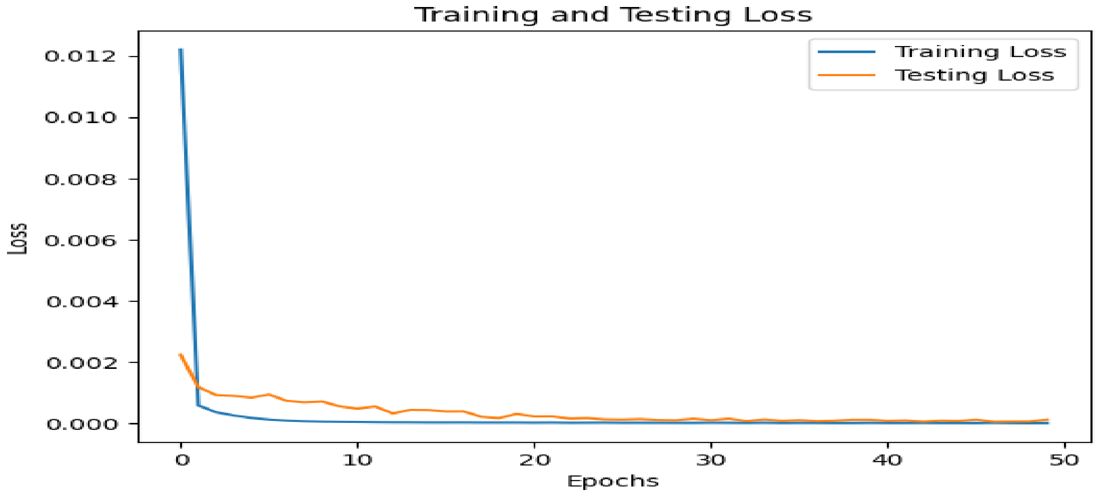

Figure 9: Training and testing losses over epoch for RNN Model

The result in Table 1 displayed both the traditional RNN and the LSTM models applied to fractional order chaotic data generated from the Lorenz system. It is obvious from figure 10 that train RMSE and test RMSE values for LSTM are low compared to the Train RMSE and test RMSE values for RNN. The lower test RMSE values for LSTM imply that they generalize better to unseen data (test data) than RNN. Because of its simpler architecture, RNN may overfit or perform poorly, whereas LSTM avoids these problems by selectively remembering and forgetting patterns. This implies that during the training stage, LSTM is more effective at identifying the fundamental dynamics and patterns of the fractional-order chaotic system. Unlike RNN, which has trouble with vanishing or exploding gradients over time, LSTM's architecture (with memory cells and gates) enables it to capture long-term dependencies in sequences more successfully [28, 41]. Because LSTM can manage complicated temporal dynamics and long-term dependencies, it is more suited for modeling fractional-order chaotic systems, as evidenced by its lower RMSE values in both the train and test sets when compared to RNN. The complex and nonlinear dynamics found in chaotic systems can be more accurately modeled by LSTMs due to their more advanced memory mechanism [28, 42]. Traditional RNNs, on the other hand, might not be able to capture the intricate attractors and peculiar behaviors that characterize chaotic systems because of their simpler recurring structure. It is obvious that the LSTM is more effective at capturing the chaotic attractors and fractional-order features of the system since it can minimize loss more efficiently than the RNN. Although modeling chaotic systems is highly difficult, the LSTM's architecture enables more accurate representation, as evidenced by the reduced loss values. In situations where ordinary RNNs would not be able to catch intricate time series patterns, LSTMs with lower RMSE values outperform typical RNNs in capturing chaotic behavior and storing long-term dependencies utilizing memory cells.

Table 1: Comparison of RNN and LSTM models' prediction performance on train and test data across several epochs.

Epoch	LSTM		RNN	
	Train RMSE	Test RMSE	Train RMSE	Test RMSE
25	0.073586	0.076948	0.188000	0.617000

50	0.055446	0.058238	0.180000	0.322000
75	0.050663	0.053167	0.131000	0.382000
100	0.052248	0.051021	0.121000	0.279000
125	0.044221	0.044684	0.094000	0.172000
150	0.042600	0.040065	0.056000	0.125000
200	0.038220	0.035155	0.032000	0.067000

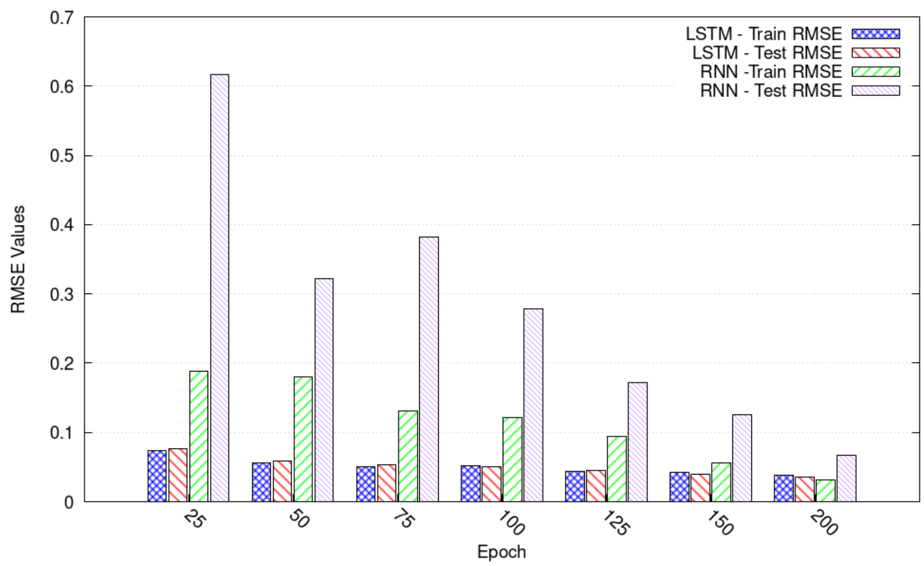

Figure 10: Bar chart representing RNN and LSTM models' prediction performance on train and test data across several epochs.

Conclusion

In this study, LSTM and a traditional RNN models are used in modeling and predicting the state of a fractional order chaotic system for one-step ahead. It is observed that train RMSE and test RMSE values for LSTM are low compared to the Train RMSE and test RMSE values for RNN. The lower test RMSE values for LSTM imply that they generalize better to unseen data (test data) than RNN. In addition, the LSTM's performs better in terms of forecasting future values in a fractional order chaotic time series due to its ability to model and learn long-term dependencies inherent in the fractional order chaotic system than the RNN, with its limited ability to retain relevant long-term information and cannot predict future states accurately. Learning about **fractional-order chaotic Lorenz systems** can indeed provide useful insights for real-world applications like **weather forecasting** by capturing memory effects (e.g., persistent weather patterns, climate cycles like El Niño), **stock market predictions, and traffic management**. Further studies can indeed be beneficial in investigating the effectiveness of predicting **fractional-order chaotic systems** of various other types of Recurrent Neural Networks (RNNs), compared to the traditional RNN and LSTM. Each variant of RNN brings unique strengths that could potentially improve the modeling of complex systems like fractional-order chaotic systems, which exhibit both nonlinearity and memory effects. By incorporating memory effects and long-term dependencies, these models can improve predictions and decision-making in these critical areas.

References

[1] Petráš, I. (2011). Fractional-order nonlinear systems: modeling, analysis and simulation. Springer.

[2] Machado, J. T. and Lopes, A. M. (2019). Fractional-order modeling of a diode. Commun Nonlinear Sci Numer Simul., 70:343–53.

[3] Elwakil, A. S. (2010). Fractional-order circuits and systems: an emerging interdisciplinary research area. IEEE Circ Syst Mag. 10(4):40–50.

[4] Almeida, L. B. (1994). The fractional Fourier transform and time-frequency representations. IEEE Trans Signal Process, 42(11):3084–91.

[5] Akhmet, M. and Fen, M. O. (2016). Replication of chaos in neural networks, economics and physics. Springer.

[6] Kiani-B, A., Fallahi, K., Pariz, N. and Leung H. (2009). A chaotic secure communication scheme using fractional chaotic systems based on an extended fractional Kalman filter. Commun Nonlinear Sci Numer Simul.14(3):863–79.

[7] Muthukumar, P., and Balasubramaniam, P. (2013). Feedback synchronization of the fractional order reverse butterfly-shaped chaotic system and its application to digital cryptography. Nonlinear Dyn., 74:1169–81.

[8] Chua, L. O., Komuro, M., Matsumoto, T. (1986). The double scroll family. IEEE Trans Circ Syst., 33(11):1072–118.

[9] Pham, V. T., Kingni, S. T., Volos, C., Jafari, S., and Kapitaniak, T. A. (2017). Simple three-dimensional fractional-order chaotic system without equilibrium: Dynamics, circuitry implementation, chaos control and synchronization. Int J Electron Commun (AEÜ), 78:220–7.

[10] Zhou, P. and Huang, K. (2014). A new 4-d non-equilibrium fractional-order chaotic system and its circuit implementation. Commun Nonlinear Sci Numer Simulat., 19:2005–11.

[11] Bourouba, B., Ladaci, S., and Chaabi, A. (2018). Reduced-order model approximation of fractional-order systems using differential evolution algorithm. J Control, Autom Electr Syst., 29:32–43.

[12] Baranowski, J., Pauluk, M. and Tutaj, A. (2017). Analog realization of fractional filters: Laguerre approximation approach. Int J Electron Commun (AEÜ), 81:1–11.

[13] Khanra, M., Pal, J. and Biswas, K. (2013). Rational approximation and analog realization of fractional order transfer function with multiple fractional powered terms. Asian J Control, 15:723–35.

[14] Machado, J. T. and Lopes, A. M. (2020). Multidimensional scaling locus of memristor and fractional order elements. J Adv Res, 25:147–157.

[15] Gulian, M., Raissi, M., Perdikaris, P. and Karniadakis, G. (2019). Machine Learning of Space-Fractional Differential Equations. SIAM J. Sci. Comput. [CrossRef]

[16] Wang, B., Liu, J., Alassafi, M.O., Alsaadi, F.E., Jahanshahi, H., Bekiros, S. (2022). Intelligent parameter identification and prediction of variable time fractional derivative and application in a symmetric chaotic financial system. Chaos Solitons Fractals, 154, 111590. [CrossRef]

[17] Woolley, J. W., Agarwal, P. K. and Baker, J. (2010). Modeling and prediction of chaotic systems with artificial neural networks. *International Journal for Numerical Methods in Fluids.* 63: 989–1004.

[18] Lorenz, E.N. (1963). Deterministic non periodic flow, J. Atmospheric Sci., 20, pp. 130-141.

[19] Lorenz, E.N. (1963). The Essence of Chaos, University of Washington Press, Seattle.

[20] Viswanath, D. and Şahutoğlu. S. (2010). Complex Singularities and the Lorenz Attractor, SIAM Review, Vol. 52, No. 2, pp. 294-314.

[21] Petráš. I. (2022). The fractional-order Lorenz-type systems: A review, Fractional Calculus and Applied Analysis, 25:362–377., https://doi.org/10.1007/s13540-022-00016-4

[22] Hochreiter, S. and Schmidhuber. J. (1997). "Long Short-term Memory". In: Neural computation 9, pp. 1735–80.

[23] Nicholson. C. (2019). A Beginner's Guide to LSTMs and Recurrent Neural Networks. https://skymind.ai/wiki/lstm.

[24] Chen, G. (2016). A Gentle Tutorial of Recurrent Neural Network with Error Backpropagation.

[25] Zhang, A., Lipton, Z. C., Li, M. and Smola, A. J. (2023). Dive into Deep Learning. 1st Editiohn, Cambridge University Pess. Uhttp://www.d2l.ai.

[26] **Chollet, F. (2018).** *Deep Learning with Python.* Manning Publications.

[27] **Zhang, Y., and Zhan, Y. (2018).** A Hybrid Approach of LSTM and CNN for Predicting Traffic Flow. *IEEE Access*, 6, 11180–11188.

[28] Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep Learning. MIT Press. http://www.deeplearningbook.org.

[29] **Sutskever, I., Vinyals, O., and Le, Q. V. (2014).** "Sequence to Sequence Learning with Neural Networks." *Advances in Neural Information Processing Systems (NIPS)*, 27.

[30] Lipton, Z.C., Berkowitz, J., Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning. arXiv preprint arXiv:1506.00019.https://arxiv.org/pdf/1506.00019.pdf

[31] Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6(02), 107–116.

[32] Van Houdt, G, Mosquer, C., and Nápoles, G. (2020). A Review on the Long Short-Term Memory Model. Artificial Intelligence Review. 53(1). DOI: 10.1007/s10462-020-09838-1.

[33] Thomas, A. (2017). Recurrent neural networks and LSTM tutorial in Python and Tensor Flow. Retrieved from http://adventuresinmachinelearning.com/recurrent-neural-networks-lstm-

tutorial-tensorflow/ [34] Olah. C. (2015). Understanding LSTM Networks. Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[35] Zaytar, M. A. and Amrani, C. E. (2016). Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks. *International Journal of Computer Applications (0975 - 8887). Volume 143 - No.11*.

[36] Monir, S. S. and Zhao, D. (2024). Veclstm: Trajectory Data Processing and Management for Activity Recognition Through LSTM Vectorization And Database Integration. (Computer Science) Machine Learning. DOI: 10.48550/Arxiv.2409.19258.

[37] Yang, C., Taralova, I. and Loiseau, J-J. (2021). Fractional Chaotic system solutions and their impact on chaotic behaviour. The 14th CHAOS 2021 International Conference, Athens (Turn to virtual), Greece. hal-03533065.

[38] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[39] Almaleh A. A. (2024). Novel Deep Learning Approach for Real-Time Critical Assessment in Smart Urban Infrastructure Systems. *Electronics (*MDPI**)**. 13(16):3286.

[40] Li, H., Rajbahadur, G. K., Lin, D., Bezemer, C-P., Jiang, Z. M. (2024). Keeping Deep Learning Models in Check: A History-Based Approach to Mitigate Overfitting. IEEE Access. 12: 70676 - 70689. DOI:10.1109/ACCESS.2024.3402543.

[41] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Phys. D Nonlinear Phenom, 404, 132306.

[42] Madondo, M. and Gibbons, T. (2018). Learning and modeling chaos using LSTM recurrent neural networks, in Proceedings of the Midwest Instruction and Computing Symposium.