

Time Series Forecasting using Recurrent Neural Networks

Schalk Visagie

Computer Science Department

Stellenbosch University

Stellenbosch, South Africa

25349589@sun.ac.za

Abstract—

Index Terms—

I. INTRODUCTION

II. BACKGROUND

TODO: Discussion on Elman, Jordan and Multi-RNNs

III. METHODOLOGY

A. Datasets

TODO: ensure Data Preprocessing justifications

S&P 500 ETF Daily OHLCV dataset was obtained from Yahoo Finance. This dataset comprises 5,031 trading days of Open, High, Low, Close, and Volume data from 19 September 2005 to 19 September 2025. This dataset is particularly relevant for evaluating RNN architectures due to its inherent non-stationarity, volatility shifts, and complex non-linear dependencies, which provide a robust benchmark for comparing the ability of different models to learn temporal patterns and predict next-step log returns. Preprocessing was necessary to ensure data quality and model stability. The 85 instances identified as outliers (>3 standard deviations) were clamp-transformed to 3 standard deviations from the mean. To address multicollinearity among the highly correlated OHLC features while retaining discriminatory power, the Open, High, and Low variables were dropped and replaced with two engineered features: the high-low range and log returns. The closing price, confirmed as non-stationary by Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests, was stabilized by conversion to log returns. Finally, all features were scaled to a $[0, 1]$ range independently for the training and testing sets to prevent data leakage.

VIX Daily OHLC dataset, sourced from Yahoo Finance, provides 5,031 observations of daily Open, High, Low, and Close data for the period spanning 19 September 2005 to 19 September 2025. This time series is well-suited for testing RNNs because its structural properties. These structural properties include sharp fluctuations, sudden spikes, and shifting sequential dependencies that create challenging non-stationary signals. These characteristics differ significantly from the other datasets which make it an ideal candidate for diversifying training data. Data preparation involved several steps. First,

73 outliers exceeding 3 standard deviations were clamped to 3 standard deviations. The non-stationarity of the closing price, verified with ADF and KPSS tests, was resolved by transforming the series into log returns. Similar to the S&P 500 data, the redundant and highly correlated Open, High, and Low features were removed and replaced with the engineered high-low range and log return features. The resulting feature set was then scaled to a $[0, 1]$ range separately for the training and testing partitions.

The **ElectricityLoadDiagrams20112014** dataset contains high-frequency electricity consumption readings in kilowatts for 370 clients, recorded every 15 minutes from January 2011 to the end of 2014. For this assignment, a single client's consumption profile, consisting of 140,256 measurements, was arbitrarily selected to create a univariate time-series forecasting scenario. This provides a distinct high-granularity, cyclical test case for evaluating the predictive performance of the RNN models. The data required minimal cleaning as it contained no missing values. However, power consumption values exceeding 3 standard deviations from the mean were clamp-transformed to handle outliers. Both ADF and KPSS tests confirmed that the series was non-stationary. To address this, a 24-hour seasonal differencing was applied, accounting for the inherent daily cyclicity of energy usage. During model training and evaluation, the data was scaled to a $[0, 1]$ range independently for the training and testing sets.

The **Synthetic Autoregressive Stationary (AR(1))** dataset was generated using Python code with the NumPy library, simulating a univariate time series from an AR(1) process defined by the equation $x_t = 0.5x_{t-1} + \epsilon_t$, where ϵ_t is white noise drawn from a normal distribution with mean 0 and standard deviation 1. This dataset consists of 10,000 sequential observations, indexed from time step 0 to 9,999, following a 500-point burn-in period to ensure the process reaches stationarity. It is particularly suitable for benchmarking RNN architectures in this project due to its inherent stationarity, linear autoregressive dependencies, and absence of trends or seasonality, offering a controlled environment to evaluate the models' ability to capture simple recurrent patterns without the confounding factors present in real-world data, thereby serving as a baseline for comparison with non-stationary datasets. The synthetic nature ensured no missing values or structural anomalies. Stationary was confirmed both by design (autore-

gressive coefficient $|\phi| = 0.5 < 1$) and through ADF and KPSS tests, requiring no differencing or detrending. Finally, the series was scaled to a $[0, 1]$ range independently for the training and testing sets to facilitate stable RNN training and prevent data leakage.

The **TimeSeries Weather Dataset**, sourced from Kaggle, contains hourly historical weather data for two locations. The location having more records (389,496 observations spanning January 1, 1980, to June 6, 2024) was selected. This dataset is a multivariate time series with 17 continuous features, including temperature, humidity, dew point, precipitation, pressure and cloud cover to name a few. This dataset was chosen for its high temporal granularity, pronounced daily and seasonal cycles as well as multivariate interactions. Its inclusion enhances dataset diversity by introducing time-series data with multiple cyclical tendencies and non-stationarity, contrasting the previously mentioned datasets. The data exhibited no missing values or major irregularities. Outliers exceeding 3 standard deviations were clamp-transformed. Highly correlated features with absolute correlation greater than 0.75 were dropped to reduce multicollinearity while preserving predictive power. The target feature is defined as the next hour's differenced temperature. Non-stationarity, confirmed by ADF and KPSS tests, was addressed through 24-hour seasonal differencing to account for daily periodicity. All features were scaled to a $[0, 1]$ range independently for the training and testing sets to ensure model stability and prevent data leakage.

B. Recurrent Neural Networks Implemented

TODO: describe the optimisation algorithm used
 TODO: describe the loss function used

C. Cross Validation Implementation

D. Overfitting/Underfitting Prevention

IV. EMPIRICAL PROCEDURE

V. RESULTS

VI. CONCLUSION

REFERENCES

- [1] S. Visagie, "Assignment 3 ml," <https://github.com/ThreadAgain/ml-assignment-3>, 2025.