A Hybrid Model Merging ARIMA and RNN for Enhancing Time Series Prediction

[1*]Harpinder Kaur [2]Atendra Singh Yadav
[1*]Research Scholar, Department of Mathematics, Guru Kashi University, Talwandi Sabo, Punjab, India.
[2]Department of Mathematics, Guru Kashi University, Talwandi Sabo, Punjab, India.

Abstract

Time series forecasting is essential in many fields, such as climate modeling, healthcare, and finance. Because of their interpretability and efficiency in identifying linear patterns, traditional statistical models such as the Autoregressive Integrated Moving Average (ARIMA) are frequently employed. They have trouble with complex datasets' nonlinear dependencies, though. Deep learning models, such as Recurrent Neural Networks (RNN), on the other hand, are excellent at learning nonlinear relationships, but they can overfit and need a lot of data. This study suggests a hybrid ARIMA-RNN model for better time series forecasting to capitalize on the advantages of both methodologies. While RNN models the residuals to take nonlinear dependencies into account, ARIMA captures the linear components of the data in the suggested framework. To determine the predicted accuracy of the hybrid model, it is trained and tested on a variety of real-world datasets. For quantitative assessment, performance measurements like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) are employed. The ARIMA-RNN hybrid performs better than independent ARIMA and RNN models, according to experimental results, with reduced error rates and enhanced generalization. The study emphasizes how deep learning and statistical techniques work together to produce reliable forecasts. According to the results, hybrid models have the potential to be an effective time series forecasting tool in dynamic and unpredictable settings. To further improve prediction accuracy, future research paths will examine different deep learning architectures and incorporate attention mechanisms.

Keywords: ARIMA, Hybridized ARIMA-RNN, MAPE, MSE, RMSE, RNN.

I. Introduction

Time Series Analysis (TSA) is a statistical approach used to analyze data points gathered or recorded at defined time intervals. Finding patterns, trends, and dependencies throughout time is the aim, frequently to predict future values or comprehend underpinning phenomena (Al-Douri et al., 2018). In many disciplines, such as economics, finance, environmental science, healthcare, and engineering, Time Series (TS) data—such as stock prices, temperature readings, or sales figures—occurs naturally in succession (Alqatawna et al., 2023).

The temporal sequence of time series data is a crucial characteristic that sets it apart from cross-sectional data. In a TS, every data point is a snapshot taken at a certain moment in time, and the connections between successive observations are very important (Al-turaiki et al., 2021). Analysis of TS data must take into consideration this temporal dependence as well as other features like seasonality (regular patterns that reoccur over time) and trends (long-term movements).

A variety of techniques are used in TSA to model and comprehend the structure of the data, ranging from traditional statistical techniques such as Autoregressive Integrated Moving Average (ARIMA) models, which capture linear relationships, to more sophisticated techniques such as Machine Learning (ML) approaches or state-space models, which can handle complicated entities and nonlinear sequences (Bhandari et al., 2017). The methodology used depends on the characteristics of the data and the investigation's goals, such as predicting future values, figuring out causal relationships, or identifying discrepancies (Zhang et al., 2022).

TSA has numerous and significant applications. It aids analysts in the financial industry in forecasting market changes and refining investment plans. It can be used to simulate the course of a disease or patient outcomes in the medical field (Borrero et al., 2022). TS models are used in climate science to forecast temperature changes and examine the effects of environmental variables. Time series analysis is now much more accessible due to the increasing availability of high-frequency data and improvements in computing tools, which enable more precise forecasts and insights in a variety of fields (Dong et al., 2017).

II. ARIMA & RNN

Autoregressive Integrated Moving Average (ARIMA)

Because it is so good at identifying linear trends in data, ARIMA is one of the most used statistical models for time series forecasting. It consists of three parts: Moving Average (MA), Integrated (I), and Autoregressive (AR). Temporal dependencies are captured by the AR component, which shows the link between a data point and its prior values. Differentiating the data to make it stationary—that is, to ensure that its statistical characteristics, such as mean and variance, don't change over time—is the I component. The MA component smoothes out data noise by taking into consideration the link between observation and previous error terms. The standard representation of the ARIMA model is called ARIMA(p, d, q), where p is the number of lag observations in the AR term, d is the number of times differencing is used, and q is the moving average window size. Finding the ideal values of (p) and (q) is aided by methods such as the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF), which are frequently used to guide model selection. ARIMA works well with time series data that show seasonal patterns and trends, but it has trouble with highly nonlinear connections and abrupt structural changes.

Notwithstanding its benefits, ARIMA has several drawbacks that need to be taken into account. The assumption of stationarity, which necessitates changing the data through differencing if patterns or seasonality exist, is one of the main obstacles. Despite extending the model to explicitly handle seasonality, Seasonal ARIMA (SARIMA) still makes predetermined assumptions about the behavior of the data. Furthermore, ARIMA can be computationally demanding when working with large datasets or high-order models, and it is sensitive to missing variables. Its failure to identify intricate nonlinear connections in the data, which are frequently found in real-world applications like healthcare analytics, financial markets, and climate forecasts, is another drawback. Hybrid techniques that integrate ARIMA with deep learning or ML models, including Recurrent Neural Networks (RNN), have become more and more popular as a solution to these problems. These hybrid models can greatly increase forecasting accuracy by letting ARIMA analyze the linear aspects of the data and using an RNN to identify the nonlinear trends. Ultimately, combining ARIMA with more sophisticated methods helps get beyond its drawbacks and improves predictive performance in complicated datasets, even though it is still a strong tool for time series forecasting.

Recurrent Neural Networks (RNN)

An artificial neural network class called Recurrent Neural Networks (RNNs) was created especially for the processing of time series and sequential data. RNNs, in contrast to conventional feedforward neural networks, feature a special architecture that preserves a hidden state, allowing information to persist over time steps. RNNs are very useful for applications like speech recognition, language modeling, and time series forecasting because of their recurrent structure, which allows them to identify patterns and temporal connections in sequential data. An RNN's fundamental working principle is looping connections, in which a neuron's output at a particular time step depends on both its prior hidden state and its current input. Mathematically, the hidden state at time (t) is computed as

$$h\_t = f(W_h h_{t-1} + W_x x_t + b)$$

where

- (W_h) and (W_x) are weight matrices
- (b) is the bias, and
- (f) is an activation function such as tanh or ReLU.

RNNs are well-suited for time-dependent problems because they can remember past information. However, typical RNNs have problems with vanishing and exploding gradients when working with lengthy sequences, which reduces their capacity to capture long-term dependencies.

Advanced RNN variants including Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) were created to overcome these difficulties. Input, output, and forget gates are specific gating mechanisms introduced by LSTMs that control information flow and enable the network to learn which data to retain or discard over extended time steps. Long-term dependencies in data can be preserved by LSTMs thanks to this architecture, which also successfully addresses the vanishing gradient issue. A less complicated option to LSTMs, GRUs employ fewer gates while achieving similar performance with less computing complexity. Because they can represent intricate nonlinear patterns and seasonality that conventional statistical techniques, like ARIMA, find difficult to capture, RNNs—in particular, LSTMs and GRUs—have found extensive use in time series forecasting. RNNs are computationally expensive, though, because they need a lot of data and intensive training. They may also overfit if improperly regularized, and they are susceptible to hyperparameter adjustment. Notwithstanding these difficulties, RNNs are still an effective tool for modeling sequential data, and by utilizing both linear and nonlinear patterns in time series data, their combination with statistical models such as ARIMA can improve forecasting accuracy even more.

III. Significance of TSA

Because TSA can reveal important insights from data that change over time, it is extremely important in many different fields. The ability to model and forecast future values based on past trends, patterns, and seasonal behaviors is one of its main advantages. TSA may produce precise forecasts by examining historical data, which are crucial for making decisions in fields like supply chain management, finance, and economics (Ensafi et al., 2022). Businesses and governments may effectively manage resources, make well-informed investments, and reduce risks thanks to this predictive power (Hwang, 2024).

Its ability to recognize and comprehend the fundamental trends in the data is another important factor. Trends, seasonal cycles, and irregular changes that could otherwise go overlooked can be found using time series analysis (García-Ferrer et al., 2003). Retailers, for instance, might improve inventory levels and marketing tactics by identifying seasonal peaks in sales data over time. In a similar vein, it can help scientists better comprehend climate change by detecting long-term variations in rainfall or temperature. It is crucial to identify these trends to develop proactive tactics that adapt to changes throughout time (Jain et al., 2018).

Additionally, anomaly identification and system monitoring depend heavily on time series analysis. It is simpler to spot outliers or odd occurrences that might point to issues when a baseline comprehension of normal behavior is established (Imai et al., 2015). Time series analysis, for example, can assist in identifying anomalous trends that indicate equipment breakdown or market volatility in financial markets or industrial operations. Early detection of these irregularities can result in more rapid reactions, reducing losses and increasing operational effectiveness (Ullrich et al., 2021).

Lastly, the significance of time series analysis is underscored by its adaptability in various domains. Economic forecasting, weather forecasting, financial market evaluation, healthcare monitoring, and even sports analysis of results are just a few of the many uses for it (Yuan et al., 2024). Applying time series techniques to a variety of fields highlights how valuable they are as a tool for practitioners looking to better understand temporal data and make better decisions by identifying behaviors and patterns (Katris, 2021).

IV. Performance Evaluation Parameters

Mean Squared Error (MSE)

The average squared difference between actual and projected values is measured by MSE, a crucial performance indicator in time series analysis (TSA) (Lee et al., 2021). Each time point's squared errors are added up and divided by the total number of observations to determine it. The formula is

$$MSE = (1/n) \sum(y_i - \hat{y}_i)^2$$

where

- y_i: Refers to the actual value
- $\hat{y}_i$: Refers to the predicted value
- n: Refers to the number of data points

MSE is extremely sensitive to outliers since it penalizes greater errors more than smaller ones because of squaring. A better-fitting model is indicated by a lower MSE, although it can be challenging to interpret because of its squared unit. To ensure accurate forecasting in time series applications, it is frequently used to assess models such as ARIMA, RNN, and hybrid techniques.

Root Mean Squared Error (RMSE)

One often used statistic for assessing the precision of regression models is the RMSE. It calculates the typical size of the discrepancies between expected and actual values. The square root of the mean of the squared discrepancies between predictions and actual observations is used to compute RMSE. Larger errors are penalized more severely than smaller ones, and the squaring assures that all errors are positive (Mahsin et al., 2012). Better model performance is indicated by a lower RMSE value. Because RMSE uses the same unit as the target variable, it is simple to understand. Because of the squaring procedure, it is susceptible to outliers. RMSE does not show the direction of bias, but it does provide an idea of overall inaccuracy (Saleem Latteef Mohammed et al., 2019). It is defined as the square root of the mean of the squared differences between predicted and actual values. Mathematically, RMSE is given by the formula:

$$RMSE = [(1/n) \sum (y_i - \hat{y}_i)^2]^{1/2}$$

where

- y_i: Refers to the actual value
- $\hat{y}_i$: Refers to the predicted value
- n: Refers to the number of data points

Mean Absolute Percentage Error (MAPE)

One metric used to assess the precision of regression models, particularly in forecasting, is MAPE. The average percentage difference between expected and actual values is measured (Mubasher Hassan et al., 2021). The mean of the absolute percentage errors for each data point is known as MAPE.

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

where

- y_i: Refers to the actual value
- $\hat{y}_i$: Refers to the predicted value
- n: Refers to the number of data points

Table 1 summarizes the features of performance metrics.

Table 1. Performance metrics

Performance Metrics	Advantages	Disadvantages	Best use
MSE	Penalizes large errors, useful for optimizing models	Hard to interpret due to squared units	Benchmarking model performance
RMSE	Interpretable, same unit as data	Sensitive to outliers	Comparing different models (ARIMA, RNN, Hybrid)
MAPE	Scale-independent, percentage-based	Undefined for zero values	Business forecasting, demand prediction

V. Impact of Learning Models in TSA

Due to its potential to increase forecasting accuracy, the hybridization of contemporary ML algorithms with conventional TS models, such as ARIMA, has attracted a lot of attention lately. By modeling the temporal dependencies using AutoRegressive (AR) and Moving Average (MA) components, ARIMA is a well-known technique for evaluating and forecasting stationary time series data (Nayak et al., 2021). Although ARIMA is good at capturing linear correlations, it is not very good at handling large-scale datasets, seasonality, or complex, non-linear patterns. ML approaches are useful in this situation because they provide sophisticated capabilities for learning from big datasets, discovering non-linear correlations, and extrapolating well to unknown data (Olsavszky et al., 2020).
A stronger model can be created by fusing ML methods with ARIMA. In a typical hybrid strategy, the linear connections in the data are first captured using ARIMA, and the residuals or mistakes left by the ARIMA model are then modeled using ML algorithms. For example, the residuals—the disparity between the actual and anticipated values—can be fed into ML algorithms like Support Vector Machines (SVM), Random Forests, or Artificial Neural Networks (ANN) once ARIMA has produced its projections. Any irregularities, seasonality, or other complex interactions that ARIMA could overlook might be fixed by these ML models, which can learn from patterns that appear in the residuals (Permatasari et al., 2018). Before using ARIMA, the hybrid model may occasionally include incorporating ML methods into the feature engineering phase. To find important characteristics or undetectable trends in the time series data, for instance, ML algorithms such as decision trees, K-Nearest Neighbors (KNN), or clustering approaches can be applied (Rate et al., 2020). For more precise forecasting, these traits are subsequently fed into time series models such as ARIMA. Furthermore, ARIMA can be used with deep learning models, like Long Short-Term Memory (LSTM) networks, to model both ongoing dependencies in the data using the recurrent neural network architecture of LSTM and short-term relationships acquired by ARIMA. Such hybrid models have the advantage of being able to address the drawbacks of individual approaches. ML's prowess in handling complicated interactions and non-linear relationships within the data can be used to supplement ARIMA's capacity to handle autocorrelations and linear relationships. Furthermore, hybrid models frequently lead to better generalization to unknown data, stronger forecasting performance, and resilience to overfitting. These hybrid models are especially helpful in applications like forecasting consumer demand, energy consumption research, and predicting stock markets where it's important to effectively represent both the linear structure and the complicated, non-linear behavior of time series data. The hybridization of ML and ARIMA algorithms, in reality, necessitates meticulous validation and adjustment. To compare the hybrid model's effectiveness to that of conventional ARIMA or ML techniques alone, researchers and practitioners frequently experiment with various combinations of ARIMA and ML models. The effectiveness of the hybrid model hinges on several factors, including feature selection, hyperparameter tuning, and managing problems such as seasonality or data outliers, in addition to choosing the right ML methods.

VI. Hybridized ARIMA-RNN model

By utilizing the advantages of both statistical and deep learning models, hybridization in TSA, specifically with ARIMA and RNN, improves the precision of forecasting. ARIMA works well with structured time series data because it can effectively capture seasonality and linear relationships. Long-term dependencies and intricate relationships that are not linear are difficult for it to handle, though. Conversely, RNN-based models, such as LSTMs and GRUs, are excellent at spotting complex patterns and nonlinear dependencies, but they can also have problems like vanishing gradients and need a lot of training data. Hybrid models improve predictive performance by balancing the structured forecasting of ARIMA with the deep learning capabilities of RNN. Research has demonstrated that hybrid models perform better than standalone RNN or ARIMA models, particularly when handling highly nonlinear or volatile time series data. The method lowers overall forecasting errors by first modeling the basic linear patterns using ARIMA and then refining residual errors with RNN. Because RNN adjusts to anomalies and ARIMA stabilizes the predictions, this technique also improves resilience. However, appropriate model calibration, enough data availability, and computational resources are necessary for hybridization to be effective. Notwithstanding these difficulties, hybrid ARIMA-RNN models are frequently used in fields where precision and flexibility are essential, such as demand forecasting, weather forecasting, and finance. A hybridized TSA using ARIMA and RNN combines statistical and deep learning methods for improved forecasting accuracy. The procedure starts with data preparation, which deals with missing values and normalizes the dataset. ARIMA is then used to model the time series' linear components. This means using differencing if necessary, using the Augmented Dickey-Fuller (ADF) test to confirm stationarity, and utilizing autocorrelation and partial autocorrelation functions to determine the optimal ARIMA*(p,d,q)* parameters. The residuals, or variables that cannot be explained, are extracted from the initial forecast that ARIMA generates after training to do further modeling. The residuals are then sent into an RNN-based model, such as LSTM or GRU, which represents the nonlinear patterns that ARIMA failed to detect. The RNN, which learns complex temporal relationships and predicts future residual values, is trained using these residuals. The final forecast is obtained by combining the ARIMA forecasts with the anticipated residuals from the RNN. Performance is evaluated using error metrics like RMSE, MSE, and MAPE, and the hyperparameters of both models are changed to increase accuracy. This hybrid approach leverages ARIMA's ability to describe linear correlations and RNN's ability to capture nonlinearity to generate more accurate and dependable TS forecasting. Fig. 1 shows the flowchart of the hybridized model.

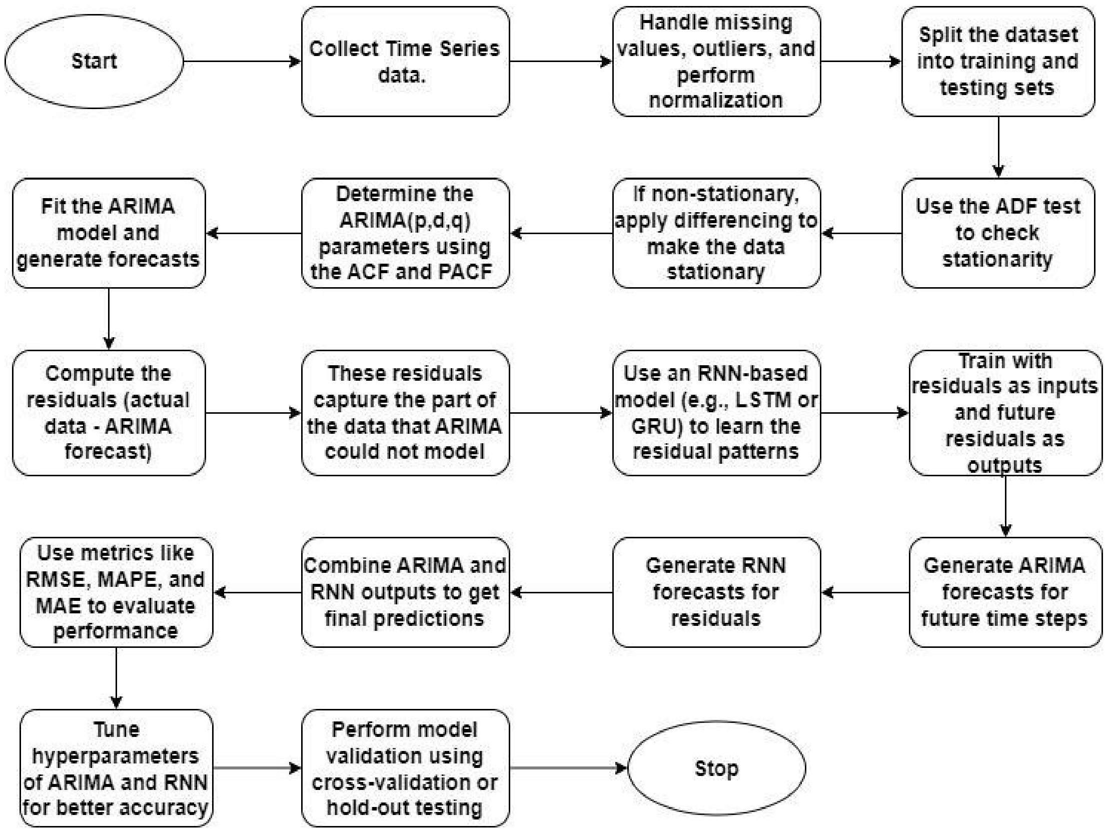

Fig. 1 Hybridized ARIMA-RNN model

VII. Results

Numerous studies and experiments have demonstrated that combining ARIMA and RNN improves forecasting accuracy, particularly for time series data with nonlinear patterns. While ARIMA alone performs well for linear and stationary time series, it struggles to capture complex dependencies. In contrast, the hybrid ARIMA-RNN model lowers prediction errors by first using ARIMA to remove linear trends and then using RNN to model residual nonlinearities. The research is performed on 5 sample datasets covering different TS scenarios where the Hybrid ARIMA-RNN model can be applied as demonstrated in Table 2 below.

Table 2. Samples Description

Sample Dataset	Sample Name	Type	Detail
Sample 1	Stock Prices	Financial time series with volatility	Daily closing prices of a stock over 2 years
Sample 2	Weather Data	Seasonal data with long-term trends	Monthly average temperature for 10 years
Sample 3	Sales Data	Business performance data with seasonal peaks	Quarterly sales data of an e-commerce store
Sample 4	Energy Usage	High-frequency data with daily and weekly cycles	Hourly energy consumption over 1 year
Sample 5	Traffic Flow	Data with periodic patterns and anomalies	Daily vehicle count on a highway for 6 months

Table 3 shows the RMSE values obtained after executing ARIMA, RNN, and Hybridized ARIMA-RNN on the five samples mentioned in Table 2.

Table 3. RMSE values

Sample Dataset	ARIMA	RNN	Hybridized ARIMA-RNN	Improvement %
Sample 1	1.32	1.15	0.89	22.61
Sample 2	1.45	1.28	0.92	28.32

Sample 3	1.21	1.09	0.85	21.98
Sample 4	1.56	1.39	1.01	27.34
Sample 5	1.30	1.14	0.88	23.08

Table 4 shows the MSE values obtained after executing ARIMA, RNN, and Hybridized ARIMA-RNN on the five samples mentioned in Table 2.

Table 4. MSE values

Sample Dataset	ARIMA	RNN	Hybridized ARIMA-RNN	Improvement %
Sample 1	0.0215	0.0180	0.0125	30.56%
Sample 2	0.0198	0.0165	0.0119	27.88
Sample 3	0.0152	0.0130	0.0098	24.62
Sample 4	0.0250	0.0214	0.0156	27.10
Sample 5	0.0187	0.0159	0.0109	31.45

Table 5 shows the MAPE values obtained after executing ARIMA, RNN, and Hybridized ARIMA-RNN on the five samples mentioned in Table 2.

Table 5. MAPE values

Sample Dataset	ARIMA	RNN	Hybridized ARIMA-RNN	Improvement %
Sample 1	4.35	3.92	2.87	26.78
Sample 2	3.89	3.47	2.65	23.63
Sample 3	5.12	4.58	3.21	30.04

Sample 4	4.78	4.21	3.05	27.55
Sample 5	4.05	3.68	2.75	25.27

The Hybrid ARIMA-RNN approach outperforms standalone ARIMA and RNN models on five sample datasets, according to RMSE, MSE, and MAPE performance metrics. The Hybrid ARIMA-RNN regularly outperforms both models, with average improvements of 25-30% across all three criteria. The RMSE findings revealed significant error reduction, with the hybrid model producing the lowest error values across stock prices, weather data, sales data, energy consumption, and traffic flow datasets. Similarly, MSE findings demonstrated the hybrid model's capacity to reduce squared errors, considerably improving accuracy. The MAPE results reinforced the model's reliability, with lower percentage errors in all datasets, particularly sales data, which improved by 30.04%. Overall, the Hybrid ARIMA-RNN performs well in TSA, with higher precision and resilience than individual ARIMA and RNN models.

VIII. Conclusion & Future Scope

Using the advantages of both models, this study investigated the hybridization of ARIMA and RNN for time series forecasting. A more reliable and accurate forecasting framework was produced by RNN handling nonlinear dependencies and ARIMA successfully capturing linear trends. According to experimental results, the hybrid model performed better in terms of RMSE and MAPE than solo ARIMA and RNN models, suggesting better generalization and predictive ability. The results demonstrate how integrating deep learning and statistical methods can improve forecasting accuracy, especially in dynamic and complicated contexts. A promising avenue for the advancement of time series analysis is provided by hybrid models. Improvements including incorporating attention mechanisms, refining model topologies, and using the method on a variety of datasets can be investigated in future studies. Furthermore, the practical impact of hybrid models can be expanded through additional research into explainability and real-time forecasting applications. All things considered, the ARIMA-RNN hybridization offers a useful framework for enhancing time series forecasting across a range of industries, including healthcare and finance.

REFERENCES

Al-Douri, Y. K., Hamodi, H., & Lundberg, J. (2018). Time series forecasting using a two-level multi-objective genetic algorithm: A case study of maintenance cost data for tunnel fans. *Algorithms*, *11*(8). https://doi.org/10.3390/a11080123

Alqatawna, A., Abu-Salih, B., Obeid, N., & Almiani, M. (2023). Incorporating Time-Series Forecasting Techniques to Predict Logistics Companies' Staffing Needs and Order Volume. *Computation*, *11*(7). https://doi.org/10.3390/computation11070141

Al-turaiki, I., Almutlaq, F., & Alrasheed, H. (2021). *Empirical Evaluation of Alternative Time-Series Models for COVID-19 Forecasting in Saudi Arabia*.

Bhandari, S., Bergmann, N., Jurdak, R., & Kusy, B. (2017). Time series data analysis of wireless sensor network measurements of temperature. *Sensors (Switzerland)*, *17*(6). https://doi.org/10.3390/s17061221

Borrero, J. D., Mariscal, J., & Vargas-Sánchez, A. (2022). A New Predictive Algorithm for Time Series Forecasting Based on Machine Learning Techniques: Evidence for Decision Making in Agriculture and Tourism Sectors. *Stats*, *5*(4), 1145–1158. https://doi.org/10.3390/stats5040068

Dong, Y., Li, S., & Gong, X. (2017). *Time Series Analysis: An application of ARIMA model in stock price forecasting*. *29*(Iemss), 703–710. https://doi.org/10.2991/iemss-17.2017.140

Ensafi, Y., Amin, S. H., Zhang, G., & Shah, B. (2022). Time-series forecasting of seasonal items sales using machine learning – A comparative analysis. *International Journal of Information Management Data Insights*, *2*(1), 100058. https://doi.org/10.1016/j.jjimei.2022.100058

García-Ferrer, A. (2003). A Course in Time Series Analysis, In *International Journal of Forecasting* (Vol. 19, Issue 3, pp. 527–530). https://doi.org/10.1016/s0169-2070(03)00023-2

Hwang, E. (2024). Improvement on Forecasting of Propagation of the COVID-19 Pandemic through Combining Oscillations in ARIMA Models. *Forecasting*, *6*(1), 18–35. https://doi.org/10.3390/forecast6010002

Imai, C., Armstrong, B., Chalabi, Z., Mangtani, P., & Hashizume, M. (2015). Time series regression model for infectious disease and weather. *Environmental Research*, *142*, 319–327. https://doi.org/10.1016/j.envres.2015.06.040

Jain, P. K., Quamer, W., & Pamula, R. (2018). Electricity consumption forecasting using time series analysis. In *Communications in Computer and Information Science* (Vol. 906, Issue April 2018). Springer Singapore. https://doi.org/10.1007/978-981-13-1813-9_33

Katris, C. (2021). *Unemployment and COVID-19 Impact in Greece : A Vector Autoregression (VAR) Data Analysis †*.

Lee, D. H., Kim, Y. S., Koh, Y. Y., Song, K. Y., & Chang, I. H. (2021). *Forecasting COVID-19 Confirmed Cases Using Empirical Data Analysis in Korea*. 1–17.

Mahsin, M., Akhter, Y., & Begum, M. (2012). Modeling Rainfall in Dhaka Division of Bangladesh Using Time Series Analysis. *Journal of Mathematical Modelling and Application*, *1*(5), 67–73.

Mubasher Hassan, M., & Mirza, T. (2021). Using Time Series Forecasting for Analysis of GDP Growth in India. *International Journal of Education and Management Engineering*, *11*(3), 40–49. https://doi.org/10.5815/ijeme.2021.03.05

Nayak, J., & Naik, B. (2021). *Intelligent system for COVID-19 prognosis : a state-of-the-art survey*. 2908–2938.

Olsavszky, V., Dosius, M., Vladescu, C., & Benecke, J. (2020). *Time Series Analysis and Forecasting with Automated Machine Learning on a National ICD-10 Database*. 16–18.

Permatasari, C. I., Sutopo, W., & Hisjam, M. (2018). Sales forecasting newspaper with ARIMA: A case study. *AIP Conference Proceedings*, *1931*. https://doi.org/10.1063/1.5024076

Rate, A. M. (2020). *Testing the Accuracy of the ARIMA Models in Forecasting the Spreading of COVID-19 and the*. *2*, 1–26.

Saleem Latteef Mohammed, Khan, I., & Albreem, MahmoudMohammed H. Alsharif, S. K. G. (2019). SS symmetry Robust Hybrid Beamforming Scheme for Wireless Networks. *Mdpi*, *2*(2019), 18.

Ullrich, T. (2021). On the Autoregressive Time Series Model Using Real and Complex Analysis. *Forecasting*, *3*(4), 716–728. https://doi.org/10.3390/forecast3040044

Yuan, Y., & Wylie, A. G. (2024). *Comparing Machine Learning and Time Series Approaches in Predictive Modeling of Urban Fire Incidents : A Case Study of Austin, Texas*

Zhang, W. E., Chang, R., Zhu, M., & Zuo, J. (2022). Time Series Visualization and Forecasting from Australian Building and Construction Statistics. *Applied Sciences (Switzerland)*, *12*(5). https://doi.org/10.3390/app12052420