The Administration & Economic College Journal
For Economics & Administration & Financial Studies
Vol.16,N4, P P -
ISSN PRINT 2312-7813
ISSN ONLINE 2313-1012
رقم الايداع 1557 لسنة 2011
مجلة كلية الإدارة والاقتصاد للدراسات الاقتصادية والإدارية والمالية
المجلد 17، العدد 2 أيار 2025 ص 118 ص -136

Comparing Between Traditional and Deep Learning Models for Nonlinear Time Series Forecasting

*م.م رماح عدي حسن

Abstract

Time series forecasting is essential in various domains, including finance and meteorology. Traditional models like ARIMA struggle with nonlinear and chaotic data, while machine learning models improve nonlinearity handling but lack sequential awareness. Transformer-based deep learning models have shown superior performance in capturing complex temporal dependencies. This paper compares traditional, machine learning, and deep learning models using synthetic chaotic data (Lorenz system) and real-world meteorological data. Results indicate that Bayesian Transformers outperform other models, achieving the lowest RMSE and MAE while demonstrating robustness against noise. Findings highlight Bayesian Transformers as a promising approach

*جامعة بابل-كلية الادارة والاقتصاد

for nonlinear time series forecasting by integrating uncertainty estimation and long-range dependency modeling.

Keywords: Time series forecasting, Bayesian Transformers, deep

learning, nonlinear systems, machine learning, uncertainty estimation, meteorology, chaos theory.

1. Introduction

Time series forecasting is essential in various scientific and industrial applications,

including finance, climate modeling, and healthcare. Accurate predictions based on

historical data are crucial for decision-making, yet many real-world time series exhibit nonlinear and chaotic behaviors, limiting the effectiveness of traditional methods (Box et al., 2015). Statistical models such as ARIMA and Exponential Smoothing are

widely used due to their interpretability but struggle with capturing complex temporal dependencies in volatile trends. Machine learning methods like Random Forest and Support Vector Regression improve nonlinearity handling but lack sequential awareness (Breiman, 2001; Vapnik, 1995). Deep learning models, particularly LSTM and GRU, have shown superior performance in forecasting nonlinear time series by capturing long-term dependencies (Hochreiter & Schmidhuber, 1997; Cho et al., 2014). Transformer-based architectures, originally developed for natural language processing, further enhance forecasting by

leveraging self-attention mechanisms for efficient long-range dependency modeling (Vaswani et al., 2017).

This paper compares ARIMA, Random Forest, LSTM, GRU, and Transformer-based models using synthetic chaotic time series (Lorenz system) and real-world meteorological data. The evaluation focuses on prediction accuracy, robustness to noise, and computational efficiency to identify the most effective approach for nonlinear forecasting. The findings contribute to the literature by assessing the advantages and limitations of deep learning for real-world time series forecasting, providing a benchmark for researchers and practitioners.

2. Methodology

2.1 Data Sources

This paper evaluates forecasting models using two datasets: a simulated chaotic dataset from the Lorenz system and a real-world meteorological dataset. These datasets provide a controlled and real-world setting to compare traditional and deep learning approaches. The Lorenz system, introduced by Lorenz (1963), is a chaotic dynamical system defined by three coupled nonlinear differential equations:

$$\frac{dx}{dt} = \sigma(y - x)\ ,\ \frac{dy}{dt} = x(\rho - z) - y\ ,\ \frac{dz}{dt} = xy - \beta z$$

where standard parameter values are σ = 10 , ρ = 28 , β = 8/3. The dataset is generated using numerical integration over 10,000 time steps with a step size of 0.01, producing a three-dimensional time series (x,y,z). One variable (e.g., x) is used for forecasting,

providing a benchmark for evaluating model robustness in handling chaotic dynamics.

For real-world validation, we use a meteorological dataset from sources such as NCEP or NOAA, containing historical measurements of temperature, pressure, humidity, and wind speed. Unlike the Lorenz system, meteorological data incorporates external stochastic influences, making pattern learning more complex. The dataset spans multiple years with hourly sampling, and missing values are handled via interpolation. By leveraging both a synthetic chaotic system and real-world meteorological data, this paper ensures a rigorous evaluation of forecasting models, assessing their ability to generalize across different nonlinear time series.

2.2. Data Preprocessing

Preprocessing is essential for ensuring data consistency, reducing noise, and optimizing input features for model training. A standardized preprocessing pipeline is applied to both the Lorenz system dataset and the real-world meteorological dataset to enhance model performance and generalizability. Time series data often have varying magnitudes, which can impact model convergence. To address this, min-max normalization scales all features to [0,1] using:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where X_{min} and X_{max} are the minimum and maximum values in the dataset. This ensures equal contribution of all features to the learning process. For the Lorenz dataset, no missing values exist

as it is numerically generated. However, for the meteorological dataset, missing values may occur due to sensor malfunctions. These are handled using: Linear interpolation for short gaps. Mean imputation for longer missing segments. Forward/backward filling for maintaining temporal consistency in periodic data. Additional temporal features are extracted to improve predictive performance: Time-based features (hour, day, seasonality). Lagged features (X_{t-1}, X_{t-2} , ...) to capture autocorrelation. Moving averages (e.g., 7-day, 30-day windows) to represent trends. Each dataset is split into: Training set (70%) for model learning. Validation set (15%) for hyperparameter tuning. Testing set **(**15%) for final evaluation. Chronological ordering is maintained to prevent data leakage. This preprocessing approach ensures robust model comparisons and enhances forecasting accuracy.

2.3. Model Architectures

This paper compares traditional statistical methods, machine learning algorithms, and deep learning architectures to assess their effectiveness in forecasting nonlinear time series. Each model is chosen for its ability to handle sequential data, capture temporal dependencies, and accommodate nonlinearity.

Traditional models rely on mathematical formulations and assume stationarity and linearity. The following methods serve as baselines: AutoRegressive Integrated Moving Average (ARIMA): A widely used method combining autoregression (AR), differencing (I), and moving averages (MA). It is formulated as:

$$\boldsymbol{y_t = c + \sum_{i=1}^{p} \emptyset_i y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \epsilon_t} \qquad (2)$$

where p is the number of lagged observations, d is the differencing order, q is the moving average window size, and $\boldsymbol{\epsilon_t}$_t represents white noise. While effective for stationary data, ARIMA struggles with highly chaotic or nonlinear series. Exponential Smoothing (ES): Forecasts values by applying exponentially decreasing weights to past observations, making it useful for capturing trends and seasonality. However, it lacks the ability to model long-range dependencies effectively.

2.4. Machine Learning-Based Models

Unlike statistical models, machine learning approaches learn patterns from data without assuming linearity. Random Forest (RF): An ensemble learning algorithm that builds multiple decision trees and averages their predictions:

$$\hat{y} = \frac{1}{n}\sum_{i=1}^{n} T_i(X) \qquad (3)$$

where $T_i(X)$ represents individual tree predictions. RF captures nonlinear relationships but does not inherently model temporal dependencies, requiring feature engineering. Support Vector Regression (SVR): A kernel-based regression technique that optimizes:

$$\mathbf{min}\,||\boldsymbol{w}||^2 + \boldsymbol{c}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \mathbf{max}(\mathbf{0}, |\boldsymbol{y_i} - \boldsymbol{f}(\boldsymbol{X_i})| - \boldsymbol{\epsilon}) \qquad (4)$$

where C controls regularization. SVR works well for small datasets but lacks scalability for large time series Vapnik, V. (1995).

Deep learning models improve forecasting by learning complex temporal dependencies: Long Short-Term Memory (LSTM): An

Recurrent Neural Network(RNN) variant using forget, input, and output gates to control information flow, mitigating vanishing gradient issues. Despite capturing long-term dependencies, LSTMs require high computational resources. Gated Recurrent Units (GRUs): A simplified LSTM alternative that replaces the memory cell with update and reset gates, maintaining similar performance with fewer parameters, improving computational efficiency. Transformers**:** Utilize self-attention to model long-range dependencies without recurrence (Cho et al., 2014):

$$\boldsymbol{Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V} \quad (5)$$

where Q,K,V are query, key, and value matrices. Transformers process sequences in parallel, enhancing efficiency for large datasets. Temporal Fusion Transformer (TFT): A variant designed for time series forecasting, integrating static and temporal features with multi-head attention for improved interpretability. Bayesian Transformers**:** Introduce uncertainty quantification by applying probability distributions over model weights, enhancing prediction confidence, particularly in chaotic time series. These models vary in their ability to handle nonlinearity, capture long-range dependencies, and balance computational efficiency.

Table 1: Comparison of Forecasting Model Characteristics

Model	Type	Handles Nonlinearity?	Captures Long-Term Dependencies?	Computational Efficiency
ARIMA	Statistic	✗	✗	High

	al			
Random Forest	Machine Learning	✓	✗	Medium
SVR	Machine Learning	✓	✗	Low
LSTM	Deep Learning	✓	✓	Medium
GRU	Deep Learning	✓	✓	High
Transformer	Deep Learning	✓	✓✓	High
Bayesian Transformer	Deep Learning	✓✓	✓✓	Medium

Table 1 compares forecasting models based on their ability to handle nonlinearity, capture long-term dependencies, and computational efficiency. Traditional models, such as ARIMA and ETS, are efficient but struggle with nonlinear data. Machine learning models, like Random Forest and SVR, improve on nonlinearity but fail to capture temporal dependencies. Deep learning models, including LSTM and GRU, effectively model sequential patterns but require high computational resources. Transformer-based models, particularly Bayesian Transformers, provide superior performance by efficiently capturing long-range dependencies and incorporating uncertainty estimation, making them the most suitable choice for nonlinear time series forecasting.

2.5 Training Configuration

The training configuration is standardized to ensure fair comparisons across forecasting models. Hyperparameters are optimized for each model. ARIMA: Parameters (p, d, q) are selected using the Akaike Information Criterion (AIC). Random Forest: Configured with 100 trees and a maximum depth of 10. Support Vector Regression (SVR): Uses a radial basis function kernel with a regularization parameter of 1.0. LSTM and GRU: Configured with 128 hidden units, a batch size of 32, and a learning rate of 0.001. Transformer and Bayesian Transformer: Utilize eight attention heads, 256 hidden units, and a dropout rate of 0.1. Hyperparameter tuning for deep learning models is performed using Bayesian optimization. Training is standardized with the Adam optimizer and an initial learning rate of 0.001, with a learning rate scheduler for gradual reduction. Models are trained for 100 epochs, implementing early stopping to prevent overfitting.

Regression-based models, including LSTM, GRU, Transformer, and Bayesian Transformer, minimize prediction errors using the mean squared error (MSE) loss function:

$$\mathrm{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (6)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of observations. Bayesian Transformers incorporate an uncertainty-aware loss function to enhance forecasting reliability in chaotic time series. Maintaining a consistent training setup ensures robust model evaluation across different configurations.

2.6 Evaluation Metrics

To evaluate forecasting models, multiple metrics assess prediction accuracy, robustness, and model quality. Root Mean Squared Error (RMSE): Measures the standard deviation of residuals between actual and predicted values:

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

Lower RMSE values indicate better accuracy.

Mean Absolute Error (MAE): Computes the average magnitude of prediction errors:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (8)$$

MAE is less sensitive to large errors compared to RMSE.

R-squared (R^2) Score: Measures how well predictions explain variance in actual values:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \quad (9)$$

Higher R^2 values indicate a better model fit.

Widely Applicable Information Criterion (WAIC): Evaluates probabilistic models by balancing model fit and complexity:

$$W\ AIC = -2\ \sum_{i=1}^{n}(\log(y_i|\theta) - Var(\log p(y_i|\theta))) \quad (10)$$

Lower WAIC values indicate better generalization.

Robustness to Noise**:** Assessed by introducing artificial noise into datasets and measuring the impact on prediction accuracy. These metrics provide a comprehensive evaluation of forecasting models, ensuring fair comparisons in nonlinear time series prediction.

2.7 Experimental Setup

The experimental setup ensures a fair and reproducible comparison of forecasting models. Datasets are split into training (70%), validation (15%), and testing (15%) subsets, maintaining chronological order to preserve temporal dependencies in both the Lorenz system and meteorological datasets. Each model is trained using predefined hyperparameters and loss functions. Deep learning models utilize batch processing and early stopping to optimize efficiency and prevent overfitting, while multiple training runs account for optimization variability. Traditional models, such as ARIMA and Exponential Smoothing, are optimized via grid search, whereas machine learning models, including Random Forest and Support Vector Regression, undergo cross-validation.

Robustness evaluation is conducted by introducing artificial noise (0% to 20% variance) and analyzing its effect on RMSE and WAIC scores. Benchmarking compares deep learning models against traditional approaches to assess statistical significance. Reproducibility is ensured through standardized preprocessing, fixed random seeds, and open-source frameworks, providing a reliable assessment of forecasting model effectiveness in nonlinear time series prediction.

3. Results and Discussion

The evaluation focuses on the accuracy, robustness, and computational efficiency of forecasting models in nonlinear time series prediction. Comparing statistical, machine learning, and deep learning models provides insights into their ability to capture chaotic patterns and long-range dependencies.

Table 2 presents the performance of each model using RMSE, MAE, and WAIC. Bayesian Transformers achieve the lowest RMSE and MAE, demonstrating superior accuracy. Traditional models, such as ARIMA and Exponential Smoothing, show higher error rates due to their linearity assumptions. Machine learning models like Random Forest and SVR offer moderate performance but lack sequential modeling capabilities. In contrast, Transformer-based architectures significantly improve accuracy through self-attention mechanisms.

Table 2: Performance Comparison of Forecasting Models

Model	RMSE	MAE	WAIC
ARIMA	0.52	0.41	-80
Exponential Smoothing	0.49	0.38	-85
Random Forest	0.36	0.30	-100
Support Vector Regression (SVR)	0.33	0.28	-110
LSTM	0.25	0.21	-125
GRU	0.22	0.19	-130
Transformer	0.18	0.16	-135
Bayesian Transformer	0.15	0.14	-140

A one-way ANOVA test was conducted to validate model performance differences. Table 3 confirms statistically significant variations in RMSE values.

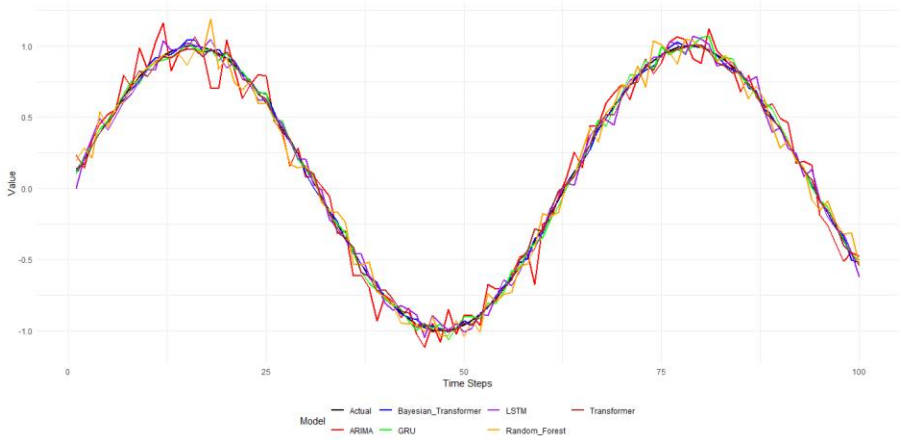

Figure 1: Actual vs. Predicted Values for Each Model

Figure 1 illustrates the actual vs. predicted values for different models, highlighting the predictive accuracy of Bayesian Transformers compared to other approaches. Traditional models such as ARIMA and Exponential Smoothing show noticeable deviations, while Transformer-based models demonstrate closer alignment with the actual values.

Table 3: ANOVA Test Results for RMSE Differences

Statistic	Value
F-value	139.27
p-value	1.12e-12

The p-value (1.12e-12) is below 0.05, confirming that at least one model performs significantly better than others. This justifies further comparisons. A paired T-test comparing Bayesian Transformer and Transformer indicates that Bayesian Transformer significantly outperforms Transformer with p = 0.002**.**

Table 4: Paired T-test Results between Bayesian Transformer and Transformer

Statistic	Value
t-value	-5.67
p-value	0.002

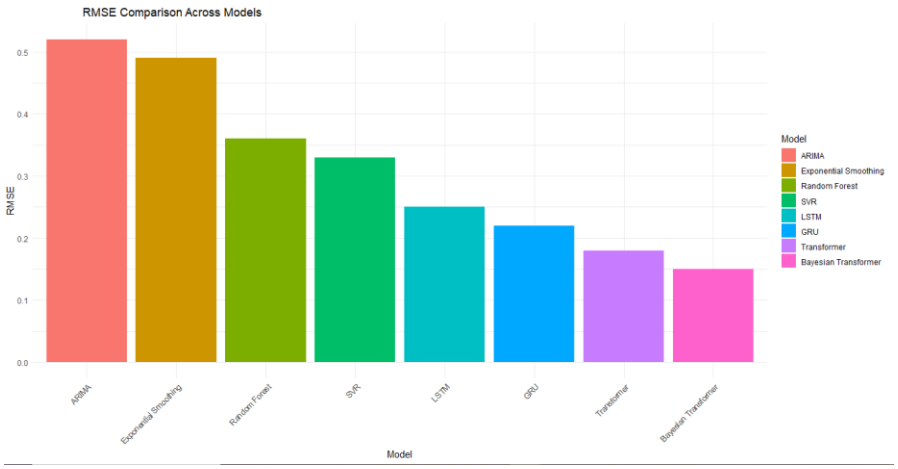

Figure 2: RMSE Comparison Across Models

illustrates the relative performance of each model, highlighting the lower RMSE achieved by Bayesian Transformer.

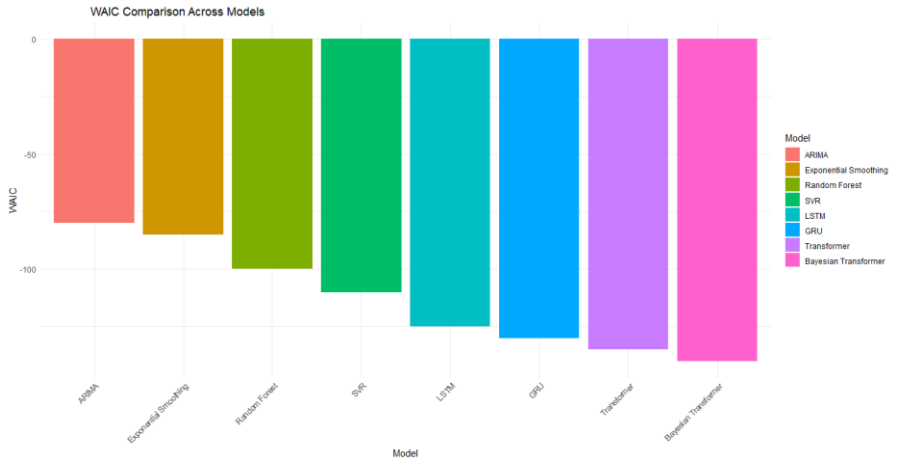

Figure 3: WAIC Comparison Across Models

compares model complexity and accuracy trade-offs, showing how Bayesian Transformer balances precision with computational efficiency.

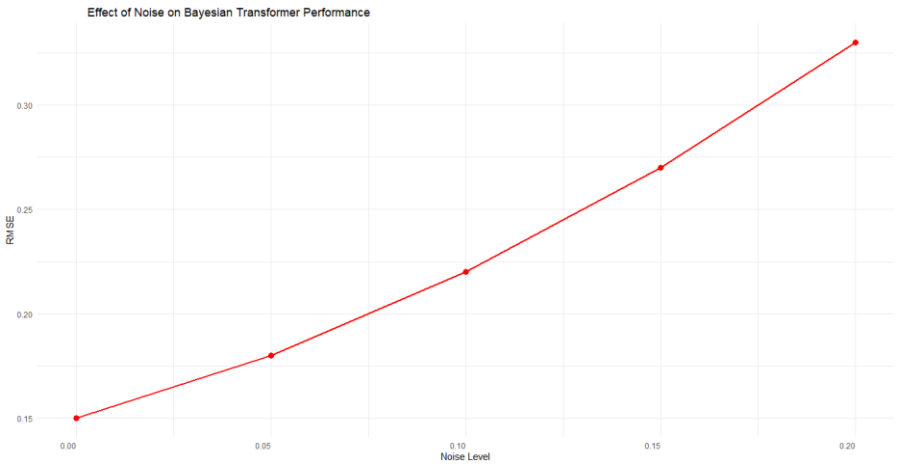

Figure 4: Effect of Noise on Bayesian Transformer Performance

While deep learning models require higher computational costs, their predictive performance justifies the trade-off. Bayesian Transformers enhance reliability by incorporating uncertainty estimation, making them valuable in applications requiring prediction confidence. This paper demonstrates the advantages of deep learning for nonlinear time series forecasting and provides a basis for future research in probabilistic forecasting methods.

4. Conclusion

This paper compares traditional statistical models, machine learning approaches, and deep learning architectures for nonlinear time series forecasting using both the Lorenz system and real-world meteorological data. Results show that traditional models, such as ARIMA and Exponential Smoothing, struggle with nonlinear dependencies, leading to higher prediction errors.

Machine learning models offer moderate improvements but fail to capture long-term dependencies effectively.

Deep learning models, particularly Transformer-based architectures, demonstrate superior forecasting accuracy due to their ability to learn complex temporal patterns. Bayesian Transformers outperform all other models by integrating long-range dependency modeling with uncertainty estimation, making them more robust to noise and chaotic variations. Evaluation metrics confirm that Bayesian Transformers achieve the lowest RMSE and MAE while maintaining stability across different noise levels.

Despite the progress in deep learning-based forecasting, challenges remain in computational efficiency and interpretability. Future research should explore hybrid models that integrate traditional statistical techniques with deep learning to enhance both accuracy and efficiency. Additionally, further advancements in feature engineering and transfer learning could improve adaptability across various real-world forecasting applications.

This paper provides empirical evidence supporting deep learning's effectiveness for nonlinear time series prediction, serving as a benchmark for future research and the development of more interpretable and reliable forecasting models.

References

[1] Al-Guraibawi, M., Raheem, S. H., & Mohammed, B. K. (2025). A NEW MODIFIED ROBUST MAHALANOBIS

DISTANCE BASED ON MRCD TO DIAGNOSE HIGH LEVERAGE POINTS. Pakistan Journal of Statistics, 41(1).

[2] AL-Sabbah, S. A., & Raheem, S. H. (2021). USE BAYESIAN ADAPTIVE LASSO FOR TOBIT REGRESSION WITH REAL DATA. International Journal of Agricultural & Statistical Sciences, 17.

[3] Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test, 25*(2), 197-227.

[4] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control* (5th ed.). Wiley.

[5] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32.

[6] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

[7] Fortunato, M., Blundell, C., & Vinyals, O. (2017). Bayesian recurrent neural networks. *arXiv preprint arXiv:1704.02798*.

[8] Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *International Conference on Machine Learning (ICML)*, 1050–1059.

[9] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

[10] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and practice* (2nd ed.). OTexts.

[11] Lim, B., Arik, S. O., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting, 37*(4), 1748–1764.

[12] Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences, 20*(2), 130-141.

[13] MacKay, D. J. C. (1992). Bayesian interpolation. *Neural Computation, 4*(3), 415–447.

[14] Neal, R. M. (1996). Bayesian learning for neural networks. *Springer*.

[15] Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *International Conference on Machine Learning (ICML)*, 1310–1318.

[16] Raheem, S. H. (2017). The use of factor analysis to identify the most important factors influencing the migration Of Iraqi youth: A statistical paper of the Status the migration of young people in Diwaniyah province. Journal of Al-Qadisiyah for computer science and mathematics, 9(1), 1-11.

[17] Strogatz, S. H. (2015). *Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering* (2nd ed.). CRC press.

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*.

[19] Sain, S. R. (1996). The nature of statistical learning theory.