AUTOMATISATION HYPERPARAMETERS TUNING PROCESS FOR TIMES SERIES FORECASTING: APPLICATION TO PASSENGER'S FLOW PREDICTION ON A RAILWAY NETWORK

Qamar EL MAAZOUZI, Asmaa RETBI, and Samir BENNANI

Rime Team-Networking, Modeling and e-Learning Team- Masi Laboratory-
Engi-neering.3S Research Center-, Mohammadia School of Engineers (EMI),
Mohammed V University in Rabat, Morocco,
qamarelmaazouzii65@gmail.com, retbi@emi.ac.ma, sbennani@emi.ac.ma

KEYWORDS: Machine Learning, LSTM Architectures, Times Series, Hyperparameters optimization, Walk-Forward Optimization, Flow Passengers, Railway

ABSTRACT:

Many industries and companies in various fields are interested in time series analysis to predict the future. However, in time series modeling, precision is lacking as time progress. In this paper, an architecture is proposed, allowing on the one hand keep the prediction accurate over time using the Walk Forward Optimization (WFO); On the other hand, automate the choice of parameters of the statistical models (ARIMA) introducing "AutoArima"; The RNN models, especially LSTM architectures (LSTM, Bi-LSTM, Stacked LSTM) using the function Optuna. Moreover, to avoid overfitting the LSTM models, an automatic function is implemented in the presented architecture. To demonstrate the validity of this research, a comparison of three models applied to a railway company to predict the flow of passengers is made. In particular, the naive model constitutes a reference base, the ARIMA model which had demonstrated its performances in several research, and finally, following the last progress in the neural networks the LSTM architecture is introduced in the paper. According to the results, the implemented architecture has great potential and more accurate predictions by using WFO. Through the comparisons of the models made, Each model has proven its performance according to the case of study. More concretely the mean absolute error obtained by LSTM for the railway stations is 0,13 compared to 0,15 obtained by ARIMA and 0,16 for the naive model, showing a small superiority for LSTM over ARIMA. On the other side, ARIMA excels on the Train lines.

1. INTRODUCTION

Throughout these years, several analyses and predictions of chronological series have been carried out and this in several fields (Kim & Moon, 2019). Following the current circumstances of the Covid pandemic, we have chosen to dig deeper into the prediction of the flow of passengers. Indeed, several transport companies, and in particular the railway companies, have been obliged to apply the measures dictated by the authorities. To this end, the railway companies have put in place a whole set of protection and sanitary security measures. Among the measures taken, is the respect for social distancing on the platforms/railways and on board the trains. We have therefore through this article calculated the prediction of passenger flows in the future allowing the operational staff of the railway, on the one hand, to make optimal planning of the number of trains, vehicles, and personnel resources to meet the requirements of limiting the filling of trains. On the other hand, and with the progressive resumption of the normal state of the circulation of the trains the planners will be capable to anticipate better planning. This prediction will have an impact on the reduction of costs and an improvement in turnover. Moreover, the company will be able to inform its customers in real-time on the forecasted situation of the train's filling, to allow them to better organize their trips. The architecture presented in this article is designed to have a more accurate and reliable prediction. To investigate whether the architecture proposed to improve the prediction, this paper explores the accuracy and performance of the flow passenger prediction in a railway company. To do so, we will report the results of a comparison of the RNN-Model, and Statistical model by using walk-forward optimization (WFO) and without WFO. Also, two methods to determine automatically the optimum parameters of ARIMA (Auto Arima) and the hyperparameters of LSTM (Optuna) were presented. In this paper, we present the importance of data preparation and visual graph to be able to detect the different components of the time series and to have accurate and clean data. And finally, the article demonstrates the interest in proposing a baseline to determine the choice of models in our case it will be the naive model.

In particular, the answers to the following questions are presented :

1. Can we have a model with more accuracy and performance by implementing the walk-forward optimization?
2. Will we have the same results to define the ARIMA Parameters by using Auto Arima rather than using the graph plots (ACF, PACF)?
3. Can we solve the dilemma of hyperparameters LSTM If we use an automatic function like optuna?
4. Can we avoid the overfitting of the LSTM model by using an automatic function?
5. Can we find the optimal model to predict the following passengers for a railway company in station /line?

To answer these questions, we address a set of comparisons and propose an architecture with functions and automated process.
The paper presents and proposes the following contributions:

- New architecture with a more accurate traveler flow prediction and with better performances.
- Automatic functions implemented in the proposed architecture to find a solution for the optimal choice of the ARIMA and LSTM parameters.
- Automatic process to avoid overfitting the LSTM model.
- The steps to be implemented for the data preparation.
- new guideline for the choice of models using baseline in our case it will be a naive approach.
- analysis comparing the flow passenger's prediction accuracy of the ARIMA, Unit-LSTM, Stacked LSTM, and BI-LSTM.

This paper is structured as follows: Section 2 reviews the related works. The essential background and theoretical explanation of the used models, and functions are given in section 3. Section 4 conducts a case study with real data from the Railway compagnie. The results are also presented in this section. Section 5 discusses the results. The conclusion of the paper and the possible future research directions are given in section 6.

2. RELATED WORKS:

Several research has been conducted to predict passenger flow in different transportation companies.
Many models were compared by Wang et al.(2021) such as (ARIMA,LIGHT GBM..). He also proposes a line to follow in the preparation of data before going to the prediction stage. He eventually proved the performance of LIGHT GBM applied to high-speed trains, unlike ARIMA which give poor results. Contrary to research that dates by Jiaotong & Jiaotong (2013) proving the accuracy of the model ARIMA combined with RBF neural network model applied on urban rail transit. In another hand, Ding et al.(2016) contributes with a GBDT model that proves to have better performances on short-term subway ridership prediction applied to a multimodal public transportation system. However, the performance of the GBDT model for short-term prediction has been approved by Pasini et al. (2019) adopted on a railway line operated by SNCF. In the same logic, and in order to predict short-term passenger flow, the author Zhao & Mi (2019) proposed a new hybrid model SSA-WPDCNN-SVR demonstrating accurate forecasting on the HSR line. On the other way, the application of deep learning models in real contexts can be difficult. That's why Li et al. (2022) highlighted the complexity of DL models proposed in several pieces of research. He proposed a simplified Graph-GAN model which can be applied to real cases. Most of the research mentioned below has been applied to the train lines. However, the research done by Nar & Arslankaya (2022) has been applied both on lines and stations, using the Regression models for lines while for stations Random Forest (RF) and support vector regression (SVR) were applied. In the same context, several experiments were conducted by Liu & Chen (2017), using the SAE DNN model on different BRT stations proving a better result obtained than DNN.Do et al. (2020) evaluated the performance of the LSTM model by comparing it with SARIMA for the prediction of traveller numbers at the airport. He concludes that Arima has the advantage of being a simple model, unlike LSTM which is quite complicated, but LSTM still more accurate.
All these research that has been conducted for the prediction of the passenger flow by comparing several statistical models, RNN and many others have been able to demonstrate their performances. However, the problem of the choice of the model by using a base line has not been evoked. Moreover, the automation for the choice of the optimal parameters in particular for the case of the LSTM architectures was not quoted

3. BACKGROUND

This section reviews the methods used: classical/Statistical methods such as the Naive model and autoregressive Integrated Moving average (ARIMA), also Neural networks with multiple LSTM architectures such as Unit-LSTM, Stacked LSTM, and bidirectional LSTM.
Finally, we briefly introduce 2 automated processes used in this article, the first one is the model evaluation selection Criteria: Walk Forward Optimization (WFO). And the second one is automated hypermeters optimization for LSTM known by Optuna.

3.1 Naïve model

This model assumes that the forecasted value for day k is the actual value for day k-1.
The model is based on historical observations, and it's used to set up a Baseline for the forecast.

3.2 ARIMA Model

The ARIMA model includes the autoregressive (AR) model, moving average (MA) model, and seasonal autoregressive integrated moving average (SARIMA) model (Reza & Baroumand, 2021).
ARIMA is composed of three parameters, a standard notation would be ARIMA with p: defined as the number of lags observed, d: known as the degree of differencing, and q: defined as the size of the moving average (A., & Zine-Dine, K., 2022)
The PACF graph is plotted to determine the value of the p, while to predict the value of q the ACF graph is used (Maurya, 2021).
Even though we can estimate a range of p, d, and q values through the plots we can use an automated process throw the Auto Arima in python.

3.3 LSTM Architectures

LSTM is a special kind of recurrent neural network (Malhotra et al., 2015) capable as demonstrated in many studies of handling long-term dependencies (Ergen, 2020).
this paper describes 3 LSTM architectures:

- A vanilla LSTM model: an LSTM model that has a single hidden layer of LSTM units (W. Li et al., 2021).
- A stacked LSTM model is when multiple hidden LSTM layers can be stacked one on top of another (Pattana-anake et al., 2022).
- A bidirectional LSTM model allows a traditional LSTM model to learn the input sequence both forward and backwards and concatenate both interpretations (Wahyono, 2020).

3.4 Automated process

3.4.1 Walk Forward validation/optimization: WFO was suggested by Robert Pardo (Pardo,1992). WFO is used to determine the optimal parameters by splitting data into multiple in-sample and out-of-sample segments.
The figure 1 shows a simplified concept of how work walk-forward optimization.

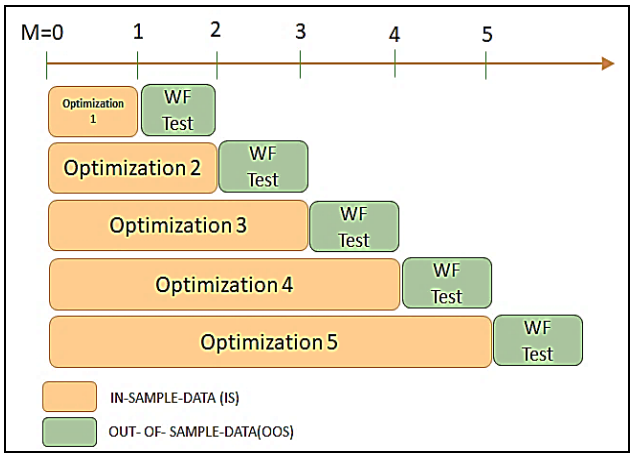


Figure 1. Walk Forward Optimization.

3.4.2 Optuna: The most challenging part of preparing and training a recurrent neural network (RNN) and especially LSTM, is the hyperparameter dilemma.
Many automatic hyperparameter tunning was developed to figure out a solution to this problem such opt (Bergstra et al.,2015), Autotune (Koch et al., 2018).
In this study Optuna is used : an open-source optimization software (Akiba et al., 2019).

4. CASE STUDIES

4.1 Data Set

The study focuses on three types of rail lines and three major stations. The dataset is derived from several data sources that constitute an integrated platform of rail operations management. The dataset consists of the set of train schedules and train runs that come from the train planning system, the forecasted number of seats in each category of the train, the number of daily sales that come from the sales system. This information was used to calculate the train occupancy rate and the number of passengers at the station level. The dataset covers the year 2021.

4.2 Data preparation

4.2.1 Structure data and Data pre-processing: As shown in figure 2, after structuring the data and obtain a cohesive data pool, we moved on to data pre-processing. In this step, we had to handle null values on the 'number of seats sold' column quite carefully. We got to achieving this task by applying the forward fill method, which simply suggests replacing a null value with the last non-null value in the series. As shown in figure 2, after structuring the data and obtain a cohesive data pool, we moved on to data pre-processing. In this step, we had to handle null values on the 'number of seats sold' column quite carefully. We got to achieving this task by applying the forward fill method, which simply suggests replacing a null value with the last non-null value in the series. After a daily down-sampling was applied to the data and went from 27176 entries to 365 by relying on the mean value of the number of seats sold each day. After the down-sampling procedure, a new problem emerges which is missing values regarding specific dates, it's fixed by interpolating using a linear method.

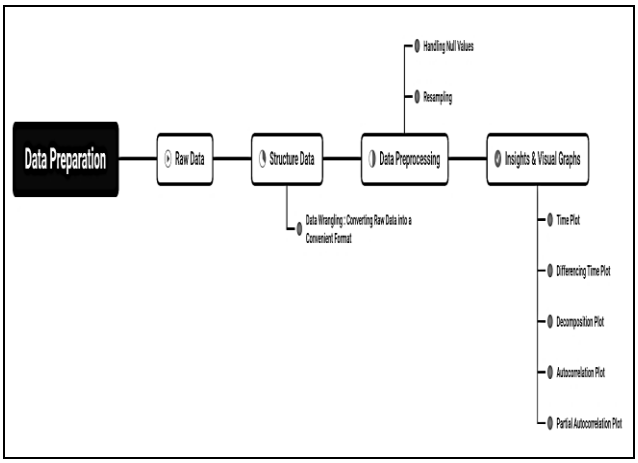


Figure 2. Steps of Data Preparation

4.2.2 Insights & Visual Graphs: Times series graphs allow for visualization of the behaviours and patterns of the data, such as seasonality, trend, and noise. On other hand, The ACF, and PACF plots can determine the parameters p,q of Arima.

- Time Plot:

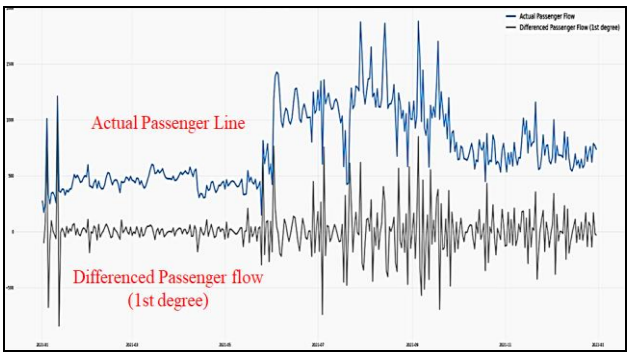


Figure 3. Time Plots daily/Differencing passenger's flow

Plotting the number of seats sold during the year 2021 (Figure 3), doesn't show any seasonality, any repeated patterns throughout the year. But there is a certain trend in it, especially during the summer.
To remove said trend, an order 1 differencing is applied to the number of seats sold column, which is basically the difference between the actual value and its lag value of the 1st order.
According to the plot as shown in figure 3, order 1 differencing gets the job done, which helps to determine the 'q' parameter's value for ARIMA, q = 1.

- Decomposing plot:

Decomposition plots show the data as is followed by its trend, then its seasonality, and the residuals (noise). These plots are generated either by using an additive or multiplicative model (Nwogu et al., 2019).

According to the previous plot, there is a combination of both exponential and non-exponential increases over time, so we plotted both.
We chose the best fitting one by analysing the residuals plot as shown in figure 4. Going with the most random between the two, and the closer it is to a white noise. In this case, the additive model is the one.

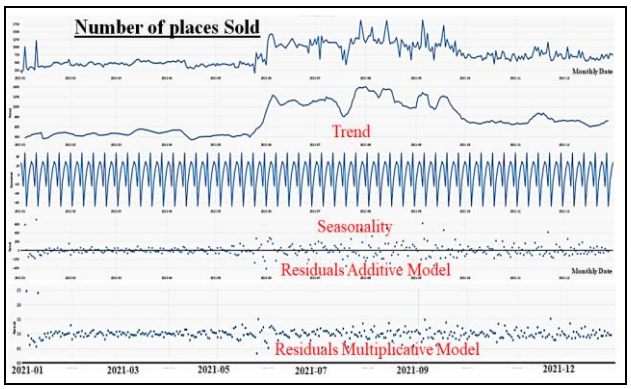


Figure 4. Decomposing Plots

- Autocorrelation (ACF)/Partial (PACF)Plot

The plot in figure 5 (a) is generated using the autocorrelation function ACF, this plot shows that there is a positive correlation between lag values up until lag 82 when it turns negative until about lag 260 when it neutralizes. The plot also demonstrates the relevant lags for the parameter 'q' for the ARIMA model, in this case, are lags 1 to 28, 28 since it is the last lag value outside the 95% confidence cone, while the correlation falling inside this cone is considered statistically insignificant.

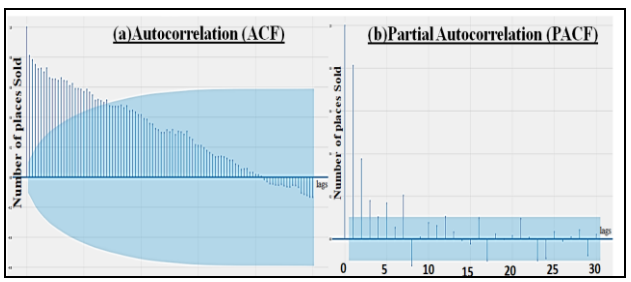


Figure 5. ACF and PACF plots for parameters ARIMA

On the other hand, another plot is generated by using the partial autocorrelation function PACF as shown in figure 5 (b). This plot is used to help determine the 'p' value in the ARIMA model. It can go up to 3, which is the last partial autocorrelation value that falls outside of the significance shading of the 95% confidence cone.

4.3 Modelling

The goal is to predict the passenger flow of the three types of trains and 3 stations, based on historical data for the year 2021.

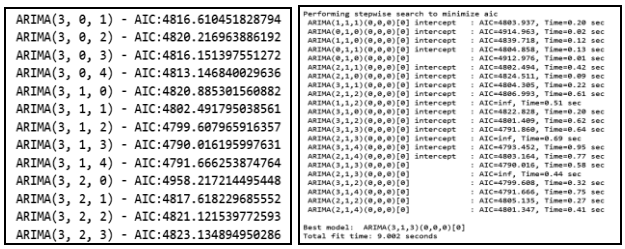
4.3.1 Model 1: Naïve Approach-Persistence model: To set up a baseline for the model's forecasts, we resorted to creating a naïve forecasting model.
With the data previously prepared, we split it into training and testing data following an 80% 20% percentage.
The model was then made in a way that the forecasted value for day k is the actual value for day k-1.

In the end, the mean squared error value is calculated, this value is the benchmark that the other models' mean the squared error will be compared against. Any model performing less than said value will be either immediately scraped or modified to perform better.

4.3.2 Model 2: ARIMA: To keep this part concise, the detailed interpretations and explanations will be focused on the High-Speed Railway, the part dedicated to the two other types of trains, and stations will be brief in comparison.

- Arima Parameters:

Even though a range of p, d, and q values estimate, through the plots, the goal in this article is to show that there are more reliable automatic methods. We first run a Python script to optimize these parameters so that we can minimize the AIC estimator the best, which is the best way to select the model pre-emptively without relying on a validation or test set. According to figure 6 (a), the best ARIMA parameters are (3,1,3).



Python Script (a) | Auto Arima (b)

Figure 6. Auto Minimize AIC & Auto Arima

Afterward, we ran an automated process done through the Auto Arima function in python, which further proves our deductions as shown in figure 6 (b).

- Walk-Forwarded Optimization

So after having figured out the best combination for the ARIMA model, we trained and fit the model on our training set and then used the walk-forward validation throughout each entry in our testing set to better the performance of the model.
To prove the effectiveness of the WFO, we compare the results obtained with the WFO and without (Figure 7).
We can deduce that the WFO allows to keep the results up to date, while if we do not use the WFO process the model becomes obsolete after a while, as it basically predicts the mean value of entry points of training.

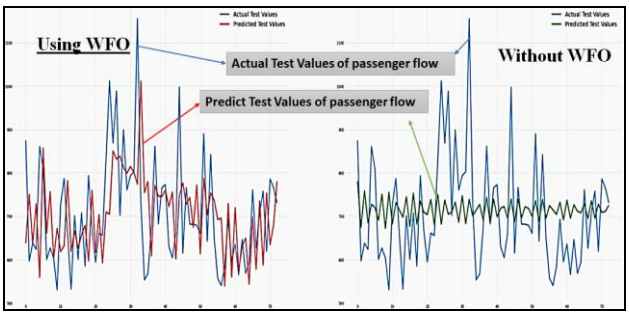


Figure 7. Results ARIMA with WFO and without

- Results

After building the ARIMA model with WFO, the diagnostics run in the results.
The residuals plot of the ARIMA model as visualized in figure 8 shows no trend nor seasonality as it should resemble a white noise's plot as much as possible, otherwise, it would mean that the model wasn't properly configured. Additionally the residuals seem to have a constant variance since the mean is close to zero and is normally distributed.
The density plot suggests normal distribution with a mean equal to zero as demonstrated in figure 8.

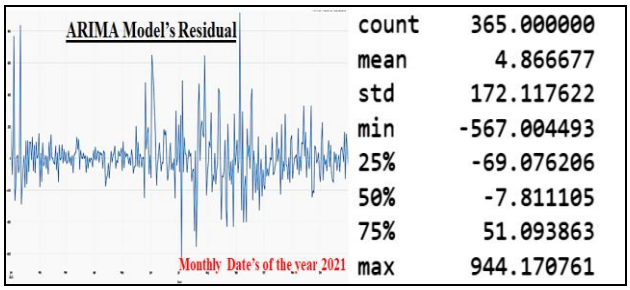


Figure 8. Diagnostics on ARIMA Results

As we can see in fig 9(a) all the dots fall perfectly in line with the red line. Any significant deviations would imply the distribution is skewed.
The Correlogram, aka, ACF plot visualized in figure 9(b) shows the residual errors are not autocorrelated. Any autocorrelation would imply that there is some pattern in the residual errors which are not explained in the model.

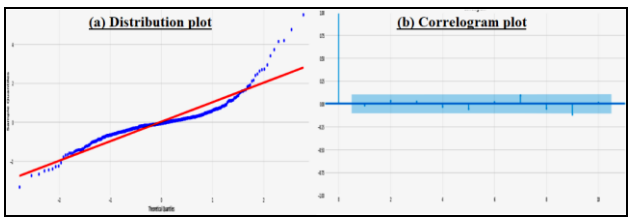


Figure 9. Distribution/Correlogram Plots

Finally, we conclude that the forecasted and actual values are better visualized through the following plot (fig 10). Which shows that the model performs better than the persistence model with a mean squared error equal to MSE = 15947.997699060277.

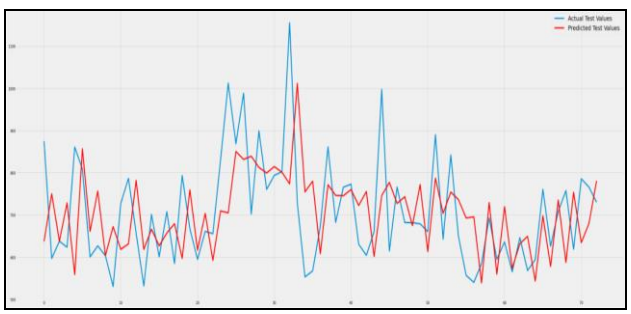


Figure 10. Forecasted Values with ARIMA

4.3.3 Model 3: LSTM Architectures

When modelling deep learning methods, in this case LSTM model, additional pre-processing must be carried out. These transformations can be broken down into three steps as demonstrated in figure 11.

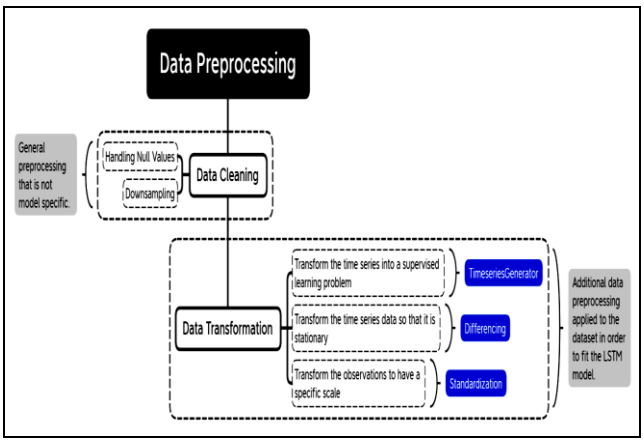


Figure 11. LSTM Data Pre-Processing

In this article, we propose a new architecture based on WFO and optuna as we can see in figure 12.

- Vanilla Model

As we already explained the most challenging part of preparing and training an LSTM, is the hyperparameter dilemma. At first, we tried, in the case of the vanilla model, out different LSTM units' possibilities, fit the model, forecast, compute the evaluation metrics, and then make changes accordingly but soon enough the process was deemed unsuccessful. That's why the procedure is automated using Optuna.
Through the optuna study 100 trials are carried out, using for each trial different combinations of hyperparameters and this during 30 epochs.
To avoid overfitting the model, the process was also automated using the early stop property provided by the Keras API.
Early stopping requires that a validation dataset is evaluated during training, in order to stop training once the model performance stops improving on the holdout validation dataset. So, we got to splitting the trainings set into train and validation data.
After this study that lasts per average of about one hour and a half, the parameters optimized architecture are the following:
- 6 LSTM units on the only LSTM hidden layer, with a mean squared error equal to MSE=16544.00020312371.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 6)                 192

 dense (Dense)               (None, 1)                 7

=================================================================
Total params: 199
Trainable params: 199
Non-trainable params: 0
_________________________________________________________________
```

Figure 13. Hyperparameters Vanilla Model

The MSE obtained using the vanilla LSTM is relatively accurate, since the model performs better than the persistence model but worse than the ARIMA model.
Given that Optuna found the best hyperparameter possible for the vanilla model, the next logical step is to stack the LSTM to improve performance.

- Stacked LSTM

The same steps were followed when modeling and optimizing the stacked LSTM model, we get the following optimized architecture:
- 30 LSTM units on the second LSTM hidden layer, with a mean squared error equal to MSE=16502.527639498807.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 3, 3)              60

 lstm_1 (LSTM)               (None, 30)                4080

 dense (Dense)               (None, 1)                 31

=================================================================
Total params: 4,171
Trainable params: 4,171
Non-trainable params: 0
_________________________________________________________________
```

Figure 14. Hyperparameters Stacked LSTM

- Bidirectional LSTM

To model and tune the Bidirectional LSTM model, the same procedures is followed, and the result is the following:
- 2 LSTM hidden layers. 8 LSTM units on the first bidirectional LSTM hidden layer, 37 LSTM units on the second bidirectional LSTM hidden layer, with an MSE= 16219.839844891496.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 3, 3)              60

 lstm_1 (LSTM)               (None, 30)                4080

 dense (Dense)               (None, 1)                 31

=================================================================
Total params: 4,171
Trainable params: 4,171
Non-trainable params: 0
_________________________________________________________________
```

Figure 15. Hyperparameters Bidirectional LSTM

- Bi-LSTM: Walk Forward Optimization

As was previously established during the Arima modelling phase, all the results acquired were the result of predictions made using the walk-forward validation methodology to renew the model each step of the way.

To further prove this point, we used the bidirectional model to predict on the testing set using both approaches. The results show that predictions obtained via WFO were much more reliable in comparison with the results calculated without using WFO, which were only slightly better than the naïve forecast model, as shown in Table 1 below:

Model	MSE	MAE	MAPE
Naïve Model	26117.998584	126.214037	0.175015
Bi-LSTM with WFO	16219.839844	102.240501	0.143472
Bi-LSTM without WF0	24613.758603	123.710606	0.173945

Table 1. Comparison performances LSTM architectures with WFV and without

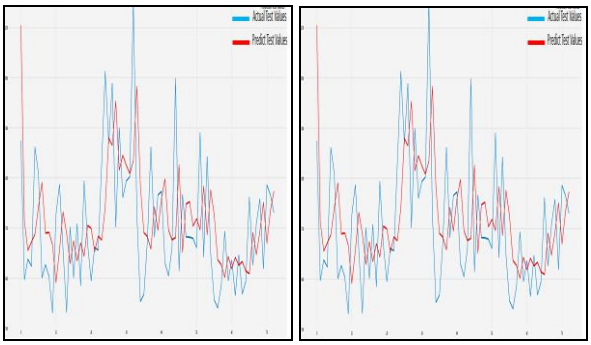
- Models Comparisons:

In this section all the experiments sum up, first, the evaluation metrics draw up in the table to measure performance for the 3 LSTM architectures (table 2):

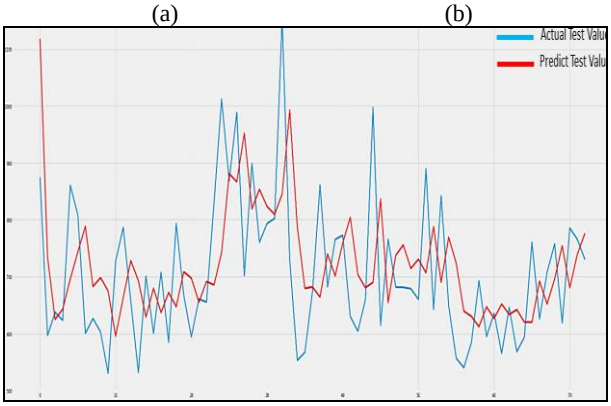
Model	MSE	MAE	MAPE
Vanilla LSTM	16544.000203	104.460558	0.146913
Stacked LSTM	16502.527639	105.457011	0.148236
Bidirectional LSTM	16219.839844	102.240501	0.143472

Table 2. Comparison performances LSTM architectures

After these comparisons, the conclusion that can be drawn is that the 3 LSTM models have very similar performances, with a slight superiority of the bidirectional model. It's used to model the other categories of trains and stations.
After that, we draw up in the table a comparative result of the models applied in the article that it is at the level of the lines or at the level of the stations (table 3). We note that ARIMA is well-performing to predict the flow of passengers at the level of the lines of the trains, contrary to BI-LSTM which excels in the prediction of the flow of passengers at the level of the stations that it is in the medium or in the long term. The figure 16 below demonstrate the visualization of prediction flow passenger with architectures LSTM on line's train.



Vanilla Model (a) Stacked LSTM (b)



Biderctional LSTM (c)

Figure 16. Prediction flow passenger in line with architectures LSTM

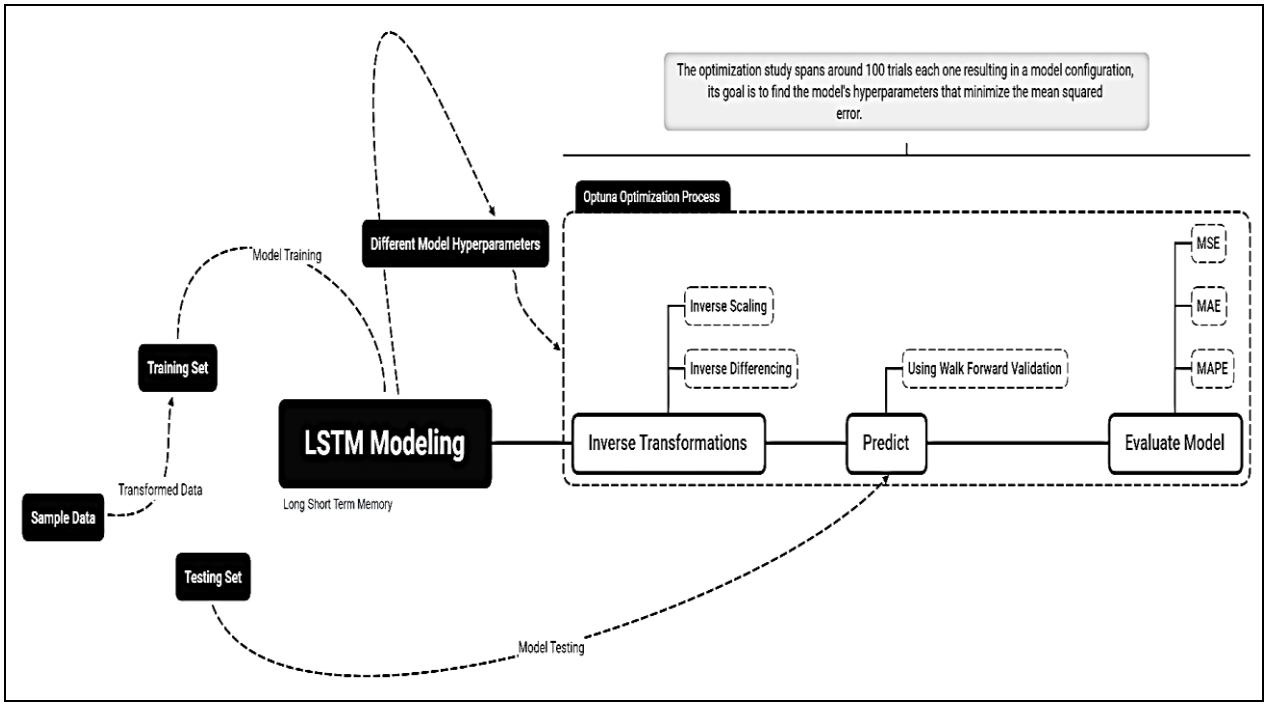


Figure 12. LSTM Architecture

Model	MSE		MAE		MAPE	
	Line	Station	Line	Station	Line	Station
Naïve Model	26117.998584	4374.066912	126.214037	49.685849	0.175015	0.165780
ARIMA	15930.737899	3065.842675	98.3166983	43.584570	0.137449	0.143913
Bidirectional LSTM	16219.839844	2500.335079	102.240501	40.808353	0.143472	0.137560

Table 3. Comparison of model's performances on lines/Stations

5. DISCUSSION

We have proposed new architecture with an approach on the one hand to automate the optimization process with the walk forward optimization (WFO), and other hand automate the choice to find the best ARIMA parameters using the Auto ARIMA function and automate the process LSTM Hyperparameters tunning introducing Optuna. ARIMA function and automate the process LSTM Hyperparameters tunning introducing Optuna..

In this work, we demonstrate the limits of the traditional way of the choice of the parameters for LSTM and ARIMA, we prove that forecasting the flow of passengers could be improved by using the architecture proposed in this article

6. CONCLUSION

The proposed architecture has great potential. We have demonstrated through the studies conducted the efficiency of the framework.

The empirical conclusions are detailed below:

(1) A new approach for model selection by defining a baseline for predictions (NAÏVE MODEL) allowing to have a clear guideline for the models to be recommended. (2) With WFO the parameters are likely to be much more effective. We have shown that the parameters are more robust and confident that overfitting has not occurred to predict the flow of passengers. (3) Instead of considering every possible combination to define p and q by using ARIMA graphs (ACF, PACF), the Auto Arima function allows having the optimal parameters. (4) in the same approach, the effectiveness of the automation of hyperparameters LSTM is demonstrated.The function called 'optuna' has brought an answer to solve the hyperparameters dielelama and has allowed having an automatic model with a forecast of the flow of the trip more reliable. (5) The different steps for the preparation and visualization of the data are defined. The proposed approach allowed us to visualize the components of time series (trend, seasonality, stationarity) and to ensure the quality of the data. This proposal allowed to understand the relevance of data preparation for a more reliable time series forecasting. (6) By comparing the used models in this paper, ARIMA shows excellent forecasting for prediction for train types, but performance degrades for predicting passenger flow at train stations let LSTM have a better MSE.

Finally we conclude with perspectives to extend this study to other cases to predict time series. Moreover, the proposed architecture can be further improved by using knowledge graphs.

REFERENCES

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). *Optuna : A Next-generation Hyperparameter Optimization Framework*. 2623–2631. https://doi.org/10.48550/arXiv.1907.10902

Rguibi MA, Moussa N, Madani A, Aaroud A, Zine-Dine K. *Forecasting Covid-19 Transmission with ARIMA and LSTM Techniques in Morocco. SN Comput Sci*. 2022;3(2):133. https://doi:10.1007/s42979-022-01019-x

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. D. (2015). *Hyperopt : a Python library for model selection and hyperparameter optimization Hyperopt : a Python library for model selection and hyperparameter optimization*. https://doi.org/10.1088/1749-4699/8/1/014008

Ding, C., Wang, D., Ma, X., & Li, H. (2016.). *Predicting Short-Term Subway Ridership and Prioritizing Its Influential Factors Using Gradient Boosting Decision Trees*. 1–16. https://doi.org/10.3390/su8111100

Do, Quang Hung & Lo, Shih-Kuei & Chen, Jeng-Fung & Le, Chi-Luan & Anh, Luong. (2020). *Forecasting Air Passenger Demand: A Comparison of LSTM and SARIMA*. Journal of Computer Science. 16. 1063-1084. 10.3844/jcssp.2020.1063.1084. https://doi.org/10.3844/jcssp.2020.1063

T. Ergen and S. S. Kozat, "*Unsupervised Anomaly Detection With LSTM Neural Networks*," in IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 8, pp. 3127-3141, Aug. 2020, doi: 10.1109/TNNLS.2019.2935975.

Jiaotong, B., & Jiaotong, B. (2013). *The application of ARIMA-RBF model in urban rail traffic volume forecast*. *Iccsee*, 1662–1665.

Kim, J., & Moon, N. (2019). *BiLSTM model based on multivariate time series data in multiple field for forecasting trading area*. Journal of Ambient Intelligence and Humanized Computing, 0123456789. https://doi.org/10.1007/s12652-019-01398-9

Koch, Patrick & Golovidov, Oleg & Gardner, Steven & Wujek, Brett & Griffin, Josh & Xu, Yan. (2018). *Autotune: A Derivative-free Optimization Framework for Hyperparameter Tuning*. https://doi.org/10.1145/3219819.3219837

Li, Hua & Zhang, Jinlei & Yang, Lixing & Qi, Jianguo & Gao, Ziyou. (2022). *Graph-GAN: A spatial-temporal neural network for short-term passenger flow prediction in urban rail transit systems.*

Li, W., Sengupta, N., Dechent, P., Howey, D., Annaswamy, A., & Uwe, D. (2021). *Online capacity estimation of lithium-ion batteries with deep long short-term memory networks*. *482*.

Liu, L., & Chen, R. C. (2017). A novel passenger flow prediction model using deep learning methods. *Transportation Research Part C: Emerging Technologies*, *84*, 74–91. https://doi.org/10.1016/j.trc.2017.08.001

Malhotra, Pankaj & Vig, Lovekesh & Shroff, Gautam & Agarwal, Puneet. (2015). *Long Short Term Memory Networks for Anomaly Detection in Time Series*.

Maurya, Sujeet & Singh, Shikha. (2020). *Time Series Analysis of the Covid-19 Datasets*. 1-6. 10.1109/INOCON50539.2020.9298390.

Nar, Melek & Arslankaya, Seher. (2022). *Passenger demand forecasting for railway systems. Open Chemistry*. 20. 105-119. 10.1515/chem-2022-0124.

Nwogu, E. C., Iwueze, I. S., Dozie, K. C. N., & Mbachu, H. I. (2019). *Choice between Mixed and Multiplicative Models in Time Series Decomposition*. *February 2022*. https://doi.org/10.5923/j.statistics.20190905.04

Pasini, K., Khouadjia, M., Ganansia, F., & Oukhellou, L. (2019). *Forecasting passenger load in a transit network using data driven models*. *World Congress on Railway Research*, 1–6. https://hal.archives-ouvertes.fr/hal-02278238

Pattana-Anake, Voravarun & John Joseph, Ferdin Joe. (2022). *Hyper Parameter Optimization of Stack LSTM Based Regression for PM 2.5* Data in Bangkok.

Pardo, R. Design, Testing, and Optimization of Trading Systems; John Wiley & Sons: New York, NY, USA, 1992; Volume 2.

Mahmoudi, Mohammad & Baroumand, Salman. (2021). *Modeling the stochastic mechanism of sensor using a hybrid method based on seasonal autoregressive integrated moving average time series and generalized estimating equations.* ISA Transactions. 10.1016/j.isatra.2021.07.013.

Wahyono, Teguh & Heryadi, Yaya & Soeparno, Haryono & Abbas, Bahtiar. (2020). *Enhanced LSTM Multivariate Time Series Forecasting for Crop Pest predictio*n. ICIC Express Letters. 14. 943-949. 10.24507/icicel.14.10.943. https://doi.org/10.24507/icicel.14.10.943

Wang, B., Wu, P., Chen, Q., & Ni, S. (2021). *Prediction and Analysis of Train Passenger Load Factor of High-Speed Railway Based on LightGBM Algorithm*. Journal of Advanced Transportation, 2021. https://doi.org/10.1155/2021/9963394

Zhao, S., & Mi, X. (2019). *A Novel Hybrid Model for Short-Term High-Speed Railway Passenger Demand Forecasting*. *IEEE Access*, *7*(1), 175681–175692. https://doi.org/10.1109/ACCESS.2019.2957612