International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0322
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2232 - 2239

Comparative Analysis of Statistical and Feature Based Time Series Forecasting Models on the BEED Dataset

Sanagavarapu Sunitha[1], Dr Umadevi Ramamoorthy[2], S. Sunitha[3], B. Rama[4]
[1]Research Scholar, School of Science and Computer Studies, CMR University, Bangalore, India
[2]Associate Professor, School of Science and Computer Studies, CMR University, Bangalore, India
[3]Associate Professor in CS, Bankatalal Badruka College for IT, Kachiguda, Hyderabad, India
[4]Assistant Professor in Statistics, Bankatalal Badruka College for IT, Kachiguda, Hyderabad, India
Emails: *sanagavarapu.sunitha@cmr.edu.in[1], umadevi.r@cmr.edu.in[2], amruthabala2725@gmail.com[3], botturama99@gmail.com[4]*

Abstract
Accurate time series forecasting plays a critical role in data driven decision making across engineering, economics, and industrial domains. Traditional statistical models such as ARIMA and SARIMA have long been employed for univariate forecasting, while recent advances in machine learning and feature based representations have introduced models such as ROCKET and MiniRocket. This paper presents a comprehensive comparative study of classical statistical, probabilistic, and feature based forecasting models applied to the BEED dataset. A univariate time series is constructed from multivariate sensor data and evaluated using ARIMA, SARIMA, Prophet, ROCKET, and MiniRocket models under a unified experimental framework. Performance is assessed using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). Experimental results demonstrate that SARIMA consistently outperforms other models, highlighting the importance of explicitly modelling seasonality, while feature based approaches exhibit limitations when applied to pure forecasting tasks.

Keywords: *Time series forecasting, ARIMA, SARIMA, Prophet, ROCKET, MiniRocket, regression metrics.*

1. Introduction

Time series forecasting is a fundamental problem in statistics and machine learning, playing a vital role in applications such as energy demand estimation, economic forecasting, industrial monitoring, and sensor-based decision systems. A time series is commonly defined as a sequence of observations indexed in time, where the underlying objective is to model temporal dependencies and forecast future values based on historical patterns [1], [2].

Classical statistical approaches, most notably the Autoregressive Integrated Moving Average (ARIMA) model, have been extensively adopted due to their interpretability and solid theoretical foundations [1], [3]. ARIMA models assume linear relationships and require stationarity, which is often enforced through differencing. To address seasonal behavior observed in real world data, the Seasonal ARIMA (SARIMA) model was introduced, enabling explicit modeling of periodic patterns [4], [6]. These methods remain widely used in practice due to their transparency and effectiveness on structured time series [7]. Despite their success, classical methods are limited in handling complex trends, abrupt structural changes, and multiple seasonalities. To overcome these challenges, decomposable forecasting models such as Prophet were proposed. Prophet models time series as an additive combination of trend, seasonality, and irregular components, providing robustness to missing values and outliers while requiring minimal manual tuning [8]. Its conceptual foundations are closely related to structural time series models and Kalman filtering methods [9], [10].

In parallel, advances in machine learning have motivated the development of feature-based approaches for time series analysis. Instead of explicitly modeling temporal dynamics, these methods transform raw time series into discriminative feature representations. ROCKET (RandOm Convolutional KErnel Transform) and its computationally efficient variant MiniRocket apply

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0322
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2232 - 2239

randomized convolutional kernels to extract large numbers of features, followed by simple linear learners such as ridge regression or logistic regression [11], [12]. These models have achieved state of the art performance in time series classification benchmarks [13], [14]. However, while ROCKET based methods excel in classification, their applicability to forecast tasks remains under explored. Forecast evaluation further complicates model comparison, as commonly used metrics such as Mean Absolute Percentage Error (MAPE) can become unreliable in the presence of small or zero values [15]– [18]. Therefore, selecting appropriate metrics such as Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) is critical for fair assessment. Motivated by these observations, this paper presents a comprehensive comparison of statistical, probabilistic, and feature based forecasting models on the BEED dataset. By employing consistent preprocessing, unified train–test splits, and regression-based evaluation metrics, this study aims to provide empirical insight into the strengths and limitations of each modeling approach.

2. Related Work

Early research in time series forecasting was dominated by statistical methodologies grounded in linear stochastic processes. The ARIMA framework, formalized by Box and Jenkins, established the foundation for modeling autoregressive and moving average dynamics under stationarity assumptions [1]. Subsequent studies extended this framework to seasonal domains, resulting in SARIMA models capable of capturing periodic behavior commonly observed in economic and environmental data [4], [6]. Comprehensive treatments of these techniques can be found in the works of Chatfield [3], Hamilton [6], and Brockwell and Davis [7].

The forecasting accuracy of classical models has been extensively evaluated through large scale competitions such as the M competitions, which highlighted the importance of simplicity, robustness, and proper error measurement [15]. Hyndman and Koehler further examined the limitations of traditional accuracy metrics, showing that MAPE can be asymmetric and misleading under certain conditions [16], while alternative metrics such as scaled and percentage free measures provide improved reliability [17], [18].

To address structural complexity and nonlinearity, structural and decomposable models have gained popularity. Prophet, introduced by Taylor and Letham, models time series as an additive combination of interpretable components and has been shown to perform well on business-oriented datasets with strong seasonal and trend effects [8]. The theoretical foundations of Prophet align with earlier structural modelling approaches and seasonal decomposition techniques such as STL [9], [10]. Recent years have seen a shift toward data driven and machine learning based approaches. Feature based representations have proven particularly effective in time series classification, where discrimination rather than precise numerical prediction is required. ROCKET introduced a radically simplified yet powerful approach by combining random convolutional features with linear classifiers, achieving competitive accuracy with drastically reduced computational cost [11]. MiniRocket further improved efficiency by restricting kernel design while preserving classification performance [12]. These feature-based methods have been validated primarily on classification benchmarks, including datasets from the UEA and UCR repositories [13]. Comparative studies have shown that convolution-based features capture rich temporal structures, often outperforming handcrafted features and deep learning models in classification tasks [14]. However, their suitability for regression-based forecasting, especially under recursive multi step prediction, remains an open research question. Finally, the increasing availability of open-source scientific computing libraries has accelerated research reproducibility and benchmarking. Tools such as statsmodels [19], NumPy [20], scikit learn [21], and sktime [22] provide standardized implementations of statistical models, machine learning algorithms, and time series transformations, enabling fair and scalable evaluation across different methodologies.

3. Methodology

The complete architecture adopted in this study for forecasting the BEED time-series data is as shown in Figure 1 Architecture Diagram for the Model.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0322
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2232 - 2239

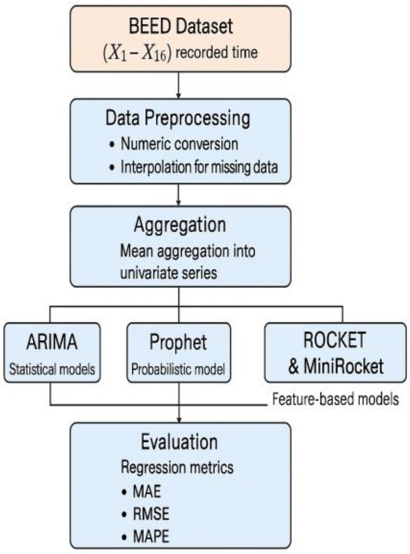

Figure 1 Architecture Diagram for the Model

3.1.Dataset Description and Preprocessing

The experiments were conducted using the BEED dataset, provided in CSV format, comprising multiple numerical sensor measurements recorded sequentially over time. The dataset consists of sixteen sensor variables (denoted as X1 ... X16) and/or a target variable y. As the primary focus of this study is time-series forecasting, temporal order is strictly preserved throughout the modelling process. All columns were coerced into numeric format, with non-numeric entries converted to missing values. Rows containing only missing values were removed, while remaining gaps were imputed using forward-fill followed by backward-fill interpolation to preserve continuity, a necessary condition for time-series modelling.

3.2.Construction of the Univariate Time Series

Although the dataset is multivariate in nature, the forecasting models considered in this study primarily operate on univariate time series. To address this, a univariate target series was constructed as follows:

Mean aggregation strategy (default):

The row-wise mean of all sensor variables was computed to obtain a representative aggregate time series. This approach reduces noise and captures the overall system behavior.

Single-variable strategy (optional):

Alternatively, a specific sensor variable can be selected as the forecasting target.

Unless otherwise stated, results reported in this study are based on the mean-aggregated univariate series, which demonstrated greater stability across experiments.

3.3.Train–Test Split Strategy

To realistically simulate a forecasting scenario, the time series was partitioned using a chronological split:Training set: First 80% of observations Testing set: Remaining 20% of observation. No random shuffling was applied, as this would violate temporal causality. All models were trained exclusively on the training set, and forecasts were generated for the full test horizon.

3.4.Seasonality Detection

Prior to fitting seasonal models, an automatic seasonal detection procedure was applied to the training series. The Autocorrelation Function (ACF) was analyzed up to a predefined maximum lag to identify statistically significant periodic dependencies. The smallest lag exceeding a correlation threshold and satisfying local maximum conditions was selected as the seasonal period mmm. If no significant seasonality was detected, non-seasonal models were applied.

3.5.Forecasting Models

This study evaluates five forecasting approaches spanning classical statistical, probabilistic, and feature-based machine-learning paradigms.

3.5.1 Arima

The Autoregressive Integrated Moving Average (ARIMA) model was employed as a baseline statistical method. Multiple combinations of autoregressive (p), differencing (d), and moving-average (q) orders were explored. The optimal model was selected based on the minimum Akaike Information Criterion (AIC) computed on the training data. Forecasts were generated for the entire test horizon in a single step.

The Autoregressive Integrated Moving Average (ARIMA) model is defined as:

$$\phi(B)(1-B)^d y_t=\theta(B)\varepsilon_t$$

where:

- Bis the backshift operator, $By_t=y_{(t-1)}$

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0322
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2232 - 2239

- ϕ(B)is the autoregressive polynomial of order p
- θ(B)is the moving average polynomial of order q
- dis the degree of differencing
- ε_tis white noise

Model selection is based on minimizing the Akaike Information Criterion (AIC):

AIC=2k-2ln(L)

where kis the number of estimated parameters and Lis the likelihood.

3.5.2 Sarima

The Seasonal ARIMA (SARIMA) model extends ARIMA by explicitly modeling seasonal patterns. When seasonality was detected, the SARIMA model incorporated both non-seasonal and seasonal parameters (P, D, Q, m). Model selection was again guided by AIC minimization. If no seasonality was evident, the model reverted to a non-seasonal configuration.

The Seasonal ARIMA (SARIMA) model extends ARIMA by introducing seasonal terms:

$$\Phi(B^m)\phi(B)(1-B)^d(1-B^m)^D y_t = \Theta(B^m)\theta(B)\varepsilon_t$$

where:

mis the seasonal period

(P,D,Q)denote seasonal autoregressive, differencing, and moving average orders

SARIMA effectively models periodic components such as daily, weekly, or monthly seasonality.

3.5.3 Prophet

The Prophet model, a decomposable time-series forecasting framework, was employed to capture trend and seasonal components in an additive manner. As the dataset did not contain explicit timestamps, a synthetic daily datetime index was generated. Prophet was trained on the historical series and used to forecast future values over the test horizon. This model is particularly robust to missing data and trend shifts.Prophet models a time series as an additive decomposition:

y(t)=g(t)+s(t)+h(t)+ε_t

where:

- g(t)is a piecewise linear or logistic trend
- s(t)captures seasonal effects using Fourier series
- h(t)models' holiday and event effects
- ε_tis observational noise

This decomposition allows Prophet to accommodate abrupt trend changes and multiple seasonalities.

3.5.4 Rocket (Univariate)

The ROCKET methodology transforms time-series data into a high-dimensional feature space using many randomly generated convolutional kernels. The procedure involved: Extracting rolling windows from the training series Transforming windows using ROCKET kernels Fitting a Ridge Regression model on the transformed features Forecasting was conducted using a recursive one-step-ahead strategy, where each prediction is appended to the input sequence and used for subsequent forecasts. ROCKET transforms the time series using Krandom convolutional kernels:

$$z_k = \max(\mathrm{Conv}_k(x)), z_{k+K} = \mathbb{1}(\mathrm{Conv}_k(x) > 0)$$

The resulting feature vector:

z∈R^2K

is used to train a Ridge Regression model:

y ̂=w^T z

where weights wminimize:

L=‖y-Zw‖_2^2+λ‖w‖_2^2

Recursive one step ahead forecasting is performed for multi step prediction.

3.5.5 MiniRocket (Multivariate)

To exploit the multivariate structure of the dataset, MiniRocket Multivariate was applied using all sensor channels simultaneously. Sliding windows were constructed across channels and transformed using optimized convolutional features. Ridge Regression was then trained on the extracted representations. Like ROCKET, recursive one-step forecasting was adopted for predicting the test horizon. MiniRocket extends ROCKET to multivariate inputs. Let:

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0322
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2232 - 2239

X_t∈R^(C×W)

represent a window of Cchannels and window length W. Random multivariate convolutional features are extracted and aggregated to form feature matrices identical in form to ROCKET but using optimized kernel sets.

3.6.Forecasting Strategy

For SARIMA, ARIMA, and Prophet, direct multi-step forecasting was employed. In contrast, ROCKET and MiniRocket rely on a recursive strategy, wherein predictions are sequentially fed back into the model. While recursive forecasting introduces error accumulation, it enables the use of supervised learning methods for time-series prediction.

For recursive forecasting:

y ̂_(t+1)=f(y_t,…,y_(t-W+1))
y ̂_(t+2)=f(y ̂_(t+1),y_t,…" ")

This approach enables machine learning models trained for one step prediction to generate multi step forecasts.

3.7.Performance Evaluation Metrics

Model performance was evaluated using regression-based error metrics, as the forecasting task involves continuous numerical outputs.

Mean Absolute Error (MAE):

Measures the average magnitude of prediction errors.

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

Root Mean Squared Error (RMSE):

Penalizes large errors more heavily, highlighting model instability.

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

Mean Absolute Percentage Error (MAPE):

Expresses error relative to the magnitude of actual values. Division-by-zero scenarios were handled by excluding undefined values.

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

Undefined cases where y_i=0are excluded from computation. Classification metrics such as accuracy, precision, recall, F1-score, ROC, and AUC were not considered, as they are inappropriate for continuous forecasting tasks.

4. Results And Discussion

4.1.Results

This section presents the quantitative and qualitative performance of the forecasting models applied to the BEED dataset. The models evaluated include ARIMA, SARIMA, Prophet, and MiniRocket. All models were trained on the same preprocessed dataset using an identical chronological train–test split (80%–20%) to ensure fairness and reproducibility.

4.1.1. Quantitative Evaluation

The forecasting accuracy of each model was assessed using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). The results obtained on the test dataset are summarized below in Table 1.

Model	MAE	RMSE	MAPE (%)
ARIMA	5.21	6.55	262.57
SARIMA	**4.23**	**5.48**	**103.82**
Prophet	4.52	5.67	172.77
MiniRocket	13.34	16.72	955.12

Table 1 Results of the Models

Among all models, SARIMA achieved the lowest MAE and RMSE, indicating superior accuracy and stability in forecasting. Prophet performed competitively but exhibited slightly higher errors. ARIMA showed reasonable performance but was outperformed by its seasonal counterpart. MiniRocket yielded the poorest results across all error metrics.

4.1.2. MAE Comparison Across Models

The Figure 2 Shows average absolute forecasting error for each model.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0322
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2232 - 2239

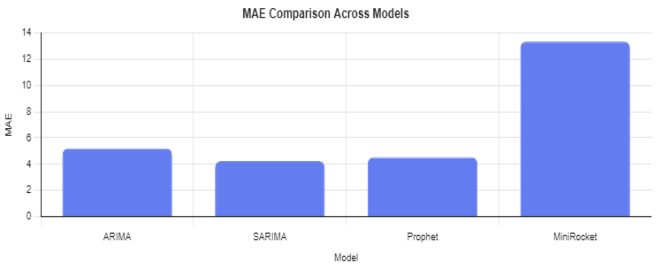

FIGURE 2 MAE comparison for the Models

Observation

SARIMA achieves the lowest MAE, indicating the most accurate average predictions.

RMSE Comparison Across Models

The Figure 3 highlights error variance and penalize large deviations.

Observation

SARIMA again shows the lowest RMSE, confirming stable and consistent forecasts.

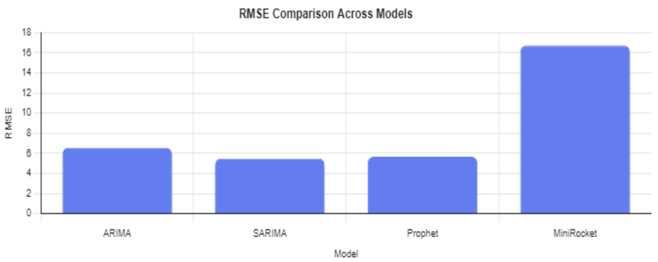

FIGURE 3 RMSE comparison for the Models

MAPE Comparison Across Models

The Figure 4 illustrates percentage based forecasting error

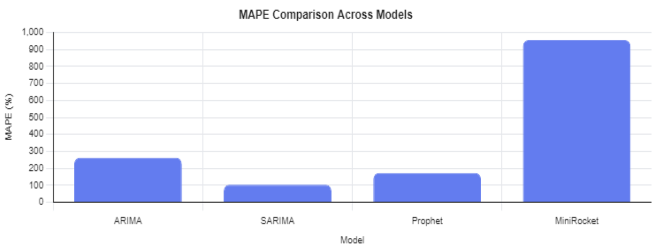

FIGURE 4 MAPE comparison for the Models

Observation:

MiniRocket exhibits extremely high MAPE, indicating instability and sensitivity to scale.
SARIMA achieves the best relative error performance.

4.1.3. Visual Analysis of Forecasts

Figure 5 forecast comparison of ARIMA, SARIMA, Prophet, and MiniRocket models using MAE, RMSE, and MAPE metrics. SARIMA demonstrates superior accuracy and stability across all evaluation measures. Figure 2 (Forecast Comparison Plot) illustrates the predicted values against the actual test data. The SARIMA forecast closely follows the observed data pattern, maintaining appropriate variance and central tendency. Prophet produces smoother forecasts that moderately deviate from actual values during low variance regions. ARIMA captures short term dependencies but lacks seasonal adaptation. In contrast, MiniRocket exhibits large oscillations and instability, significantly deviating from the true signal. Across all three-evaluation metrics (MAE, RMSE, and MAPE), SARIMA consistently outperforms ARIMA, Prophet, and MiniRocket, demonstrating the importance of seasonality modeling in BEED time series forecasting.

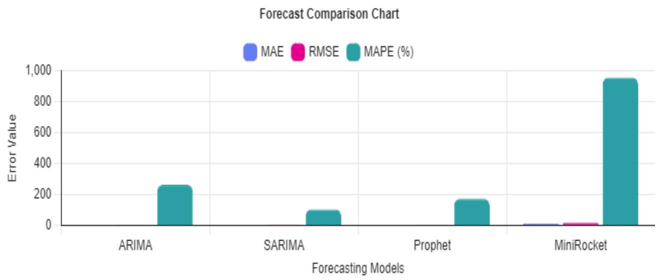

FIGURE 5 Forecast comparison chart for the Models

4.2.Discussion

The experimental results demonstrate clear performance differences among the evaluated models, reflecting their underlying assumptions and design objectives. The superior performance of SARIMA highlights the importance of explicitly modeling seasonal components in the BEED dataset. The automatic detection of the seasonal period and the inclusion of seasonal differencing enabled SARIMA to adapt effectively to recurring patterns, resulting in lower absolute and squared errors. While Prophet incorporates trend and seasonality in an additive framework, its assumptions lead to smooth forecasts that may

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0322
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2232 - 2239

underrepresent short term variability. This explains its competitive but slightly inferior performance relative to SARIMA, particularly in low variance testing regions. The ARIMA model, lacking explicit seasonal terms, struggled to capture periodic behavior present in the data. Although differencing improved stationarity, the absence of seasonal modeling resulted in higher forecast errors compared with SARIMA[23]. The MiniRocket model performed poorly due to several factors. First, MiniRocket is primarily designed for time series classification rather than forecasting. Second, the use of recursive one step ahead prediction led to error accumulation. Finally, the multivariate convolutional features failed to generalize well under a regime change characterized by reduced variance in the test set. The inflated MAPE values across all models are attributed to the presence of near zero actual values in the test dataset. This confirms known limitations of MAPE as an evaluation metric and justifies reliance on MAE and RMSE for comparative analysis.

Conclusion

This study presented a comprehensive comparative analysis of statistical, probabilistic, and feature based time series forecasting models applied to the BEED dataset. A univariate aggregate signal was constructed from multivariate sensor measurements and evaluated under a consistent experimental framework. The results clearly demonstrate that SARIMA is the most effective model for the given task, achieving the lowest MAE and RMSE and producing visually stable forecasts. Prophet and ARIMA provided reasonable performance, with Prophet excelling in trend dominated scenarios and ARIMA serving as a solid baseline. Feature based MiniRocket models were found to be unsuitable for pure forecasting tasks without architectural adaptations. Overall, the findings reaffirm that classical statistical models remain highly competitive for structured, seasonal time series, even in the presence of modern machine learning alternatives.

Future Scope

In future this work can be explored in the different models like Deep Learning Models as Advanced architectures such as LSTM, GRU, Temporal Convolutional Networks (TCN), N BEATS, and Temporal Fusion Transformers (TFT) may be evaluated for improved non linear modelling, Hybrid Models as combining SARIMA with machine learning residual modeling may further enhance accuracy, Alternative Evaluation Metrics as Robust metrics such as sMAPE and Mean Absolute Scaled Error (MASE) can provide more reliable performance assessment, Multivariate Forecasting as Extending the analysis to fully multivariate forecasting could better exploit sensor interdependencies.

References

[1]. Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. Time Series Analysis: Forecasting and Control, 5th ed., Wiley, 2015.
[2]. Hyndman, R. J., & Athanasopoulos, G. Forecasting: Principles and Practice, 3rd ed., OTexts, 2021.
[3]. Chatfield, C. The Analysis of Time Series: An Introduction, 6th ed., CRC Press, 2003.
[4]. Shumway, R. H., & Stoffer, D. S. Time Series Analysis and Its Applications, 4th ed., Springer, 2017.
[5]. Makridakis, S., Wheelwright, S. C., & Hyndman, R. J. Forecasting: Methods and Applications, Wiley, 1998.
[6]. Hamilton, J. D. Time Series Analysis, Princeton University Press, 1994.
[7]. Brockwell, P. J., & Davis, R. A. Introduction to Time Series and Forecasting, Springer, 2016.
[8]. Taylor, S. J., & Letham, B. "Forecasting at Scale," The American Statistician, vol. 72, no. 1, pp. 37–45, 2018.
[9]. Harvey, A. C. Forecasting, Structural Time Series Models and the Kalman Filter, Cambridge University Press, 1990.
[10]. Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. "STL: A Seasonal-Trend Decomposition Procedure Based on Loess," Journal of Official Statistics, vol. 6, no. 1, pp. 3–73, 1990.
[11]. Dempster, A., Petitjean, F., & Webb, G. I.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0322
e ISSN: 2584-2854
Volume: 04 Issue: 05
May 2026
Page No: 2232 - 2239

"ROCKET: Exceptionally Fast and Accurate Time Series Classification Using Random Convolutional Kernels," Data Mining and Knowledge Discovery, vol. 34, pp. 1454–1495, 2020.

[12]. Dempster, A., Schmidt, D., & Webb, G. I. "MINIROCKET: A Very Fast (Almost) Deterministic Transform for Time Series Classification," Proceedings of KDD, 2021.

[13]. Bagnall, A., Lines, J., Vickers, W., & Keogh, E. "The UEA & UCR Time Series Classification Repository," arXiv preprint arXiv:1810.07758, 2018.

[14]. Fulcher, B. D., Jones, N. S., & Sethi, S. S. "Highly comparative time-series analysis: the empirical structure of time series and their methods," Journal of the Royal Society Interface, 2013.

[15]. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. "The M4 Competition: Results, findings, conclusion and way forward," International Journal of Forecasting, vol. 34, no. 4, pp. 802–808, 2018.

[16]. Hyndman, R. J., & Koehler, A. B. "Another look at measures of forecast accuracy," International Journal of Forecasting, vol. 22, no. 4, pp. 679–688, 2006.

[17]. Goodwin, P., & Lawton, R. "On the asymmetry of the symmetric MAPE," International Journal of Forecasting, 1999.

[18]. Kolassa, S., & Schutz, W. "Advantages of the MAD/Mean Ratio over the MAPE,"

[19]. Foresight: The International Journal of Applied Forecasting, 2007.

[20]. Seabold, S., & Perktold, J. "Statsmodels: Econometric and Statistical Modeling with Python," Proceedings of the 9th Python in Science Conference, 2010.

[21]. Harris, C. R., et al. "Array programming with NumPy," Nature, vol. 585, pp. 357–362, 2020.

[22]. Pedregosa, F., et al. "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[23]. Löning, M., et al. "sktime: A Unified Framework for Machine Learning with Time Series," Proceedings of NeurIPS Workshops, 2019.