Analysis of Long Short-Term Memory Models Forecasting Performance on Chaotic Systems with Changing Parameters

Roland Bolboacă[1] [0000−0002−4825−8786] (✉) and Piroska Haller[1] [0000−0002−5611−2429]

Electrical Engineering and Information Technology Department
George Emil Palade University of Medicine, Pharmacy, Science, and Technology of Targu Mures
38 Gheorghe Marinescu Street, Târgu Mureş, 540139, Mureş, Romania
{roland.bolboaca, piroska.haller}@umfst.ro

Abstract. Chaotic systems remain difficult to forecast due to their sensitivity to initial conditions and parameter variations. This work evaluates the long-term forecasting capability of Long Short-Term Memory (LSTM) models on one of the most well-known examples, namely the Lorenz system, across multiple operating regimes introduced by changes in parameters. In this paper, an evaluation framework is presented that systematically classifies the predictability of operating regimes, separating system behaviors into long, medium, and short-term predictable and unpredictable classes. Comprehensive experiments compare the performance of LSTM-based models with other recurrent approaches and transformer-based solutions over multiple forecast horizons, showing promising results for recurrent models in both prediction accuracy and training/testing times. To support reproducibility and extensions, all source code and datasets are made publicly available.

Keywords: Machine Learning · Chaotic Systems · Forecasting Performance · Long-short Term Memory.

1 Introduction

Chaotic systems are evaluated by a branch of mathematics and physics known as chaos theory. Chaos theory studies complex dynamical systems with unpredictable behaviors caused by their extreme sensitivity to initial conditions [1,2]. Today, chaotic systems are studied across a wide range of domains and areas of study, including, but not limited to, mathematics, engineering, education, computer science, and weather. These systems exhibit chaotic behaviors, phenomena also observed in many natural and artificial systems, including fluid dynamics, celestial mechanics, traffic flow, and financial markets.

Chaotic dynamical systems governed by deterministic laws exhibit long-term unpredictability of their state variables (e.g., trajectories), as small perturbations in the initial conditions can lead to unpredictable future responses [3]. The predictability limit in time depends on the tolerated uncertainty and the measurement errors of the state variables. The dynamics of such systems are also determined by a set of static parameters for a given configuration or instance. Different parameter values may yield entirely different behaviors, while this system appears to exhibit a wider variety of behaviors and operating regimes. Even decades after its introduction, the Lorenz system remains one of the most studied and well-known examples of chaotic dynamics. At the same time, small changes in the system parameters can suddenly change the periodic behavior of the system into a chaotic one and vice versa, increasing the difficulty of the system state predictions.

From a data analysis perspective, multi-regime behavior is defined as variations over time in statistical properties, correlations, spectral content, or dynamics within time series data collected from the system [4]. A time series consists of observations $\left[x_1, x_2, \ldots, x_n\right]$ collected at fixed or varying time intervals $\Delta t_i = t_i - t_{i-1}$. This example defines a univariate time series, typically obtained from a single sensor. Multivariate time series extend this concept to multiple related variables collected from the same system, often at the same time, and are further represented using standard vectorial bold notations as $\mathbf{X}$. Here, $\mathbf{X}$ is a feature vector consistng of all measured variables.

From a physical world perspective, multi-regimes illustrate changes in system behavior caused by internal or external perturbations, most frequently through parameter variations. Such changes may result from human intervention, external conditions, aging, cyberattacks, or natural degradation in both natural and engineered systems [5].

Modeling system behavior has long been of interest due to its practical use in applications such as monitoring and anomaly detection, predictive maintenance, remaining useful life estimation, and weather forecasting. Resulting models are typically used in two modes: prediction and simulation. Prediction uses observed data to estimate future system states, while simulation propagates the model using its own previous predictions.

When only measured time series data describing system dynamics are available, accurate models can still be built using different approaches. One option is the use of machine learning to model system behavior and predict future states over short and long horizons. This study assumes that system parameters may change over time, causing shifts between operating regimes. Since these parameters are not directly observed, it is important to examine how such variations affect the model's overall predictability.

Statistical methods, ARX-based models, recurrent neural networks (RNNs), and more recently, transformers are widely used for modeling and forecasting linear, nonlinear, and chaotic systems. Among these approaches, LSTM models have shown promising performance while addressing limitations of basic RNNs [6]. Although transformers [7] have recently attracted interest in the context of long-term forecasting, they have high computational complexity and large data requirements, especially for resource-constrained devices.

As highlighted above, there is a clear need to evaluate which states can be reliably forecast over short and long time horizons using LSTM models and to identify the forecast limits where predictions remain meaningful. The following sections analyze the Lorenz system, the role of the parameter ρ, and LSTM model predictability limits. Additionally, this paper offers a general benchmark of several forecasting approaches for chaotic systems and offers the following contributions.

- An analysis of the Lorenz system across a wide range of parameter values and random initial conditions.
- An evaluation of short- and long-term multivariate forecasting with LSTM models for horizons from 2 to 256 steps.
- Identification of predictable and non-predictable operating regimes over extended time horizons.
- Comparison with other recurrent neural networks and transformer-based forecasting solutions.
- Publicly available Python code [1] and datasets [2].

The remainder of the paper is organized as follows. Section 2 presents relevant related studies. Next, section 3 describes the proposed evaluation framework. This is followed by experimental details that are given in Section 4. The results of this study are presented and briefly discussed in Section 5 and the paper concludes in Section 6.

2 Related Studies

The Lorenz system and chaotic systems in general have been extensively studied for more than two decades, with numerous approaches proposed for parameter analysis and forecasting using both traditional and machine learning methods. This section presents recent and relevant studies related to the topics addressed in this article, including long- and short-term prediction of the Lorenz system [8], parameter sensitivity analysis [9], and data-driven modeling of chaotic dynamics [10].

In a recent review on chaotic time series prediction, Kui and Lai [11] pointed out the high computational cost of deep learning models such as transformers, which have quadratic complexity, as well as their limitations in long-term forecasting, where these methods often require large datasets or high computational resources. Similarly, Hans Krupakar [12] showed that prediction errors generally increase with forecast horizon for the state-of-the-art models, while approaches using gating mechanisms and multiscale attention seem to maintain stable performance.

Although long-term predictions of the Lorenz and other chaotic systems have been analyzed by multiple scholars, Shen et al. [13] discussed the real-life predictability of the atmosphere to be two weeks, even with state-of-the-art machine learning approaches. Similarly, Mikhaeil et al. [14] analyzed the difficulty of learning chaotic dynamics with RNNs where they demonstrated that if the modeled system's dynamics are chaotic, the gradients will always explode during training. The authors recommended the use of more advanced models, including LSTMs, and also recommended the use of annealing procedures to improve the short-term predictions first before moving up to long-term forecasting.

In a related study, Teixeira et al. [15] showed the complexity of the Lorenz system regarding the time-step of the simulation, where trajectories with small but different time-steps decouple from each other after a finite amount of time. The authors also

[1] https://github.com/rolandbolboaca/Multi_state_eval_chaos
[2] https://doi.org/10.7910/DVN/MBGA1K

showed sensitivity to variations in the ρ parameter from a simulation perspective. However, their work is limited to small values for ρ.

The behavior of the Lorenz system, and chaotic systems in general, has been widely studied regarding changes in the control parameter, revealing operating regimes ranging from periodic trajectories to fully chaotic behavior [16]. In terms of predictability, Peter Chu [17] identified two main factors: sensitivity to initial conditions and sensitivity to variations in the parameter ρ (denoted as r in his work). His results indicated that prediction error does not vary linearly with ρ, but instead increases and decreases across different values. However, the analysis was limited to the range $\rho \in [24, 30]$.

In a similar direction, Song and Cattani [18] analyzed the transition of the system towards chaos regarding the ρ parameter and proposed a fast detection method for weak singularities using the bisection algorithm. The authors analyzed multiple trajectories using Lyapunov exponents and identified an important threshold of ρ between 24.05 and 24.10 that separates periodic and chaotic states, showing clear chaotic patterns for ρ values above this threshold.

In recent years, machine learning approaches have been extensively utilized for chaotic systems modeling. For example, Zhang and Xiao [19] investigated the application of standard RNN architectures to model time series originating from chaotic systems. The authors proved that such architectures can perform both one-step and multi-step ahead accurate forecasts. This work, however, only analyzed short to medium term forecasting, with up to 20 steps ahead and without parameter and state change evaluations.

Dubois et al. [20] proposed an LSTM-based modeling approach for the Lorenz system using a deep learning architecture with LSTM and dense layers. Their model achieved accurate predictions compared to simpler methods such as the Kalman filter and used all system variables as inputs to predict future states. The authors also included first and second derivatives as additional inputs and showed that the model can predict multiple steps ahead. However, their analysis was limited to a single operating regime with $\rho = 28$. In addition, long-term forecasts were corrected using a dense network that assimilates online measurements, meaning predictions were updated with real data, which conflicts with the goal of long-term forecasting.

The prediction performance of LSTMs has also been investigated by Adeniji et al. [21] for one-step ahead prediction. Their work highlighted generalization performance improvements of LSTMs compared to traditional RNNs on fractional-order chaotic Lorenz systems. Prater et al. [22] investigated the long-term prediction performance of LSTMs on time-series data. The authors reached similar conclusions as the ones obtained in this paper, that is, that the LSTM models learned to predict the mean value but not the distinguished patterns (frequency or seasonality) for long-term forecasting. Similar results were obtained in this article for certain unpredictable ρ values.

The behavior of the Lorenz system regarding the parameter ρ and its short- and long-term predictability has been widely studied. However, most existing work focuses on a limited range of parameter values, single-step predictions, or does not examine performance across different operating regimes. This paper addresses these limitations by introducing an evaluation framework that analyzes how parameter variations affect forecasting performance and applies unsupervised learning to classify predictable and unpredictable regimes. The classification identifies regimes that remain predictable over long horizons, regimes where accuracy degrades rapidly beyond short-term forecasts, and regimes that remain predictable over short and medium horizons before deteriorating for longer horizons.

3 Proposed Evaluation Framework

The proposed evaluation framework is designed as a tool that researchers and practitioners can use for assessing long-term LSTM forecasting on data generated by chaotic systems. While short-term performance has been studied [6], this paper extends that analysis by examining how forecasting performance varies with system parameters and prediction horizons. Performance is evaluated using Mean Squared Error (MSE), Shape Based Distance (SBD), and Energy Distance (ED) [23].

Based on these results, system operating regimes are classified as predictable or non-predictable. The framework makes the distinction between regimes that remain predictable over long horizons and those that fail even at short horizons and identifies the prediction horizon where performance rapidly degrades. As noted by Peter Chu [17], predictability in chaotic systems can be studied through changes in initial conditions or system parameters. This work focuses on the second case, analyzing prediction performance across different operating regimes caused by variations in a system parameter.

LSTM models [24] were designed to overcome the problem of vanishing and exploding gradients of standard RNNs. They remain among the most effective architectures for modeling sequential data, capable of learning both short- and long-term dependencies [6]. An LSTM layer contains memory cells controlled by four gates: input, candidate, forget, and output. Each cell includes two recurrent connections that represent short- and long-term memory. Gates regulate how information is stored, forgotten, and used as output. All experiments use the standard LSTM model as originally introduced in [24].

3.1 Evaluation Methodology

In the context of forecast evaluation, the LSTM models are utilized in a variation of the sequence-to-sequence approach. In a standard sequence-to-sequence approach, for the given set of variables $\mathbf{X}$, the model will output $\hat{\mathbf{Y}}$, with n values for each variable. In the context of LSTMs, the sequence length denotes the number of samples for each variable that are utilized as input before making predictions. In recent forecast-based terminology, this is similar to a look-back window [25]. The observed outputs (e.g., ground truth values) are further denoted as $\mathbf{Y}$.

Due to the limitations of the LSTM model in sequence-to-sequence modeling, to predict n future values, the input sequence length must be greater than or equal to n. Similarly to [26], the sequence length is selected as the highest value of desired forecast horizons τ, with overlapping predictions for smaller forecast horizons $\tau < n$. For a given value of n and a prediction horizon of τ, where $\tau \leq n$, the model will take as input, for each variable, the n past values $[x_{t-n}, x_{t-n+1}, ..., x_t]$ and predict n future values $[y_{t-n+\tau}, y_{t-n+\tau+1}, ..., y_t, ..., y_{t+\tau+1}]$. Of the n predicted values, the actual τ future forecast values are $\left[y_{t+1}, y_{t+2}, \ldots, y_{t+\tau+1}\right]$ without input overlap, for each variable Y_i. Although the model also outputs the values $\left[y_{t-n+\tau}, y_{t-n+\tau+1}, \ldots, y_{t-1}, y_t\right]$, these values are discarded for now. After a prediction of n values, from which the τ values are kept, the input sequence is moved forward τ values.

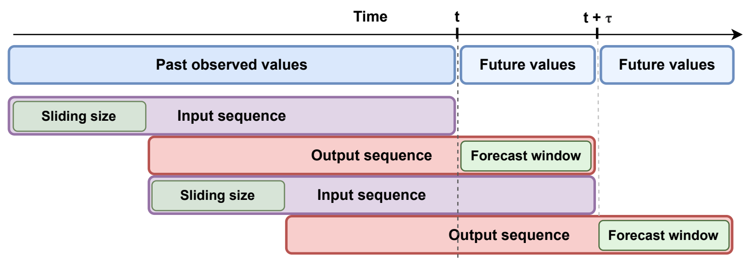

Fig. 1. LSTM sliding window sequence to sequence long term forecast methodology overview.

The forecast methodology is detailed in Figure 1 where two forecasts are shown, from time t to $t + \tau$, and from $t + \tau$ to $t + 2 \cdot \tau$. As noted above and shown in the figure, the sliding size is equal to the forecast horizon τ. During the experiments, each model is trained and tested using the same sequence length for a given scenario. All LSTM models operate in a stateful mode, meaning that the hidden and cell states are maintained across sequences and batches rather than being reset after each one. The effect of sequence length on performance was previously examined by others as well, for example in [27] and [6].

3.2 Evaluation Metrics

As noted above, the Lorenz system exhibits a wide range of behaviors, from periodic to chaotic, depending on the value of the parameter ρ, with similar but not identical repeating patterns for certain ρ values. Consequently, three different metrics are utilized to evaluate prediction performance, namely MSE, SBD, and ED. MSE compares predictions and observations point-by-point. ED measures differences between the distributions of predicted and observed values. SBD evaluates similarity across all possible temporal shifts, revealing how well the model reproduces the shape of the trajectory even with delays [23].

To create the forecasting classes, high-dimensional data points are generated and clustered using an unsupervised algorithm. For each high-dimensional data point, each dimension represents the i-th metric function value $\mathcal{E}_i^{\rho,\tau}(\mathbf{Y}, \hat{\mathbf{Y}})$ computed for each ρ_j with $j = 1, \ldots, s$ and τ_l with $l = 1, \ldots, r$. Here, s denotes the number of analyzed operating regimes, e.g., the number of values for ρ, and r denotes the number of forecast horizons evaluated. For a given ρ_j and τ_l, the features are $[\mathcal{E}_1^{\rho_j \tau_l}, \mathcal{E}_2^{\rho_j,\tau_l}, \ldots, \mathcal{E}_N^{\rho_j,\tau_l}]$, where N denotes the number of metrics utilized. For readability and without loss of generality, the parentheses are dropped for the metric functions $\mathcal{E}_i^{\rho,\tau}$. The resulting points are stored in a matrix, further denoted by $\mathbf{M}$.

3.3 Clustering Approach

To create clusters, the KMeans [28] algorithm is utilized, which produces K clusters, where K is selected a priori. The KMeans algorithm takes as input a list of multidimensional data points D, where each point is represented by a row of the matrix $\mathbf{M}$, the number of clusters K, and optionally the number of iterations. The algorithm will output the cluster assignments $\mathbf{C} = \{C_1, C_2, ..., C_K\}$ and the centroids for each cluster, denoted $\mu_1, \mu_2, ..., \mu_K$.

Clustering is performed in three steps, of which the last two steps are repeated iteratively. During the first step, which is the initialization of the algorithm, K random centroids are chosen as existing data points. During the second step, at each iteration, each data point is assigned to the closest centroid, utilizing the Euclidean distance between the point and all centroids. The third step involves updating the centroid with new positions. For each of the K centroids, the new positions are calculated as the average value over all points assigned to the selected centroid. The second and third steps are repeated until the centroid positions do not change significantly or until the number of iterations is reached.

The optimal number of clusters K is selected using four clustering evaluation metrics, namely the Silhouette Score, Calinski–Harabasz Index, Davies–Bouldin Index, and Inertia (e.g., elbow method). Each metric evaluates clustering quality by varying K and computing the Euclidean distance. The elbow method identifies the point where the sum of squared errors begins to stabilize, the Silhouette Score assesses cluster cohesion and separation, the Calinski–Harabasz Index measures the ratio of between-cluster to within-cluster variance, and the Davies–Bouldin Index quantifies inter-cluster similarity. The final choice of K is made using a visual inspection of these metrics plotted against the number of clusters.

4 Experimental Setup and Evaluations

The experiments were conducted on a Linux-based desktop system running kernel version 5.15.0. The hardware setup included an AMD Ryzen 5 7600X CPU at 4.7 GHz, 32 GB of DDR5 RAM, and an RTX 4070 SUPER GPU with 12 GB of GDDR6X memory. The implementations were designed in MATLAB and Python. The transformer-based models were implemented using the DARTS library [29], which provides a range of tools for time series forecasting and anomaly detection.

4.1 Lorenz System Datasets

The Lorenz system represents a simplified model of atmospheric convention and the unpredictable behavior of the weather. In the original paper [30], the author introduced a theory to model the dynamics of a fluid warmed from below and cooled from above. In the field of chaos theory, this system remains a paradigm, as it captures a wide variety of features of chaotic dynamics. Close initial conditions lead to very different trajectories, making the Lorenz system a deterministic chaotic dynamical system [31].

The Lorenz system equations are shown in Equation 1. This system is defined by the following parameters: σ which is the Prandtl number, ρ which represents the Rayleigh number, and β which is a geometric factor [31]. As originally described, the values of the parameters σ, ρ, and β are usually set, respectively, to 10, 28 and 8/3. Nonetheless, the Lorenz system exhibits a wide range of dynamical behaviors and operating regimes depending on the ρ parameter values, including steady states (fixed points), bistability, periodic oscillations, period-doubling cascades, quasiperiodicity, intermittency, and fully developed (pure) chaos [16,3].

$$\begin{cases} \dfrac{dx}{dt} = \sigma(y - x) + \xi_x(t), \\ \dfrac{dy}{dt} = x(\rho - z) - y + \xi_y(t), \\ \dfrac{dz}{dt} = xy - \beta z + \xi_z(t). \end{cases} \tag{1}$$

The datasets used for the experimental evaluation were generated using MATLAB, i.e. utilizing the ODE45 solver with multistep integration. For each respective scenario, the simulations start with random initial conditions in the range [0, 1]. The time span for each simulation ranges from 1 to 500 with a time step of 0.05. For each value ρ in the range [5, 225], a number of 10 simulations were generated that encompass 19,810 data points with no transient state removal. Two datasets were created, one for training and one for testing, each containing 891,450 data points, generated with random initial conditions. At each simulation step, independent Gaussian noise terms $\xi_x(t)$, $\xi_y(t)$, and $\xi_z(t)$ were added to each state variable, where $\xi_i(t) \sim \mathcal{N}(0, 1)$, similar to the approaches in [6,2]. For each value of $\rho \in [5, 225]$ and subsequently for each value of $\tau \in [2, 256]$, a new LSTM model was created, trained with 10 simulations worth of data, and tested on test sets that included 10 new simulations, all simulations generated with random initial conditions for the three variables.

4.2 Parameters, Hyperparameters and Clusters

LSTM performance is strongly affected by hyperparameter selection. Except for the input sequence length, which was chosen to be larger than the forecast horizon, the remaining hyperparameters were selected through a combination of grid search and empirical evaluations. The final configuration included two hidden layers with 32 and 16 units, 300 training epochs, an initial learning rate of 0.01, a decay rate of 0.99 every 100 epochs, a batch size of 64, an input length of 256, and the Adam optimizer with MSE loss.

Based on the four clustering metrics described in Section 3.3, the number of clusters was set to $K = 3$ from a tested range of [2, 15]. Additional experimentation with larger numbers of K did not result in noticeable improvements. Forecast horizons τ were evaluated between 2 and 256 steps, while the parameter ρ was varied from 5 to 225. In total, 45 values of ρ were tested across 85 forecast horizons.

5 Evaluation Results

To better highlight the working context, the importance of this study and to help the reader better understand the issues addressed, the forecasting performance of an LSTM

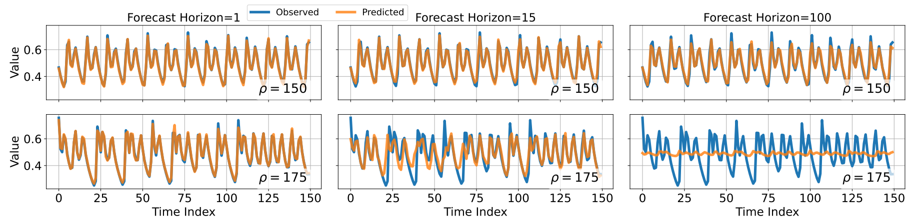

Fig. 2. Illustration of the LSTM long- and short-term forecasting performance for two operating regimes of the Lorenz system with parameter $\rho = 150$ and $\rho = 175$

model on the Lorenz system for two selected values of ρ is shown in Figure 2. For an accurate and comparable visualization, all values were scaled in the range [0, 1]. The figure illustrates that the LSTM provides accurate one-step-ahead predictions for both parameter values. The figure also shows forecasts at 15 and 100 steps ahead. Results indicate that some regimes remain predictable across multiple horizons, while others show noticeable degradation for long-term forecasts but still perform well on shorter forecast horizons.

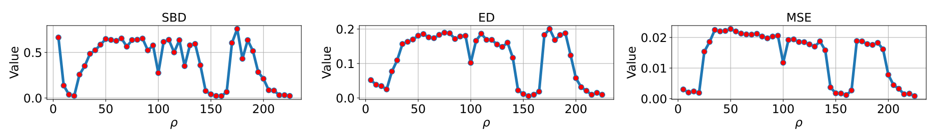

Fig. 3. Illustration of the SBD, ED, and MSE over all ρ values with a forecast horizon of 41.

The forecast capability was evaluated using previously described metrics, and the values were visually represented. An example of the selected metrics is shown in Figure 3 for $\tau = 41$, where a visual analysis of the results for all regimes showed a clear degradation in performance at $\rho = 175$. This was also illustrated in Figure 2.

5.1 System Operating Regime Forecasting Results

The initial evaluation showed that for all tested ρ values, prediction errors increased slowly up to a forecast horizon of $\tau = 15$, with minimum values of 0.00005 (MSE), 0.0014 (SBD), and 0.0006 (ED). In all experiments, the maximum values were 0.031 (MSE), 0.272 (ED), and 0.908 (SBD). While the results revealed a clear separation of ρ regions, some regimes yielded contradictory decisions across the three metrics.

Recall that the matrix **M** was constructed using all metric values based on the values of ρ and τ, followed by the application of the KMeans algorithm with $K = 3$. The clustering results, which include all ρ values, forecast horizons τ, and the three metrics utilized, are shown in Figure 4. To avoid scale issues between metrics, the calculated values for each metric were normalized in the range [0,1] before clustering. The results

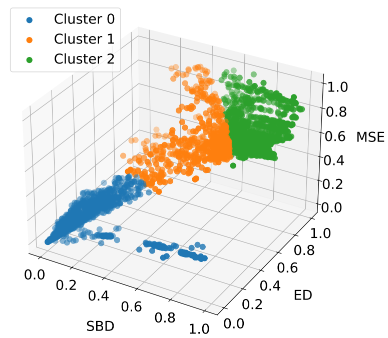

Fig. 4. Clustering results based on forecast performance w.r.t. ρ and τ

highlight that the clusters encompass points with progressively increasing values for all metrics. An exception exists where the MSE and ED values remain constant, while only the SBD values increase. After a close analysis, it was revealed that all these points belong to the same trajectory for $\rho = 5$.

Overall, the cluster distribution was relatively balanced, indicating the degradation of prediction performance as the forecast horizon increased. The high concentration of points in Cluster 0 indicated that for many combinations of ρ and τ, particularly for small τ, system trajectories remain predictable across all metrics. In contrast, Cluster 2 contains many points that were associated with longer horizons, revealing system regimes where prediction performance deteriorated significantly. Cluster 1 represents some intermediate cases, where multiple ρ values indicate average prediction performance over all evaluated metrics.

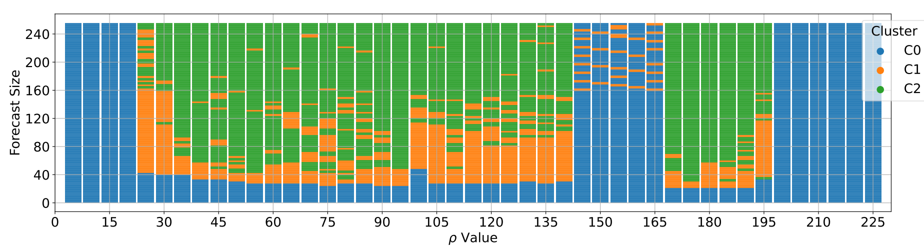

Fig. 5. Illustration of the clustering results by ρ and forecast size

Figure 5 shows the cluster composition based on degrees of membership, where for each value of ρ, the corresponding forecast horizons τ are assigned to clusters. The first cluster, with the lowest values across all metrics, includes points from all ρ values, and for some ρ cases all horizons fall entirely within this cluster. Analysis of the second and third clusters shows that several ρ values appear in both, with a higher concentration in the third cluster. These results indicate that for some ρ values, predictions remain accurate across multiple horizons, while for others performance degrades at specific forecast horizons. These results hold similarities to other forecasting results of chaos indicators, for example [2].

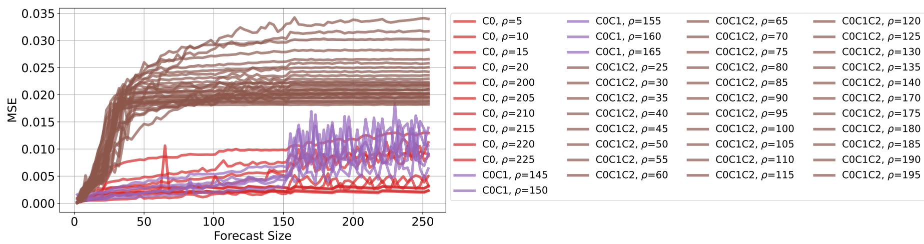

Fig. 6. Illustration of the MSE computed for all ρ values and for each forecast horizon τ across the three defined classes. The legend illustrates the specific clusters to which each ρ has been assigned.

Based on previous observations of the results, predictability classes were defined as follows: (I) Class 1: the LSTM correctly predicts on all forecast horizons; (II) Class 2: the LSTM predicts accurately on small and medium horizons, while performance deteriorates for some long-term forecasting; (III) Class 3: accurately predicted for small horizons, and performance deteriorates afterward. The results are shown in Figure 6 where the three classes are indicated with different colors (Red - first class, Purple - second class, Brown - third class). The figure also shows the evolution of the forecast errors measured using the MSE for all ρ values with all τ horizons. The legend attached to the figure shows, for each ρ, the membership to the three clusters.

As observed, ρ values assigned to only the first cluster exhibit smooth transitions in MSE across all τ values, while ρ values assigned to multiple clusters show higher slopes, indicating faster performance deterioration with increasing τ. These results are consistent with those in Figure 6. Having data points in the second cluster, where all metrics have average values, indicated that the performance for some horizons was gradually worsening. As observed in Figure 6, there was no single ρ value for which all τ values were assigned to the second cluster alone. This aspect revealed that there were no smoother transitions w.r.t. the forecast horizons, where the prediction performance slowly deteriorates over a larger number of horizons.

To analyze how performance evolves with increasing forecast horizon τ for each value of ρ, additional experiments were performed. Following a similar approach to [32], a line was fitted to the log of the MSE values as a function of τ within each cluster, and the slope was computed for each ρ value. The results, grouped by class and cluster, are presented in Table 1. The third class shows a clear exponential growth of prediction error with forecast horizon, confirming the chaotic nature of the Lorenz system for specific ρ values. In contrast, the first class exhibits lower average slopes, indicating a smoother increase in prediction error across horizons. For the third class, slopes decrease within the second cluster and also show error saturation owed to the bounded nature of the Lorenz system.

Table 1. Average and standard deviation of the polyline slopes across assigned classes and clusters.

Class	Cluster 0		Cluster 1		Cluster 2	
	Mean	Std.	Mean	Std.	Mean	Std.
Class 1	0.00558	0.00341	-	-	-	-
Class 2	0.00942	0.00153	0.00005	0.00183	-	-
Class 3	0.14947	0.03114	0.00275	0.00390	0.00021	0.00015

5.2 Comparison Results

The LSTM model was compared with other recurrent approaches, including GRU and standard RNN models, as well as two transformer-based methods, which include a standard transformer and the Temporal Fusion Transformer (TFT) [33]. The comparisons were performed using data from all three predictability classes with ρ values of 5, 200, and 220 for long term, 145, 150, and 165, for medium term, and 25, 80, and 175, short term. Additionally, with forecast horizon τ with values of 5, 50, 100, and 250.

All recurrent-based models were configured using the same hyperparameters as described above, while the transformers used the following values (obtained using hyperparameter tunning): direct forecasting, look-back window 256, batch size 64,epochs 200, model dimension 64, attention heads 4, encoder-decoder layers 2, dropout rate 0.1, learning rate 0.001, hidden size (TFT) 64, and stride $= \tau$.

Table 2 summarizes the average performance over all ρ values for each forecast horizon τ. As observed, performance declined as the forecast horizon increased for all models. The LSTM achieved the best MSE results for horizons of 50, 100, and 150, followed by the GRU, RNN, and the transformer-based models. For the longest horizon of 250, the GRU performed best, followed by the LSTM and RNN, with small differences in MSE but larger differences in SBD. Among the recurrent models, the RNN showed the lowest performance, particularly for short and medium horizons. In summary, LSTM and GRU obtained similar results, which is consistent with previous studies showing comparable performance between these models depending on the application context [34,35].

Overall, the standard transformer model achieved slightly worse results than recurrent-based models, with a few exceptions. For a forecast horizon of 5, transformer-based methods performed best on all metrics. The standard transformer achieved an MSE of 0.0007, the TFT model 0.0006, and the LSTM 0.0009. In terms of ED, the TFT outperformed the other models across all horizons, followed by the LSTM and the standard transformer. For SBD, results were similar around a horizon of 50 steps, while at 5 steps the TFT achieved 0.021 SBD, compared to 0.034 SBD for the LSTM and 0.023 SBD for the standard transformer. For horizons greater than 5, LSTM, GRU, and RNN models showed better MSE and SBD performance. In general, transformers performed worse for larger horizons in this scenario. While the TFT achieved the best results for short horizons, no single model consistently outperformed the others on all metrics and forecast horizons.

Table 2. Average performance metrics across forecast horizons for RNN-based and transformer-based models. Best value per row shown in bold.

Metric	τ	RNN	GRU	LSTM	Transf.	TFT
SBD	5	0.1920	0.0600	0.0340	0.0239	**0.0212**
	50	0.3297	0.3224	**0.3148**	0.3306	0.4148
	100	**0.3390**	0.3622	0.3511	0.5443	0.9259
	250	0.4327	**0.3729**	0.4021	0.6938	0.9498
ED	5	0.0640	0.0163	0.0083	0.0090	**0.0046**
	50	0.0839	0.0719	0.0760	0.0596	**0.0094**
	100	0.0942	0.0924	0.0916	0.1051	**0.0086**
	250	0.1266	0.1098	0.1287	0.1424	**0.0091**
MSE	5	0.0169	0.0015	0.0009	0.0007	**0.0006**
	50	0.0097	**0.0087**	0.0091	0.0348	0.0413
	100	0.0112	0.0112	**0.0108**	0.0255	0.0370
	250	0.0141	**0.0133**	0.0140	0.0204	0.0384
Parameters		**5,963**	19,011	23,387	414,000	218,000
Train Time [s]		286	105	**102**	128	369
Test Time [s]		0.81	0.36	0.37	**0.33**	1.34

The TFT model comprised 218,000 parameters, fewer than the standard transformer (414,000), while recurrent models ranged from 6,000 to 24,000 parameters. Training and testing times were also higher for the TFT model: LSTM and GRU required 102–105 seconds on 19,810 data points, compared to 369 seconds for the TFT. The longest testing times were observed for the TFT, followed by the standard RNN, despite all models being trained and evaluated on a GPU with optimized transformer implementations using Dart's Accelerator with Pytorch Lightning Trainer.

These results are consistent with previous studies showing that transformers can perform better, similarly, or worse than simpler recurrent architectures depending on the application context. While highlighting current limitations of LSTM architectures, the results also show that their lower computational cost offers significant potential for further performance improvements through architectural and training enhancements [36,37].

Additionally, Figure 7 shows the phase-space trajectory visualization projection for the three models, which further highlights the prediction performance differences. As also shown above, transformer-based models perform better for short-term forecasting. However, for longer horizons (e.g., 100 steps), transformer predictions diverge significantly from the observed trajectories.

6 Conclusions and Future Work

This paper introduced an evaluation framework for the investigation of the long- and short-term forecasting performance of LSTM models on chaotic systems, with a focus

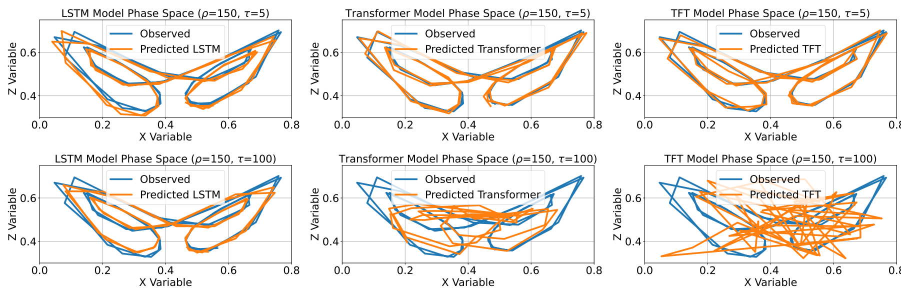

Fig. 7. Projected phase-space trajectories (50 points) of the Lorenz system using the x–z plane for $\rho = 150$.

on the Lorenz system with parameter variations. A wide range of operating regimes was examined by varying the parameter ρ between 5 and 225, evaluating 45 parameter values over 85 forecast horizons from one to 256 steps ahead. The proposed framework includes an unsupervised classification of operating regimes into three categories: regimes that remain predictable over both short and long horizons, regimes that are predictable only in the short term, and regimes that are predictable in the short and medium term and slowly degrade for longer horizons. The results indicate that the Lorenz system can be accurately predicted across all analyzed regimes for short horizons, with LSTM performance maintained for up to 15 steps ahead.

The study also compared LSTM performance with additional recurrent architectures and transformer-based models while also highlighting the strengths and limitations of LSTMs in the context of long-term forecasting of chaotic systems. All datasets and source code were made publicly available to support reproducibility and future extensions of the framework. This work provides the basis for our future research that will focus on long-term forecasting improvements for nonstationary chaotic systems and enhancements for regimes identified as unpredictable in the long term.

Data and Code Availability

The data that supports the findings of this study are openly available at Harvard Dataverse at https://doi.org/10.7910/DVN/MBGA1K. The complete source code that generated the results and figures from this article is provided as a Jupyter notebook under the MIT License on GitHub at https://github.com/rolandbolboaca/Multi_state_eval_chaos.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. S. Yu, W. Chen, and H. V. Poor, "Real-time monitoring of chaotic systems with known dynamical equations," *IEEE Transactions on Signal Processing*, vol. 72, pp. 1251–1268, 2024.

2. R. Bolboacă and P. Haller, “An empirical study on the limits of black box forecasting in chaotic systems with changing parameters,” *Chaos, Solitons & Fractals*, vol. 211, p. 118779, 2026.
3. C. Sparrow, *The Lorenz equations: bifurcations, chaos, and strange attractors*, vol. 41. Springer Science & Business Media, 2012.
4. V. R. Vavilthota, R. Ramanathan, and S. N. Aakur, “Capturing temporal components for time series classification,” in *International Conference on Pattern Recognition*, pp. 215–230, Springer, 2024.
5. G. J. Odom, K. B. Newhart, T. Y. Cath, and A. S. Hering, “Multistate multivariate statistical process control,” *Applied Stochastic Models in Business and Industry*, vol. 34, no. 6, pp. 880–892, 2018.
6. R. Bolboacă and P. Haller, “Enhanced long short-term memory architectures for chaotic systems modeling: An extensive study on the lorenz system,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 34, p. 123152, 12 2024.
7. P. Chen, Y. Zhang, Y. Cheng, Y. Shu, Y. Wang, Q. Wen, B. Yang, and C. Guo, “Multi-scale transformers with adaptive pathways for time series forecasting,” in *International Conference on Learning Representations*, 2024.
8. T. Wu, X. Gao, F. An, X. Sun, H. An, Z. Su, S. Gupta, J. Gao, and J. Kurths, “Predicting multiple observations in complex systems through low-dimensional embeddings,” *Nature Communications*, vol. 15, p. 2242, Mar. 2024.
9. D. Patel and E. Ott, “Using machine learning to anticipate tipping points and extrapolate to post-tipping dynamics of non-stationary dynamical systems,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 33, p. 023143, 02 2023.
10. X. Wang, J. Feng, Y. Xu, and J. Kurths, “Deep learning-based state prediction of the lorenz system with control parameters,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 34, p. 033108, 03 2024.
11. Y. Kui and Q. Lai, “Deep Learning Methods for Chaotic Time Series Prediction,” *Knowledge-Based Systems*, p. 114441, Sept. 2025.
12. H. Krupakar and K. V. A, “A Review of the Long Horizon Forecasting Problem in Time Series Analysis,” June 2025.
13. B.-W. Shen, R. A. Pielke, X. Zeng, and X. Zeng, “Lorenz's View on the Predictability Limit of the Atmosphere,” *Encyclopedia*, vol. 3, pp. 887–899, Sept. 2023.
14. J. Mikhaeil, Z. Monfared, and D. Durstewitz, “On the difficulty of learning chaotic dynamics with rnns,” in *Advances in Neural Information Processing Systems* (S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, eds.), vol. 35, pp. 11297–11312, Curran Associates, Inc., 2022.
15. J. Teixeira, C. A. Reynolds, and K. Judd, “Time Step Sensitivity of Nonlinear Atmospheric Models: Numerical Convergence, Truncation Error Growth, and Ensemble Design,” *Journal of the Atmospheric Sciences*, vol. 64, pp. 175–189, Jan. 2007.
16. D. Patel, D. Canaday, M. Girvan, A. Pomerance, and E. Ott, “Using machine learning to predict statistical properties of non-stationary dynamical processes: System climate,regime transitions, and the effect of stochasticity,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, p. 033149, Mar. 2021.
17. P. C. Chu, “Two Kinds of Predictability in the Lorenz System,” *Journal of the Atmospheric Sciences*, May 1999.
18. T. Song and C. Cattani, “Fast Detection of Weak Singularities in a Chaotic Signal Using Lorenz System and the Bisection Algorithm,” *Mathematical Problems in Engineering*, vol. 2012, no. 1, p. 102848, 2012.
19. J.-S. Zhang and X.-C. Xiao, “Predicting chaotic time series using recurrent neural network,” *Chinese Physics Letters*, vol. 17, no. 2, p. 88, 2000.

20. P. Dubois, T. Gomez, L. Planckaert, and L. Perret, "Data-driven predictions of the lorenz system," *Physica D: Nonlinear Phenomena*, vol. 408, p. 132495, 2020.
21. A. E. Adeniji, K. S. Oyeleke, K. S. Ojo, and A. M. Lasisi, "Modeling And Prediction Of Fractional-Order Chaotic Lorenz System Using RNN And LSTM Networks," *The Journals of the Nigerian Association of Mathematical Physics*, vol. 69, pp. 139–152, Mar. 2025.
22. R. Prater, T. Hanne, and R. Dornberger, "Generalized Performance of LSTM in Time-Series Forecasting," *Applied Artificial Intelligence*, vol. 38, p. 2377510, Dec. 2024.
23. J. Paparrizos, H. Li, F. Yang, K. Wu, J. E. d'Hondt, and O. Papapetrou, "A survey on time-series distance measures," *arXiv preprint arXiv:2412.20574*, 2024.
24. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
25. Y. Nie, "A time series is worth 64words: Long-term forecasting with transformers," *arXiv preprint arXiv:2211.14730*, 2022.
26. A. Drouin, É. Marcotte, and N. Chapados, "TACTiS: Transformer-Attentional Copulas for Time Series," June 2022.
27. R. Bolboacă and P. Haller, "Performance analysis of long short-term memory predictive neural networks on time series data," *Mathematics*, vol. 11, no. 6, pp. –, 2023.
28. S. Lloyd, "Least squares quantization in pcm," *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.
29. J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. Van Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin, *et al.*, "Darts: User-friendly modern machine learning for time series," *Journal of Machine Learning Research*, vol. 23, no. 124, pp. 1–6, 2022.
30. E. N. Lorenz, "Deterministic Nonperiodic Flow," *Journal of the Atmospheric Sciences*, vol. 20, pp. 130–141, Mar. 1963. Publisher: American Meteorological Society Section: Journal of the Atmospheric Sciences.
31. S.-K. Yang, C.-L. Chen, and H.-T. Yau, "Control of chaos in lorenz system," *Chaos, Solitons and Fractals*, vol. 13, no. 4, pp. 767–780, 2002.
32. B. Xue, "Phz4710 – introduction to biological physics." `https://cmp.phys.ufl.edu/PHZ4710/`, 2022. University of Florida, accessed November 8, 2025.
33. B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International journal of forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.
34. M. Pirani, P. Thakkar, P. Jivrani, M. H. Bohara, and D. Garg, "A comparative analysis of arima, gru, lstm and bilstm on financial time series forecasting," in *2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE)*, pp. 1–6, IEEE, 2022.
35. S. Yang, X. Yu, and Y. Zhou, "Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example," in *2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, pp. 98–101, IEEE, 2020.
36. A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 379(9), pp. 11121–11128, 2023.
37. P.-A. Buestán-Andrade, M. Santos, J.-E. Sierra-García, and J.-P. Pazmiño-Piedra, "Comparison of lstm, gru and transformer neural network architecture for prediction of wind turbine variables," in *International Conference on Soft Computing Models in Industrial and Environmental Applications*, pp. 334–343, Springer, 2023.