Configuration Manual

JISNO GEORGE

Student ID: 23324937

School of Computing

National College of Ireland

Supervisor: Brian Byrne

1. Overview

This artefact reproduces the time series forecasting analysis for two domains:
1. Structured and seasonal data: Residential daily electrical power consumption (2017 dataset).
2. Volatile and non-stationary data: Bitcoin daily weighted prices (2011-2021 dataset).

The work compares the performance of different modelling approaches (GRU, XGBoost, and baseline persistence) across datasets with distinct statistical properties.

Key Python Libraries:

Library	Version	Purpose
pandas	2.0+	Data manipulation and cleaning
numpy	1.24+	Numerical operations
matplotlib	3.7+	Static plotting
statsmodels	0.14+	Statistical and econometric modelling
tensorflow/keras	2.15+	GRU neural network modelling
xgboost	1.7+	Gradient boosting regression
scikit-learn	1.3+	ML utilities and evaluation

Hardware configuration:

Component	Specification
Processor	Intel i5/i7 or AMD equivalent (3.0 GHz+)
Installed RAM	8 GB minimum (16 GB recommended)
System Type	64-bit OS, x64-based processor
GPU	Optional NVIDIA CUDA-enabled GPU for deep learning
OS	Windows 10/11, macOS, or Linux

2. Prerequisites

- Environment: Python 3.10+ (Google Colab recommended for reproducibility)
- Core packages: pandas, numpy, matplotlib, statsmodels, tensorflow, xgboost, scikit-learn
- Data source:
  - Electrical: public dataset with environmental factors and zone-wise consumption for 2017.
  - Bitcoin: Bitstamp historical daily prices from 2011-2021.

3. Input Data

- Power consumption CSV (2017, daily aggregation by script).
- Bitcoin CSV: bitstampUSD_1-min_data_2012-01-01_to_2021-03-31.csv resampled to daily averages.

4. Setup

- In Colab: Runtime -> Change runtime type -> Python 3
- Install dependencies:
  pip install numpy pandas matplotlib statsmodels scikit-learn tensorflow xgboost
- Upload data to the notebook workspace or mount Google Drive.

5. Running the Notebook (section map)

1. Ingest & Clean: drop NaNs, remove unused columns, set datetime index, aggregate daily.
2. EDA: plot trends, autocorrelations, seasonal patterns.
3. Feature Engineering: lagged values, rolling averages, day-of-week indicators.
4. Model Training:
   - GRU (power consumption, univariate) with TimeseriesGenerator (lookback=16).
   - XGBoost (power consumption, multivariate features).
   - Persistence model (Bitcoin) as baseline.
5. Forecasting: recursive multi-step predictions.
6. Evaluation: RMSE, MAE, MAPE; compare actual vs predicted.
7. Visualisation: trend charts, residual plots, training/validation loss curves.

6. Expected Outputs

- Tables: model performance metrics (RMSE, MAE, MAPE).
- Figures: forecast vs actual plots, training loss curves, residual distributions.

7. Reproducibility Notes

- Ensure consistent date formats and sorted indices.
- Aggregation to daily frequency smooths high-frequency noise.
- GRU results may vary slightly depending on runtime hardware.

8. Troubleshooting

- Missing data: check CSVs and clean before processing.
- TensorFlow memory errors: reduce batch size or use CPU-only runtime.
- XGBoost import errors: reinstall package and restart kernel.
- Date parsing: ensure %Y-%m-%d format and proper index sorting.