# Advanced Time Series Forecasting Project Report

## 1. Introduction

This project focuses on Advanced Time Series Forecasting using a synthetic multivariate energy dataset.

The goal is to build a complete forecasting pipeline using:

• LSTM neural network

• SARIMA statistical baseline

• SHAP explainability

• Feature engineering and hyperparameter tuning

The work strictly follows the requirements defined in the assessment module.

## 2. Dataset Description

A synthetic multivariate time series dataset with 2000+ hourly records is programmatically generated.

It includes realistic energy consumption behaviour based on temperature, humidity, and seasonality.

Features:

• load (target)

• temperature

• humidity

• hour

• day_of_week

• is_weekend

Additional engineered features:

• load_lag_1, load_lag_2, load_lag_3, load_lag_24

• load_rollmean_3, load_rollmean_24

## 3. Feature Engineering

Feature engineering enhances the predictive power of the model by incorporating past context.

Methods used:

• Lag features to capture short-term and daily dependencies

• Rolling mean windows for smoothing long-term patterns

• Time-based calendar features

• Data scaling using StandardScaler

These steps significantly improve the LSTM model’s understanding of temporal dynamics.

## 4. Baseline Model: SARIMA

A SARIMA(2,1,2) × (1,1,1,24) model is used as the baseline.

Reasons for using SARIMA:

• Captures seasonality (24-hour cycle)

• Provides a statistical benchmark

• Helps compare classical vs deep learning performance

Forecasts are generated for the test window and aligned with LSTM predictions.

## 5. LSTM Deep Learning Model

LSTM (Long Short-Term Memory) is a recurrent neural network architecture suitable for sequential data.

Key details:

• Input window size: 24 hours

• Tuned using KerasTuner (units, dropout, learning rate)

• Early stopping applied during training

• Trained on scaled sequences of features

The LSTM learns nonlinear and long-term dependencies that SARIMA cannot capture effectively.

## 6. Hyperparameter Tuning

KerasTuner RandomSearch was used to search for optimal LSTM configurations.

Parameters tuned:

• Number of LSTM units (32–128)

• Dropout rate (0.0–0.5)

• Learning rate (1e-2, 1e-3, 1e-4)

The best model is selected using validation loss and then retrained.

## 7. Explainability with SHAP

To interpret the LSTM model, SHAP DeepExplainer is applied.

Actions performed:

• SHAP values computed for a subset of test samples

• Time-step SHAP values averaged for each feature

• Feature importance plotted

Key findings:

• Lag features had the strongest influence

• Time-based features contributed moderately

• Temperature and humidity influenced load depending on seasonality

## 8. Model Evaluation

Evaluation metrics used:

• RMSE (Root Mean Squared Error)

• MAE (Mean Absolute Error)

• MAPE (Mean Absolute Percentage Error)

Both SARIMA and LSTM models were compared on the test set.

Expected outcome:

• LSTM outperforms SARIMA due to better nonlinear modeling

• SARIMA still provides a strong and interpretable baseline

## 9. Conclusion

This project successfully demonstrates an end-to-end forecasting pipeline combining classical and deep learning models, enriched with explainability.

Key points:

• Dataset generation, preprocessing, engineering completed

• SARIMA baseline implemented

• LSTM tuned and trained effectively

• SHAP explainability provided insight into feature contributions

The LSTM model is more accurate and suitable for real-world energy forecasting scenarios.