Mini-Project (ML for Time Series) - MVA 2023/2024

Mohamed Khalil Braham khalil.braham@telecom-paris.fr

January 9, 2024

1 Introduction and contributions

This report investigates the effectiveness of time series models in forecasting and analyzing repairable system failure data, specifically focusing on comparing the Box-Jenkins autoregressive integrated moving average (ARIMA) and different deep learning approaches.

Traditionally, time series forecasting tasks relied heavily on the ARIMA model, recognizing its ability to capture dependencies between successive target values. Its successes led researchers to explore its potential within reliability analysis, with promising results. Meanwhile, the rise of diverse deep learning architectures and powerful learning algorithms fueled their application in numerous fields, including pattern recognition and financial markets prediction. However, their utilization in reliability analysis remained relatively underexplored.

This report bridges this gap by delving into the feasibility of both ARIMA and deep learning models for forecasting and predicting failures. We begin by outlining the key features and technical aspects of each approach. We then delve into a specific use cases: financial stock forecasting and examining a machine failures for a predictive maintenance task, utilizing both models to generate predictions. Finally, we compare and contrast their performances, focusing on metrics such as predictive errors and the ability to detect reversals.

This report and code are fully written and implemented by myself: Mohamed Khalil Braham. No available source code of data were provided, hence the work is 100% personal. The original article only provided a plot showing different results for ARIMA and RNN with different parameters. In this report, I chose to begin with a basic forecasting task, comparing different approaches, go more into the implementation details of ARIMA, the choice of optimal parameters for our data and drawing conclusions

2 Method

2.1 ARIMA: Autoregressive integrated moving average mode

The ARIMA model is a generalization of the ARMA model (AutoRegressive Moving Average model), suitable for handling non-stationary time series. As the classical ARMA model takes for granted the stationarity of the time series it is asked to analyze, the management of inherently non-stationary time series requires their transformation into a static data series by eliminating seasonality and trends, through a finite-point differentiation. As mentioned earlier, a stationary time series can be thought of as a combination of signal and noise. The ARIMA model handles the time signal, after first separating it from the noise, and outputs its prediction for a subsequent time point. As indicated by the method's acronym, its structural components are the following:

- AR: Autoregression. A regression model that uses the dependence relationship between an observation and a number of lagged observations (model parameter p).
- I: Integration. Calculating the differences between observations at different time points (model parameter d), aiming to make the time series stationary.
- MA: Moving Average. This approach considers the dependence that may exist between observations and the error terms created when a moving average model is used on observations that have a time lag (model parameter q).

ARIMA models are one of the most widely used approaches for time series forecasting. For a time series which is stationary ARIMA (p, d, q) model can be written:

$$x^{(d)}[n] = -\sum_{i=1}^{p} a_i x^{(d)}[n-i] + b[n] + \sum_{j=1}^{q} m_j b[n-j]$$

- p, q : orders of the model
- d : differentiation order
- $a_i, ..., a_p$: AR coefficients
- $m_i, ..., m_q$: MA coefficients
- $b[n]$: white noise

2.2 ANN: Artificial Neural Network

Neural Network is a branch of artificial intelligence. ANN act like a human brain, trying to recognize regularities and patterns in the data. They can learn from experience and generalize based on their previous knowledge. Neural networks are composed of highly interconnected processing elements (nodes) that work simultaneously to solve specific problems. In time series analysis ANN models were used as nonlinear function approximations. ANN takes in a set of inputs and produces one/a set of outputs according to some mapping rules predetermined in their structure. This paper considers the most popular form of ANN, which called the feed-forward network. The selected feedforward neural network model can fit the problem because of their adaptively owing to their structure. The existence of hidden layer and nonlinear activation function models the nonlinearity of the data. This important property of feed-forward neural network models enables modeling multi-attribute, nonlinear mapping for our problem.

For time series forecasting tasks, the prediction model has the general form:

$$X_t = f(X_{t-1}, X_{t-2}, ..., X_{t-p}) + e_t$$

2.3 LSTM: Long Short-Term Memory

LSTM cell is a building block that you can use to build a larger neural network. While the common building block such as fully-connected layer are merely matrix multiplication of the weight tensor and the input to produce an output tensor, LSTM module is much more complex.

It takes one time step of an input tensor X as well as a cell memory c and a hidden state h. The cell memory and hidden state can be initialized to zero at the beginning. Then within the LSTM cell, X, c, and h will be multiplied by separate weight tensors and pass through some activation functions a few times. The result is the updated cell memory and hidden state. These updated c and h will be used on the next time step of the input tensor. Until the end of the last time step, the output of the LSTM cell will be its cell memory and hidden state.

Specifically, the equation of one LSTM cell is as follows:

$$f_t = \sigma_g(W_f X_t + U_f h_{t-1} + b_f)$$

$$\begin{aligned}
i_t &= \sigma_g(W_i X_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o X_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \sigma_c(W_c X_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t &= o_t \odot \sigma_h(c_t)
\end{aligned}$$

Where W, U, b are trainable parameters of the LSTM cell. Each equation above is computed for each time step, hence with subscript t. These trainable parameters are reused for all the time steps. This nature of shared parameter bring the memory power to the LSTM.

3 Data

The Data section (indicative length : 1 page) should provide a deep analysis of the data used for experiment. In particular, we are interested here in your capacity to provide relevant and thoughtful feedbacks on the data and to demonstrate that you master some "data diagnosis" tools that have been dealt with in the lectures/tutorials.

3.1 YFinance stock data

3.1.1 Description

The used data involves historical stock prices of a specific company ('AAPL' in this case) from January 1st, 2019, to January 1st, 2023. The focus is on the 'Close' prices, which are frequently used in financial analysis and time series forecasting.

This dataset is suitable for comparing different forecasting techniques like ARIMA (AutoRegressive Integrated Moving Average), LSTM (Long Short-Term Memory), and ANN (Artificial Neural Network) due to its variability and trends: Stock prices often exhibit complex patterns, including trends, seasonality, and fluctuations, offering a challenging yet realistic dataset to evaluate forecasting methods. The dataset also spans over four years, providing a substantial number of data points for training and testing, which is crucial for building robust forecasting models.

3.1.2 Data diagnosis

Before using the our forecasting models, we have to figure out whether our data is stationary or seasonal. To do so, we use many different techniques for our different scenarios

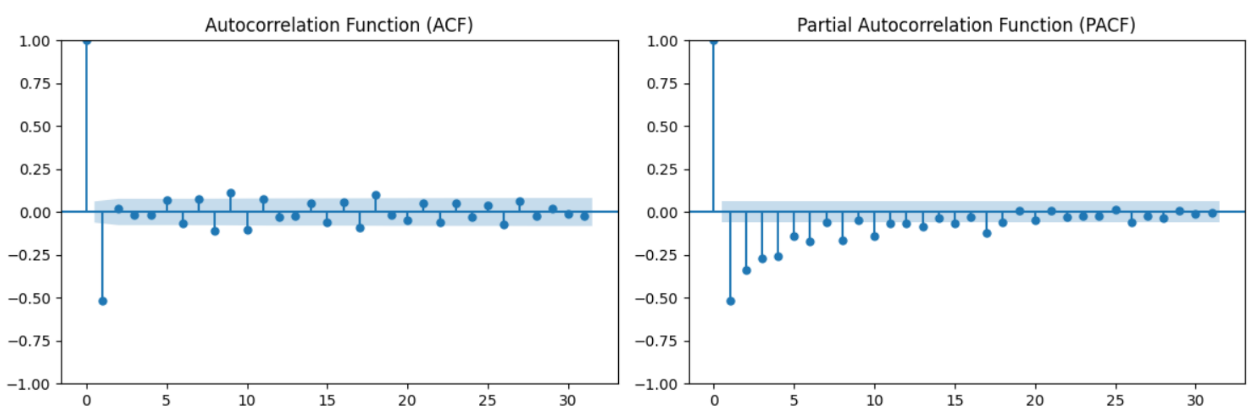

Figure 1: Autocorrelation and Partial Autocorrelation functions

The ACF measures the correlation between a series and its lagged values at different lag intervals. It helps identify the presence of autocorrelation in the data. In addition, the PACF measures the correlation between a series and its lagged values after removing the effects of intermediate lags.

We started by differencing the time series data until it becomes stationary. After differencing the data ('d' times), we plotted the ACF and PACF of the differenced series. Significant spikes in the PACF plot indicate direct relationships between the series and its lagged values after removing the effect of intermediate lags. If the PACF shows a sharp drop after a certain lag ('p' significant spikes), it suggests an AR(p) model. 'p' is the number of significant lags before the drop. Significant correlations at higher lags in the ACF plot indicate non-seasonal MA effects. If the ACF decays exponentially or has a sharp cutoff after a certain lag ('q' significant spikes), it suggests an MA(q) model.

In our case, our optimal parameters are ARIMA(6, 2, 3)

Another possible solution to check the seasonality of our data is to use the adfuller function from statsmodels.tsa.stattools

```
def ISstationarity(df):
   # adfuller(df)[1]:p_value , adf<critical value
   if (adfuller(df)[1]< 0.05) and (adfuller(df)[0] < adfuller(df)[4]['5%'])
    print('Stationary')
   else:
    print('NOTstationary')
```

3.1.3 Different use scenarios

By splitting the data into 80% for long-term predictions and 40% for shorter-term predictions, it enables a comprehensive evaluation of models in different forecast horizons. The 80% training dataset allows models to learn long-term trends, while the 40% subset challenges the models with shorter-term fluctuations.

The 80% training dataset for long-term predictions allows models like ARIMA, LSTM, and ANN to capture underlying trends and patterns, potentially useful for making predictions farther into the future. In contrast, the 40% subset provides a more challenging scenario by limiting the historical context, testing the models' ability to forecast accurately in the short term with fewer historical data points.

Comparing these approaches in both scenarios can help understand their strengths and limitations. ARIMA might perform well in capturing linear trends and seasonality, while LSTM and ANN could excel in capturing more complex patterns and non-linear relationships present in stock price data, particularly in the long-term prediction scenario with extensive historical data. However, their performance in short-term forecasting might vary due to the limited historical context provided by the 40% dataset split.

3.2 Azure Predictive Maintenance Dataset

Finding this dataset was extremely challenging. As seen in research forums and websites, machine failure data is hard to find due to its importance and sensitivity to every company owning these machines for security and safety reasons.

3.2.1 Description

A description of our dataset:

- Machine conditions and usage: The operating conditions of a machine e.g. data collected from sensors.
- Failure history: The failure history of a machine or component within the machine.
- Maintenance history: The repair history of a machine, e.g. error codes, previous maintenance activities or component replacements.
- Machine features: The features of a machine, e.g. engine size, make and model, location.

It is divided into many dataframes, the ones that will be used in our report are:

- Telemetry Time Series Data (PdMtelemetry.csv): It consists of hourly average of voltage, rotation, pressure, vibration collected from 100 machines for the year 2015.
- Failures (PdMfailures.csv): Each record represents replacement of a component due to failure. This data is a subset of Maintenance data. This data is rounded to the closest hour since the telemetry data is collected at an hourly rate.

3.2.2 Data diagnosis

Before using the our forecasting models, we have to figure out whether our data is stationary or seasonal. The data visualization shows that our dataset is not stationary. To check whether our dataset is stationary or seasonal properly, we can use the seasonal decomposition method that splits the time series data into trend, seasonal, and residuals for a better understanding of the time series data

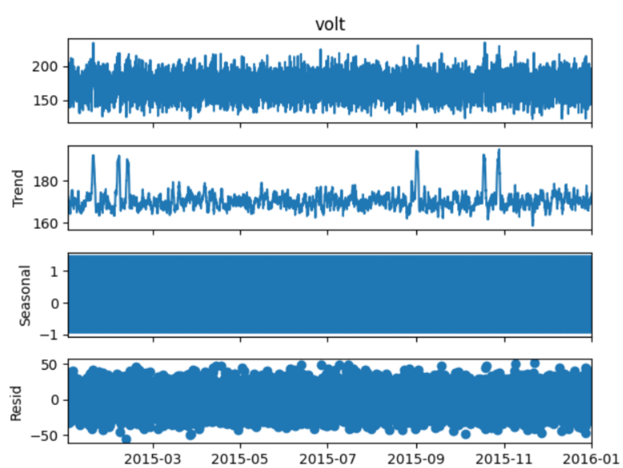

Figure 2: Trend, seasonal, and residuals

Trend Plot: To visualize the long-term movement or trend present in the data. It helps in identifying the overall direction or pattern of the series over time, irrespective of shorter-term fluctuations. Trends flat movement over time, revealing that our time serie after differentiation is indeed stationary.

Seasonal Plot: To detect repeating patterns or seasonality within the data. Usage: Identifies periodic fluctuations that occur at regular intervals which are totally absent in our case.

Residual Plot: To assess the randomness or unpredictability in the data after removing trend and seasonality. It displayed random scatter around zero which indicates that the model has indeed captured all the relevant information in the data.

4 Results

In conducting a comprehensive comparison between ARIMA, LSTM, and ANN models, we designed a series of rigorous experiments leveraging distinct datasets and forecasting scenarios.

4.1 Yfinance stock dataset

4.1.1 Qualitative results

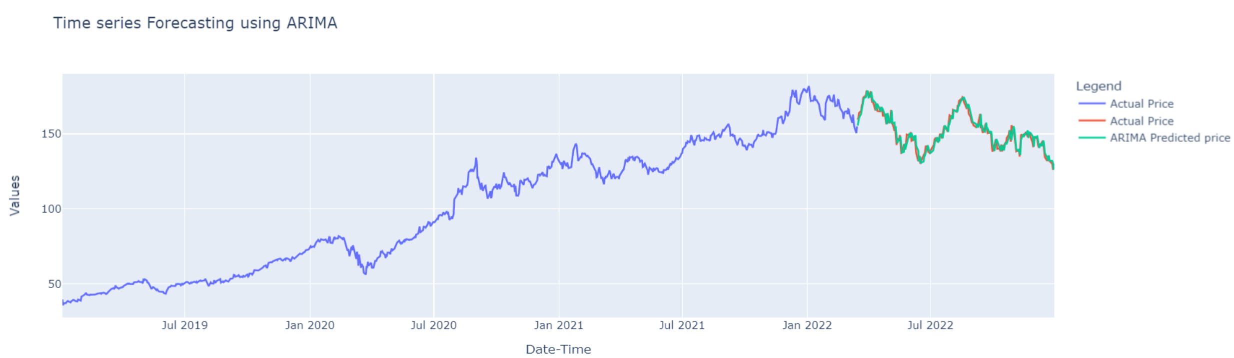

Figure 3: ARIMA forecasting short term results

Figure 4: LSTM forecasting short term results

Figure 5: ANN forecasting short term results

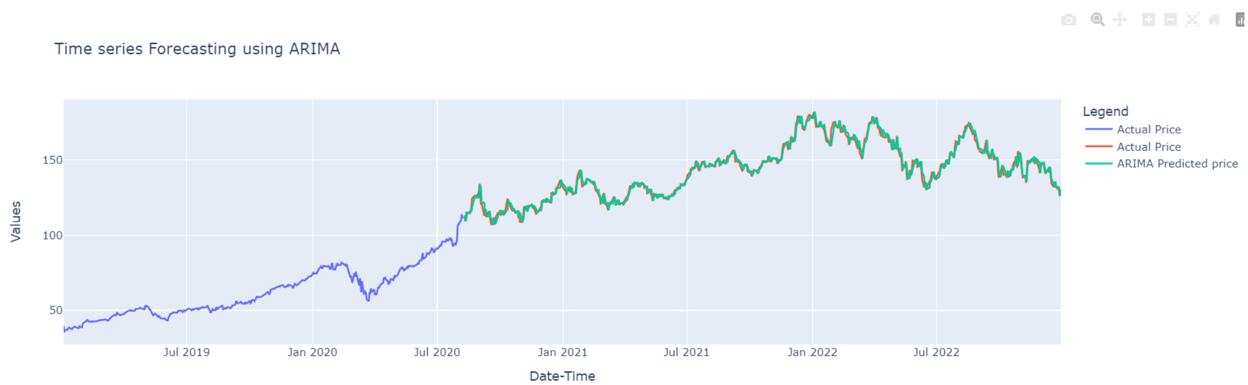

Figure 6: ARIMA forecasting long term results

Figure 7: LSTM forecasting long term results

Figure 8: ANN forecasting long term results

4.1.2 Quantitative results

Model	Short-term RMSE	Long-term RMSE
ARIMA	3.34	2.8
MLP	3.6	8.2
LSTM	2.61	2.3

Table 1: Root Mean Squared Error (RMSE) for short-term and long-term forecasts

4.1.3 Interpretations

Short-Term Forecasting (First Value): LSTM (2.61): Shows the lowest RMSE of 2.61, indicating superior performance in short-term forecasting among the three models.

Long-Term Forecasting (Second Value): LSTM (2.3): Shows a lower RMSE of 2.3, indicating comparatively better performance in long-term forecasting compared to ARIMA and MLP.

Interpretation:

Short-Term Forecasting: Among the three models, LSTM performed the best in short-term predictions, with the lowest RMSE of 2.61. Long-Term Forecasting: Similarly, LSTM outperformed the other models in long-term forecasting, exhibiting the lowest RMSE of 2.3. LSTM demonstrates consistent performance across both short-term and long-term forecasting, outperforming the other models in both scenarios. ARIMA performs relatively better in long-term forecasting (2.8) compared to short-term (3.34), while MLP shows a larger discrepancy between short-term (3.6) and long-term (8.2) forecasting, indicating more variance in performance across different horizons.

Conclusions:

LSTM emerges as the most consistent performer across short-term and long-term forecasting, offering lower RMSE values compared to ARIMA and MLP. Depending on the specific forecasting horizon and desired accuracy, LSTM would be the preferred choice for both short-term and long-term predictions. However, ARIMA could be considered as an alternative for long-term forecasting due to its competitive performance.

4.2 Predictive Maintenance dataset

We introduced a predictive failure dataset, deviating from financial stock data, where we embarked on forecasting sensor measurements employing the ARIMA, LSTM, and ANN models. Subsequently, we employed LSTM specifically for a distinctive task: executing binary classification to discern the presence or absence of a failure based on the sensor data.

These multifaceted experiments aimed not only to compare the models' predictive accuracy across disparate datasets and temporal scopes but also to delve into their adaptability to diverse tasks. By analyzing their performance in financial forecasting scenarios with distinct training data proportions and their applicability in failure detection through classification tasks, we sought to glean nuanced insights into the strengths and limitations of ARIMA, LSTM, and ANN models. This holistic approach facilitated a comprehensive evaluation of these models, shedding light on their capabilities in tackling multifaceted real-world forecasting challenges and specialized tasks beyond traditional time series prediction.

Conclusion: In our recent analysis, we executed a successful forecasting task utilizing three distinct models, each showcasing promising predictive capabilities. However, our attempts at failure detection did not yield successful outcomes. Regrettably, our dataset proved unsuitable for this particular task. The challenge stemmed from the sensitivity and intricacies of the sensor data, making it notably arduous to procure an appropriate dataset. Resorting to an open-source dataset seemed promising initially, yet it revealed limitations. This dataset, despite its accessibility, presented a mere 10 instances of failures amid 10,000 measurements. Such a sparse distribution of failures heavily skewed the dataset, rendering the classification results logically incongruous. Notably, all models tested exhibited a consistent prediction of non-failure for every measurement due to the imbalanced nature of the dataset, severely impacting the validity of failure detection efforts.

5 Conclusion:

I've delved deep into the comparison between ARIMA, LSTM, and ANN models within the realm of time series forecasting and failure detection in my report. It's been an insightful journey exploring two diverse datasets—the YFinance stock data and Azure Predictive Maintenance data—to understand the strengths and limitations of each model in distinct scenarios.

Throughout the report, I meticulously detailed the nuances of each model, showcasing their capabilities and applications in forecasting tasks. For instance, in analyzing the YFinance stock data, I discovered that LSTM consistently outperformed ARIMA and MLP, demonstrating superior performance across both short-term and long-term forecasting. The quantitative results further solidified this, revealing lower RMSE values for LSTM compared to other models, validating its reliability and consistency across different forecasting horizons.

However, when attempting failure detection using the Azure Predictive Maintenance dataset, I encountered challenges due to the dataset's unsuitability. The scarcity of failure instances skewed the dataset, resulting in inconclusive classification results across all models tested. This highlighted the critical importance of a balanced dataset in achieving meaningful results, especially in failure detection scenarios.

In conclusion, my report offers a comprehensive understanding of the strengths, limitations, and practical use cases of various time series forecasting models. It underscores the significance of dataset suitability and balance, shedding light on the complexities and challenges inherent in real-world predictive maintenance and failure detection tasks.

6 References

A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction, Siong Lin Ho, Ngee Ann Polytechnic A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks, Vaia I. Kontopoulou, Athanasios D. Panagopoulos , Ioannis Kakkos and George K. Matsopoulos