An Approach Based on Time-Series and Neural Networks for Safety Railway Incident Prediction

Zaynabe RAGALA, Asmaa RETBI, Samir BENNANI
Rime Team-Networking, Modeling and e-Learning Team- Masi Laboratory- Engineering
3S Research Center-Mohammadia School of Engineers (EMI)
Mohammed V University in Rabat
Morocco
{zaynaberagala@research.emi.ac.ma} {retbi@emi.ac.ma} {sbennani@emi.ac.ma}

ABSTRACT: *Every day, thousands of people travel by train, not escaping the leniency of unforeseen events, the fluidity of the rail network being able to be disrupted by equipment breakdowns. Therefore, predictive maintenance is relevant and necessary to help anticipate these breakdowns and thus act against any mechanical, electrical, or technical constraints or obstacles that could disrupt or prevent the normal circulation of trains. This process is carried out using artificial intelligence approaches and various machine learning and deep learning models. This article will implement an approach that combines two essential concepts: time series and neural networks. We will start with the univariate analysis of the number of failures per day using a range of machine learning and deep learning algorithms, namely LSTM, BiLSTM, GRU, and SVR. The results show that we manage to minimize the prediction error; for example, with the GRU model, we get an RMSE of 0.487, but with increasing data, we get an RMSE of 0.463.*

Moreover, the problem encountered is the detection of peaks; the models cannot detect outliers, hence the use of the SVR model, which gives better coordination between the test data and the predicted data, with a gamma value of 0.03. Then, we tested the VAR model and the LSTM with several outputs; the latter gives satisfactory results with an accuracy rate of 92% and an RMSE of 0.006. Finally, we address the problem of classification of the nature of failures. We used several machine learning algorithms, such as SVM, KNN, Random Forest, then tested a method of "Ensemble Learning," the Vote. In the latter, we combined the three algorithms used previously, which increased the accuracy of the test to 61.73%.

Keywords: Railway, LSTM, BiLSTM, GRU, SVR, VAR, SVM, KNN, Failure Prediction, PdM

Received: 18 July 2021, Revised 14 October 2021, Accepted 19 December 2021

DOI: 10.6025/dspaial/2022/1/1/11-27

1. Introduction

Due to rapid technological advancements, predictive analytics has propelled the AI market by going beyond understanding historical data [1]. Machine learning approaches can handle multidimensional and multivariate data and extract hidden relation ships within data in complex and dynamic environments.

Predictive analytics can predict failures before they occur by detecting warning signs in real-time, such as a significant change in vibration, temperature, or machine pressure. But several obstacles have hindered its development until now. First of all, it is difficult for companies to set up a data analysis strategy [2], and there is a risk of incorrect fault diagnosis.

This paper is the continuation of an already initiated work, which aims to implement an intelligent platform dedicated to railway transport; in this paper, we will present an approach based on time series and neural networks for the prediction of faults and their impacts and their classifications—developing a deep learning approach that combines time series and neural networks, to have better accuracy, by comparing the difference in performance between several models and algorithms: the VAR, SVR, LSTM, BiLSTM, and GRU for prediction and the SVM, KNN, Random Forest, and voting for the classification of the nature of the failures.

This paper is organized into six sections; section 2 will present the context, next section 3 will overview the theoretical framework, then section 4 will present some related works, after section 5 will present the case study to apply the different techniques used, and finally a discussion of the results obtained in the last section.

2. Context

Rail systems are one of the most important means of transportation and play a crucial role in the global economy. Compared to other means, railroads offer a more comfortable experience. In addition, they are more affordable, which makes them one of the most popular means of transportation. Railroad tracks are one of the most important components of rail systems. However, the continuous impact of the repetitive passage of trains, the high speed of the railway network, the axle load, and the environmental conditions cause the deterioration of the rails and machines. The presence of even a small defect in the tracks or machines can lead to more severe defects, resulting in substantial maintenance costs and reducing the reliability and availability of the system.

The fault repair process requires an agent to travel to the fault site to diagnose the fault for repair. To do this, a history of breakdowns must be available, the aim being to carry out an analysis on several aspects: Statistical, temporal, diagnostic, and maintenance [3]. However, logistical constraints follow the maintenance teams.

This work is part of implementing a breakdown forecasting system. The operations and maintenance teams can be notified a week or even a month before the breakdown of assets. This work is part of the implementation of a breakdown forecasting system so that the operations and maintenance teams can be notified a week or even a month before the breakdown of assets, allowing them to establish precise schedules and resources thus avoiding eliminating a service train and disrupting the system.

The value of this work is to implement several models adapted to the prediction of failures, the impact of these failures on the regulation, and the classification of the nature of failures by using machine learning and deep learning approaches on datasets retrieved from the SNCF open data, to implement a predictive analysis that will be extremely useful and valuable in terms of long-term planning.

3. Theoretical Framework

In this section, we present briefly the methods and models used.

3.1. Time Series

Time series forecasting is a crucial area of machine learning, as various forecasting problems have a temporal component. A series of observations taken chronologically is known as a time series [4].

3.1.1. Principle

A time series is a set of observations measured sequentially in time. The researchers continuously make these measurements and consider a discrete set of points in time. [4]

Time series forecasting uses historical values and associated patterns in predicting future activity. This process normally includes trend analysis, cyclical fluctuation analysis, and seasonality issues. [4].

3.1.2. Univariate and Multivariate Time Series

The number of values during a particular point of time identify the variation between a univariate (UTS) and multivariate (MTS) time series [5]:

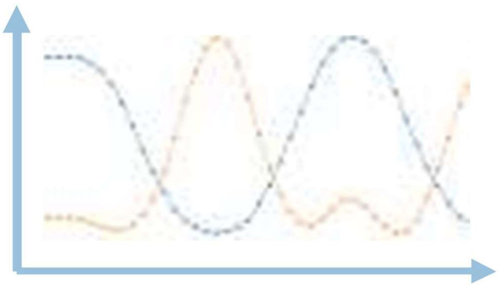

Figure 1. Multivariate analysis with 2 dimensions

Multivariate time series analysis aims to analyze several time series together. Its rationale is the potential presence of interdependencies between the different time series [5].

These interdependencies, when adequately quantified, can lead to animprovement in the reliability of forecasts [5].

3.2. The Models

3.2.1. Models used for the Prediction of Time Series

VAR :

• The vector autoregressive (VAR) model is commonly used for multivariate time series analysis [6].

• In some environments, the variables of interest are linearly related, and the VAR model is normally a good option for representing and predicting the behavior of dynamic multivariate time series.

• The VAR model is an extension of the univariate autoregressive model.

LSTM

• LSTM is the acronym for "long short-term memory networks." A variety of recurrent neural networks (RNN) have the ability for learning long-term dependencies, during the sequence prediction issues. The LSTM is capable of receiving inputs which leads to understand the process the entire data sequence, except for single data points such as images. It is a particular type of RNN, which exhibits exceptional performance in a wide variety of problems [5].

• Using a series of gates, forget gate, input gate, output gate, each with its own RNN, LSTM manages to retain, forget or ignore data points based on a probabilistic model.

• The input gate updates the cell state, and the output gate determines the value of the next hidden state. This state contains information about the previous inputs. The forget gate decides which information can be considered into account and which can be ignored.

• BI-LSTM

Bidirectional recurrent neural networks (BRNN) link two hidden layers of different directions to the same output [8]. Given the form of deep generative learning, the output layer will receive data from the two states such as past (backward), and future (forward) at the same time [5].

A bidirectional LSTM, or biLSTM, is a sequence processing system which has normally two LSTMs: the first receives the input in a forward direction and the next in a backward direction. BiLSTMs thus enhance the value of data given in the network, which ultimately improve the context open to the algorithm.

• GRU: Gated Recurrent Unit
Gated Recurrent Units (GRU) are a gating mechanism in recurrent neural networks. GRU is like a Long Short Term Memory (LSTM) with a forgetting gate but has fewer parameters than LSTM, as it lacks an exit gate [9].

The structure of the GRU allows it to adaptively capture dependencies of large data sequences without discarding information from earlier parts of the sequence. This is possible thanks to its gate units, similar to LSTMs [9].

• SVR: Support Vector Regression
The issue in the regression is to find a function that equals the mapping of an input domain into real data based on a training sample [10].

The SVR has considered as an effective tool for real-valued function estimation, even it not much known. In the supervised learning approach, SVR deploys a symmetric loss function, which equally penalizes high and low estimation errors [10].

3.2.2. Models used for Classification

SVM
The SVM, a machine learning algorithm that can be applied for classification or regression problems.

It employs a method which is named as "kernel" to transform the data. Further based on these changes, it finds an optimal boundary between the possible outputs. We can also state that it performs extremely complex data transformations and then determines how to separate the data based on the defined labels or outcomes [11]. In the SVM algorithm, it is a matter of considering each piece of data as a point in an n-dimensional space, where n is the number of features and the value of each is the value of a specific coordinate. The next step is to classify by identifying the hyperplane that differentiates the classes [11].

KNN
• K-Nearest Neighbours is a standard classification algorithm which considers particularly on the choice of the classification metric. It is "non-parametric" (only k must be fixed) and is formed mainly on training data [12]. The K-NN algorithm takes the relevance between the new case/information and the available cases and places the recent case in the most similar category to the available ones.

• The K-NN algorithm stores all available data and classifies a new item based on its similarity. This process means that when a new data item appears, it can be easily classified into a well-fitting category using the K-NN algorithm [12].

Random Forest
• The "forest" formed is a set of decision trees, most often trained with the "bagging" method. The principle of this method is that the combination of several models increases the global result. Indeed, the random forest gathers a set of decision trees and combines them to achieve a more accurate and stable prediction [13]. One of the significant advantages of random forest is that it can be used for classification and regression examples, which constitute the majority of current machine learning problems. Let us consider random forest in classification since the classification is sometimes considered the cornerstone of machine learning.

Voting
• Voting or a 'majority voting ensemble' is a 'machine learning ensemble' model that combines the predictions of several other models [14]. It is a technique that can improve model performance, ideally achieving better performance than any individual model used in the ensemble.

• A voting ensemble works by combining the predictions of several models for classification or regression. In the case of regression, we calculate the mean value of the models' predictions. For classification, the forecasts for each label are summed, and the process predicts the label with the most votes[14].

4. Related Work

This section presents the related work carried out in this perspective. Firstly, a multitude of papers that focus on the use of time series for fault prediction in different use cases and in particular in the railway domain, in [15] an approach combining three-dimensional convolutional neural networks (3D CNN), LSTM, and a fully connected neural network (FCNN) has been realized for the prediction of the delay of Chinese and Dutch trains. The results clearly show that the prediction errors of the CLF-Net are lower than those of the basic models (Markov model (MM), Artificial neural network (ANN), Support vector regression (SVR), Random forest (RF), and others) for the different railroad lines, so it showed satisfactory performance in predicting different delay times.

Secondly, authors in [16] predict train delays using a deep learning model while considering the interactions and propagation of delays between trains in a group. The contribution of this work has proved as a RNN prediction model (LSTM) where the real delay time is the dependent variable. It is more effective to assess the prediction effect of the model, two benchmark systems are chosen and compared to the LSTM model: the random forest (RF) model and the artificial neural network (ANN) model. The LSTM model has better MAE and RMSE values, and its prediction accuracy reaches 86.91% in the 30s.

However, for [17], authors opted for the ARIMA model for trend analysis and prediction processes allows for advance prediction when the train is due for maintenance, which provides for maintenance to be performed before the train experiences adverse conditions.

Next in air transport [18] and in order to predict the remaining useful life of aircraft engines, they used ARIMA to estimate the values of the predictor variables. Then, they used the result of the previous step as input to a support vector regression model (SVM).

In addition, and from the analysis of this state of the art, we notice that univariate prediction is very present in the research works, unlike those related to multivariate analysis of time series, rare are the use cases of prediction with multiple outputs in the case of multivariate time series and working with different types of data, numerical and categorical. In addition, the use of deep learning approaches is useful and gives better results in terms of accuracy but this depends on the nature and size of the input data. This does not deny the performance provided by machine learning algorithms in some cases of using time series for failure prediction.

Finally, it has been noticed that few works are working on a set of problems at the same time. Hence the particularity and the originality of this work which will consist in the use of a dataset containing categorical data, to make the univariate prediction of the number of failures per day, then the prediction of the delays of the trains which will be an embodiment of the multivariate case and finally to make the classification of the natures of the railway failures.

5. Case Study

5.1. Data Sets

The data used are those of the SNCF accessible in Open Data. They contain the origin, type of incident, date, nature of incident by region, location, and severity of all safety events involving a malfunction of the railway system, whether of internal or external origin since January 2015, supplemented by regularity data from the same source.

An RSE is a safety incident related to the actual movement that endangers or is likely to endanger the lives of people transported and in the vicinity of railway facilities, including staff, employees of service providers, and subcontractors.

5.2. Data Visualization

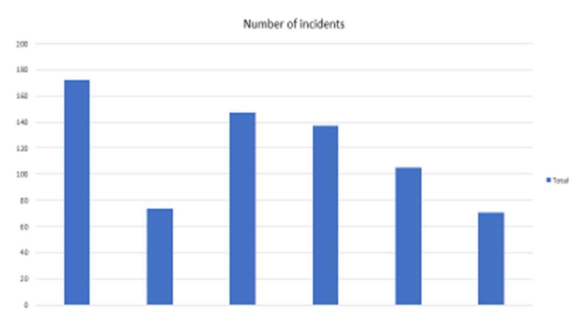

Figure 2. Cumulative incidents since 2015

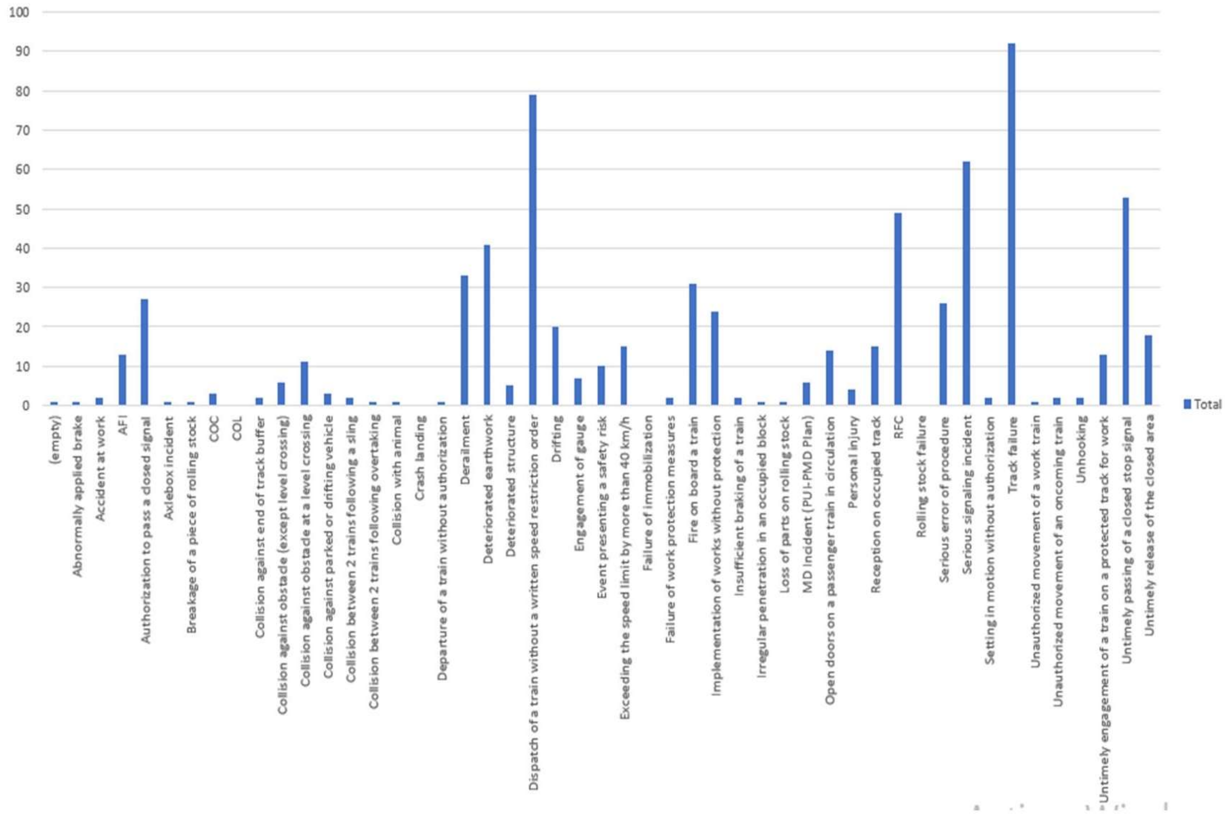

Figure 3. Distribution of incidents by nature

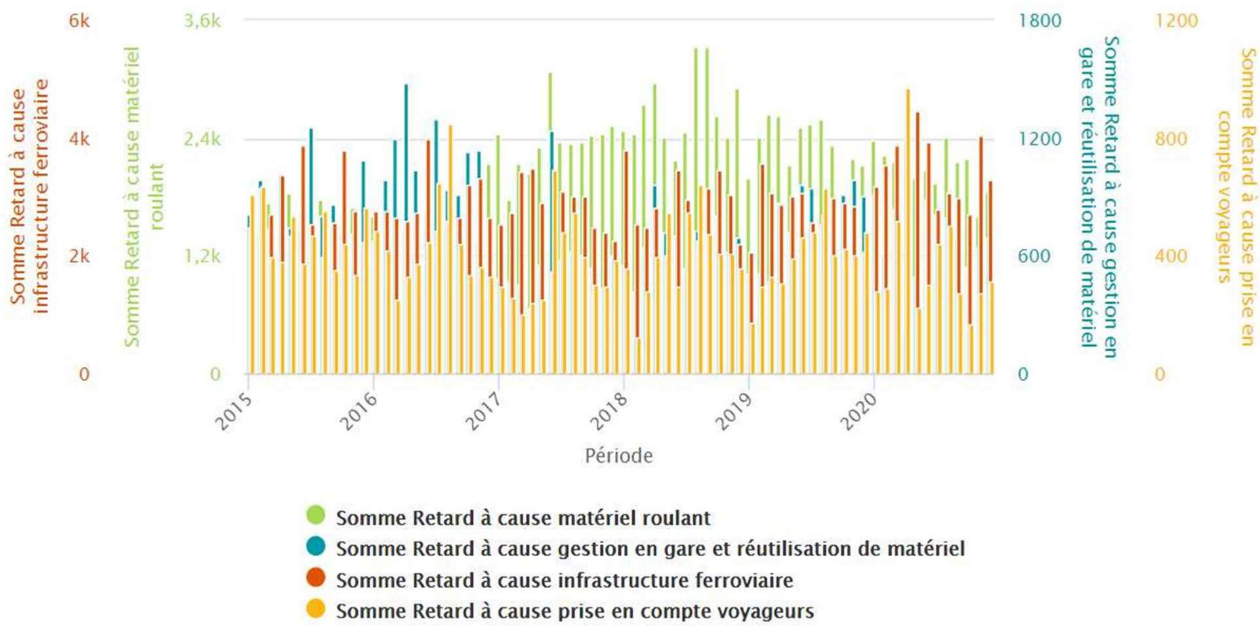

Figure 4. Cumulative delays by Nature

5.3. Approach

To make the prediction of the number of breakdowns per day, we will be led to have a distribution of time series, containing the dates of occurrence of the breakdowns and the number of breakdowns, for this step we will use different methods and algorithms used in the prediction with univariate time series, such as the model LSTM, BiLSTM, GRU and SVR that we will evaluate later.

In order to predict the delays of the incidents, which are decomposed in direct delay of the train in breakdown, indirect delay of the other trains that are stopped not to disturb the railway traffic and the total delay accumulation caused by the breakdown in minutes, we will need to take into consideration all the other factors that can impact the delay of a train, namely, the date, the nature of the breakdown in question, the series of the material that has broken down, we will resort to the techniques used for the multivariate time series, such as the VAR and LSTM.

In order to have a preliminary vision on the incidents to be faced soon, we will have to classify the nature of the breakdowns, while taking into account the delays caused by the said breakdown and the series of the train, we will have to process the categorical data, then will apply the various algorithms of automatic learning, as well as SVM, KNN, Random Forest, and other techniques of "Ensemble Learning".

5.4. Data Pre-processing

The time component has its own technical constraints: time zones, daylight saving time and different format All of these components can potentially be problematic. You have to spend the necessary time and visualize your data to make sure that these parameters are correct and constant.

A good technique to avoid headaches with time formats is to store all time components in a single numeric 'timestamp' format in the same time zone (typically UTC). This way all dates and times have a clear and unambiguous meaning.

Then, it is strongly advised to regularize your time series, i.e. to make all time intervals constant. There are several techniques, the simplest being to round the time component to the nearest hour, minute or other unit. This usually gives very good results and avoids the implementation of an interpolation algorithm.

Finally, to include the nature of the failures and the series of the train in failure, we must make a categorical data encoding and we will use the Label Encoding, this approach is very simple and consists in converting each value of a column into a number.

5.5. Results of the different models

Number of failures per Day:

We compiled the different models with the 'ADAM' optimizer, we used the Early Stopping method to avoid the Overfitting problem and finally the Jittering method of data augmentation, it is an additive noise to the data.

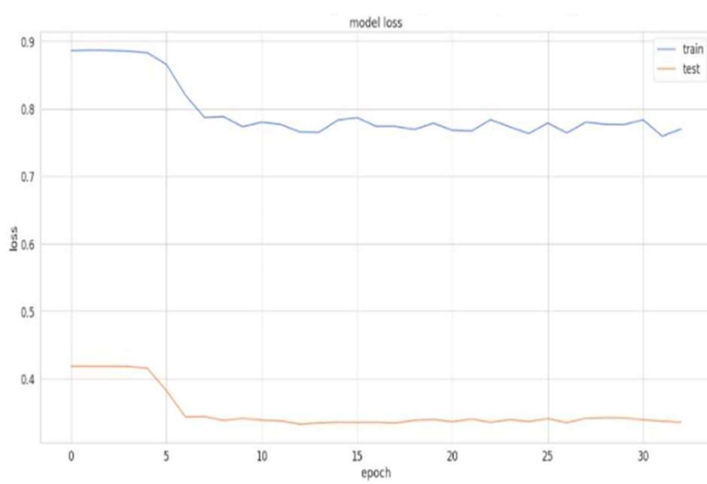

Figure 5. Loss of LSTM

LSTM:
For the configuration of the model architecture, we used an input layer of 128 units and 3 Dropout layers with a rate of 0.5 as a regularization method. Then, for the hidden layers, we used 5 LSTM blocks of 32 units and two Dropout layers with a rate of 0.7 and finally a single output layer.

Then we display the predicted and actual values in the same graph, which gives the following graph:

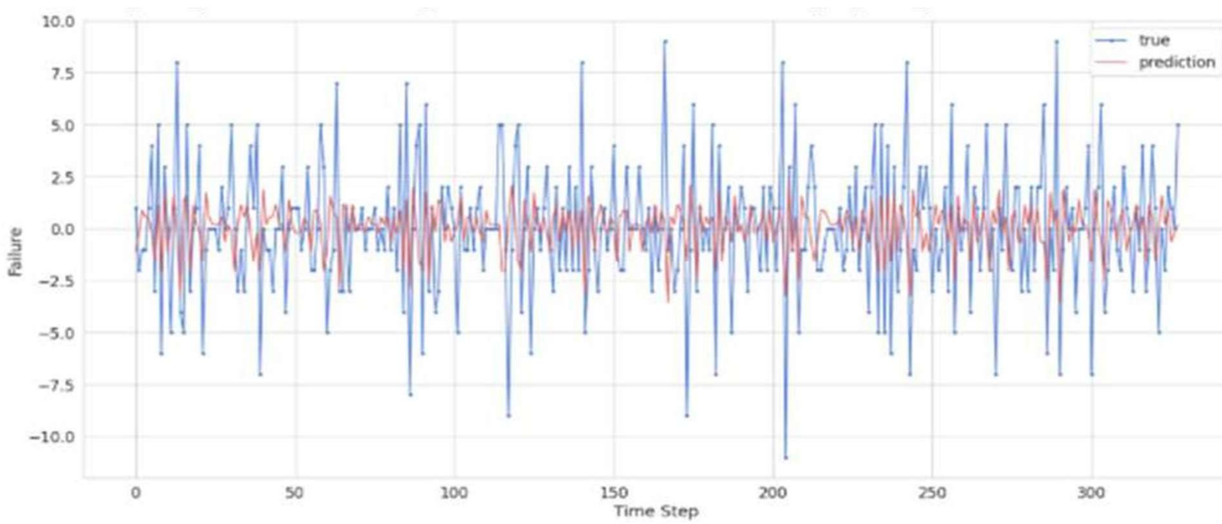

Figure 6. LSTM Predicted values Vs real values

LSTM –AUG:

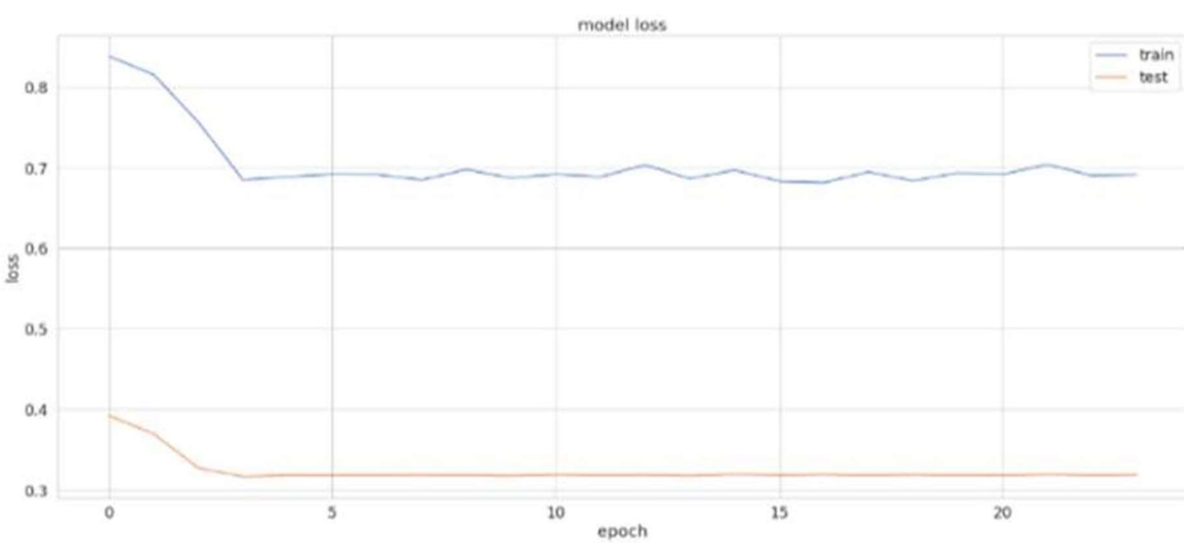

Figure 7. Loss of LSTM AUG

BILSTM:
For the model configuration, three blocks of the bidirectional LSTM of unit equal to 32, a single Dropout layer of rate 0.7 and a single output layer were used.

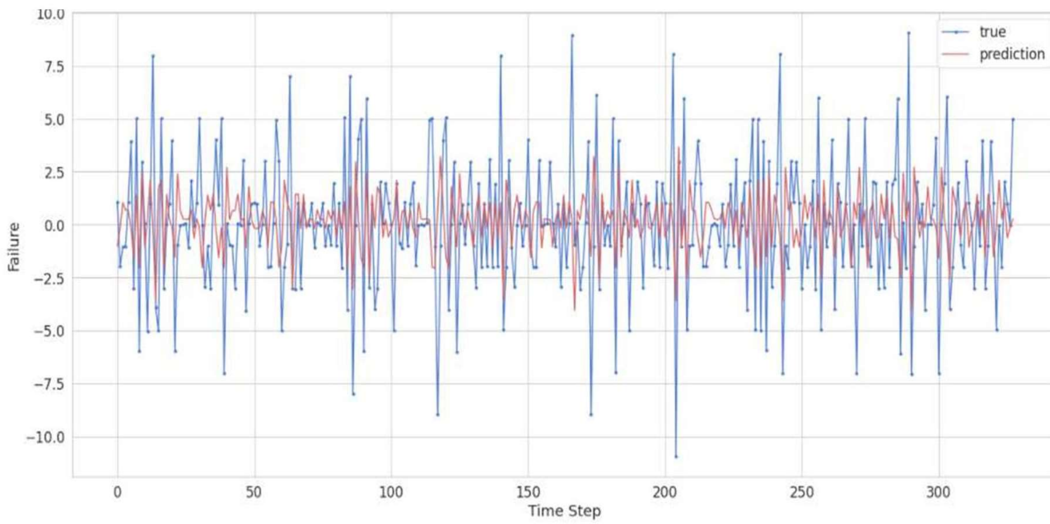

Figure 8. LSTM AUG Predicted values Vs real values

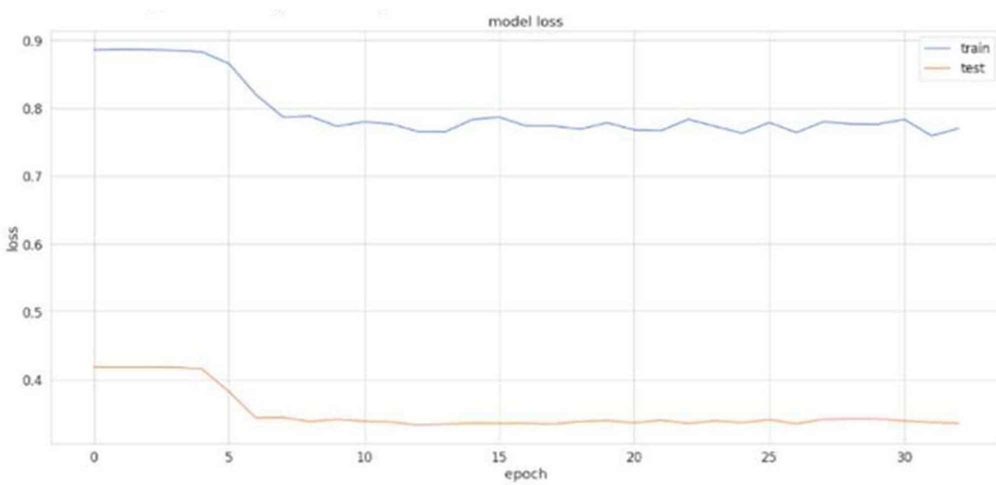

Figure 9. Loss of BILSTM

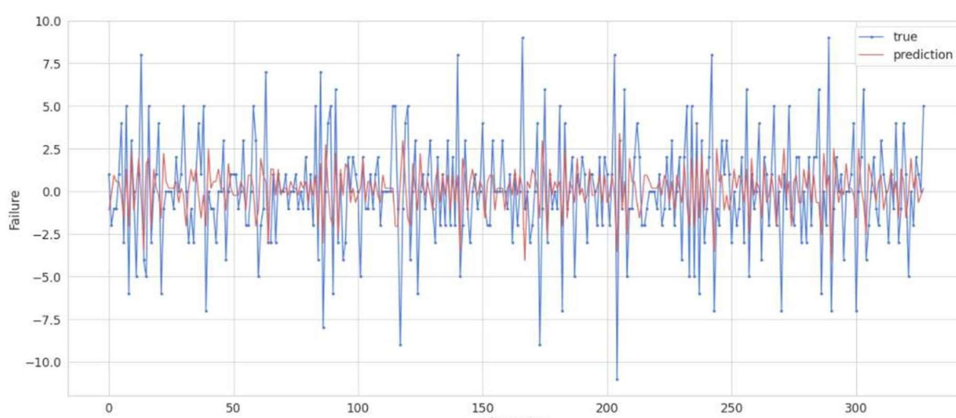

Figure 10. BILSTM Predicted values Vs real values

BiLSTM-AUG:

Using the same architecture of the Bidirectional LSTM explained before and applying the “jittering” function of data augmentation, we obtain the following graph of loss versus several epochs:

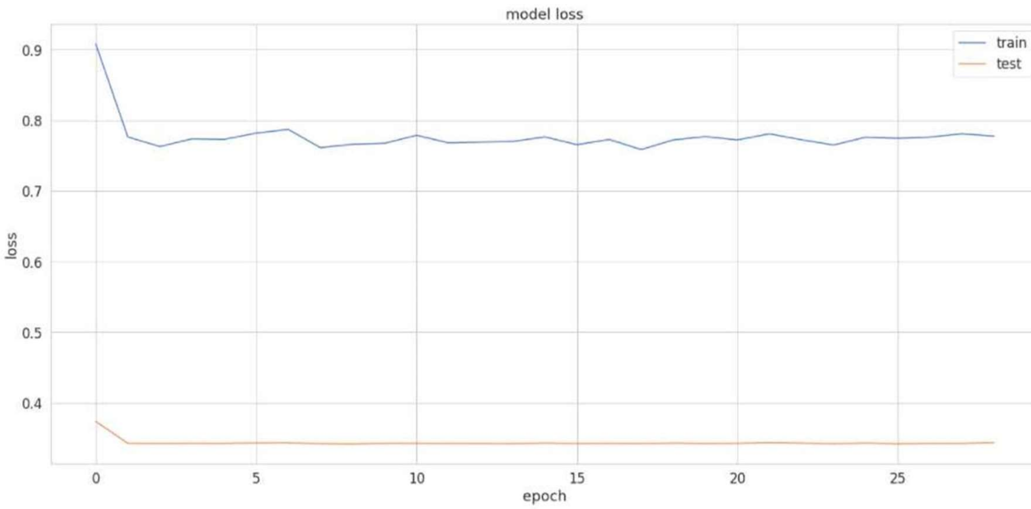

Figure 11. Loss of BILSTM AUG

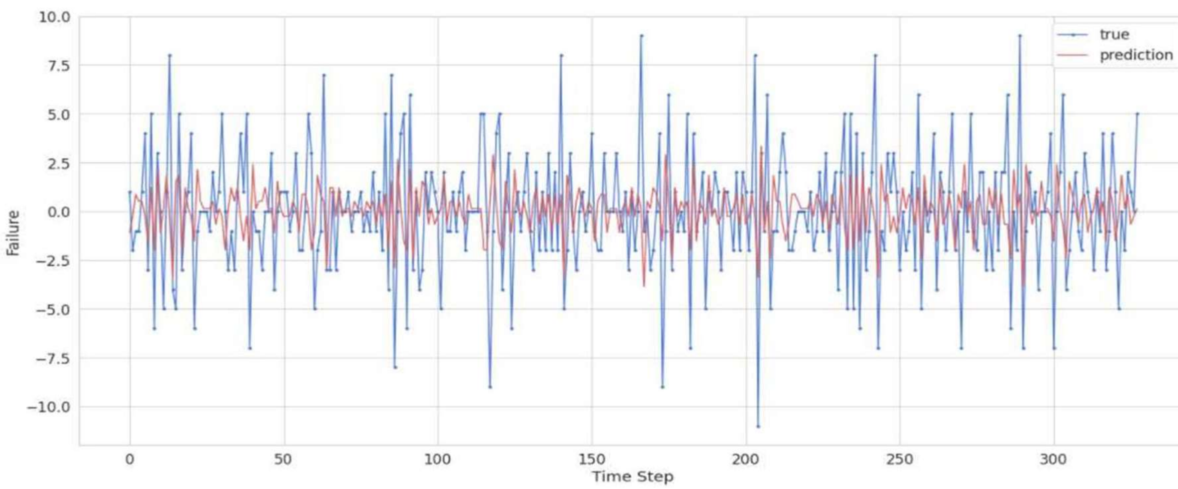

Figure 12. BILSTM AUG Predicted values Vs real values

GRU

For the configuration of the model architecture, we used an input layer of unit 32, with the L2 regularization method of a hyperparameter of 0.02 and 3 Dropout layers with a rate of 0. 2 as the regularization method. Then, for the hidden layers, we used 5 GRU blocks of unit 32 and a single Dropout layer with a rate of 0.7, and finally, a single output layer.

SVR:

After preparing the data and normalizing it with Min_Max_Scaler, and separating the data into 70% for training data and 30% for testing.

The most important SVR parameter is the type of kernel. It can be linear, polynomial or Gaussian, here we select an RBF kernel, a Gaussian type. We tried the model with two 'gamma' rbf kernel coefficients, let's see the results:

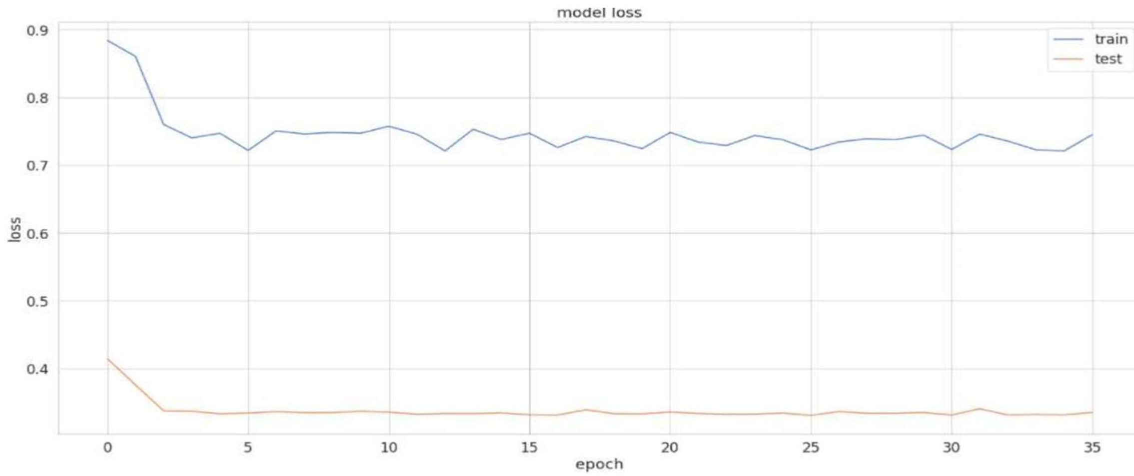

Figure 13. Loss of GRU

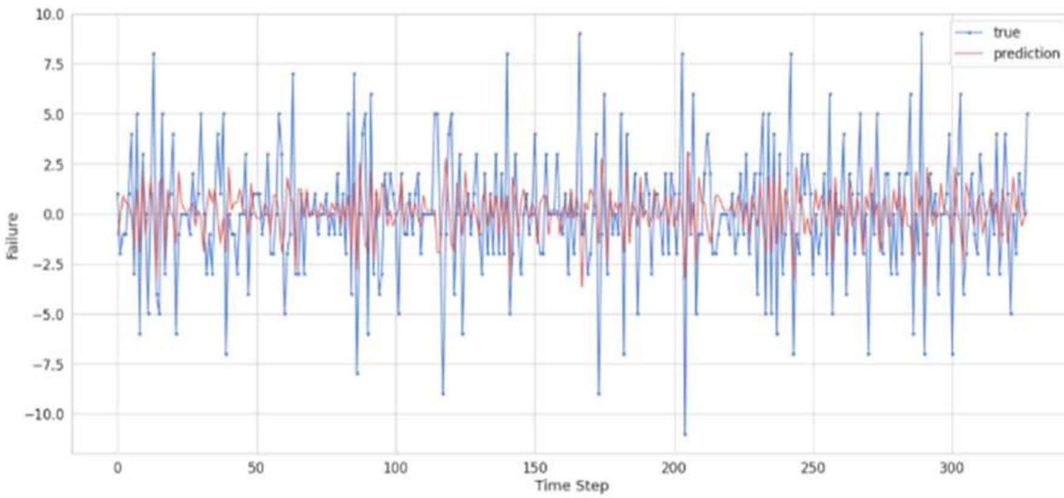

Figure 14. GRU Predicted values Vs real values

GRU-AUG

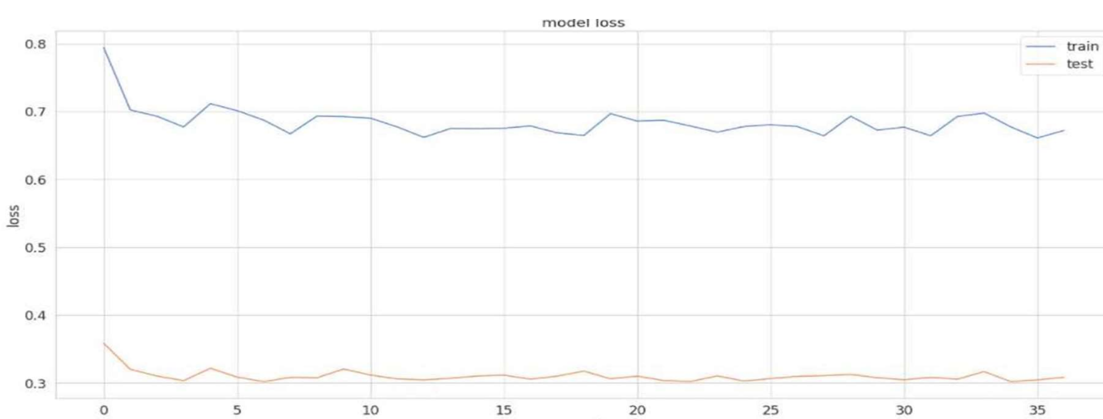

Figure 15. Loss of GRU AUG

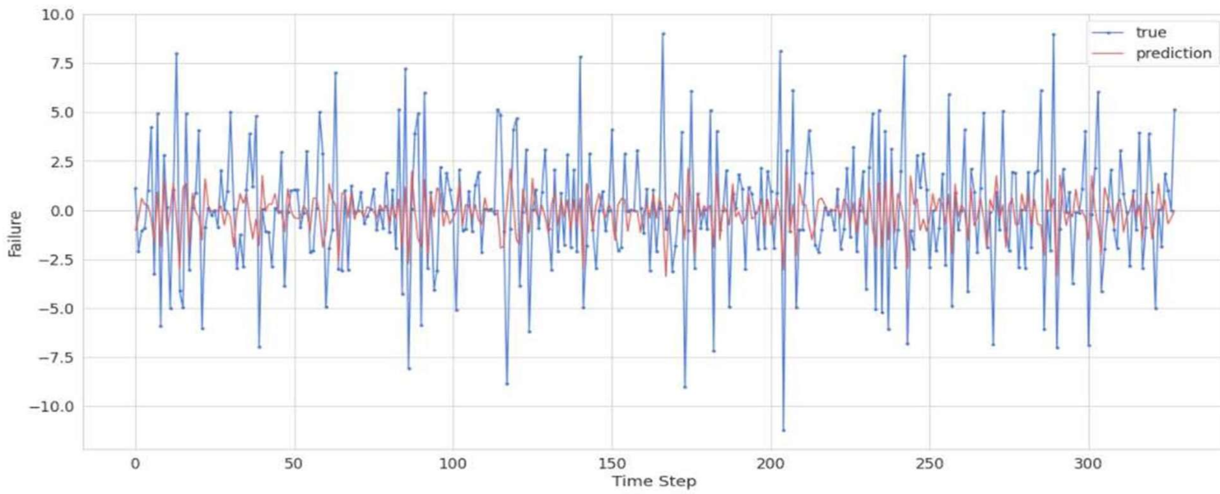

Figure 16. GRU AUG Predicted values Vs real values

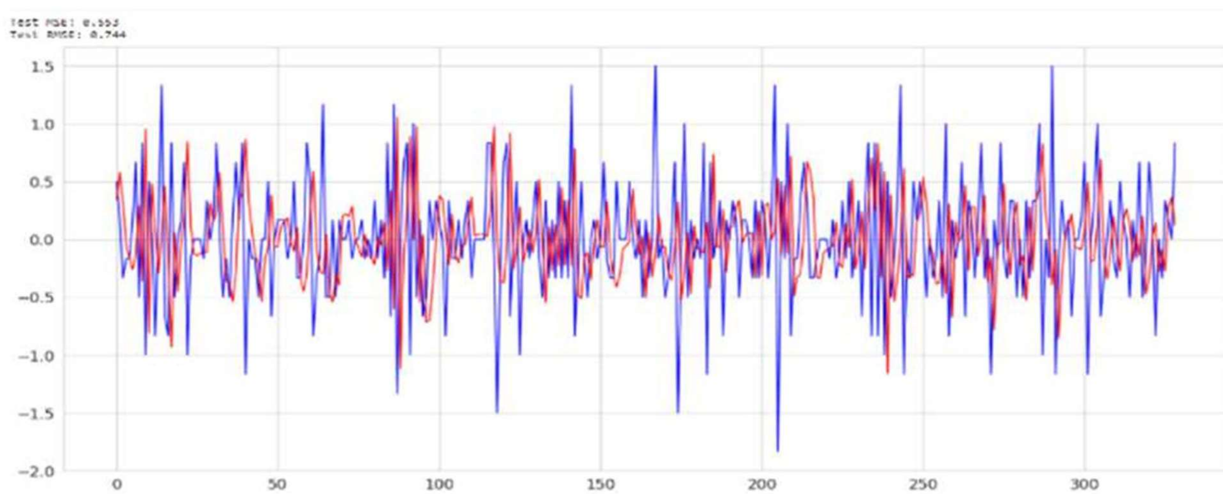

Figure 17. SVR (0.01) Predicted values Vs real values

SVR Gamma = 0.03:

Model	SVR Gamma0.03	SVRSVR Gamma0.01	LSTM	BiLSTM	GRU	LST MAUG	BiLST MAUG	GRUAUG
MSE	0.772	0.553	0.234	0.237	0.226	0.195	0.237	0.214
RMSE	0.879	0.744	0.484	0.487	0.487	0.476	0.487	0.463

Table 1. Results of the Different Models

The results of the different models are detailed bellow:

Model	VAR
RMSE	★ Direct delay : 424,54& ★ Indirect delay : 166,33& ★ Cumulative: 586,82
MAE	★ Direct delay : 411,21 ★ Indirect delay : 166,25 ★ Cumulative: 577,28

Table 2. Var Results

SVR Gamma = 0.01:

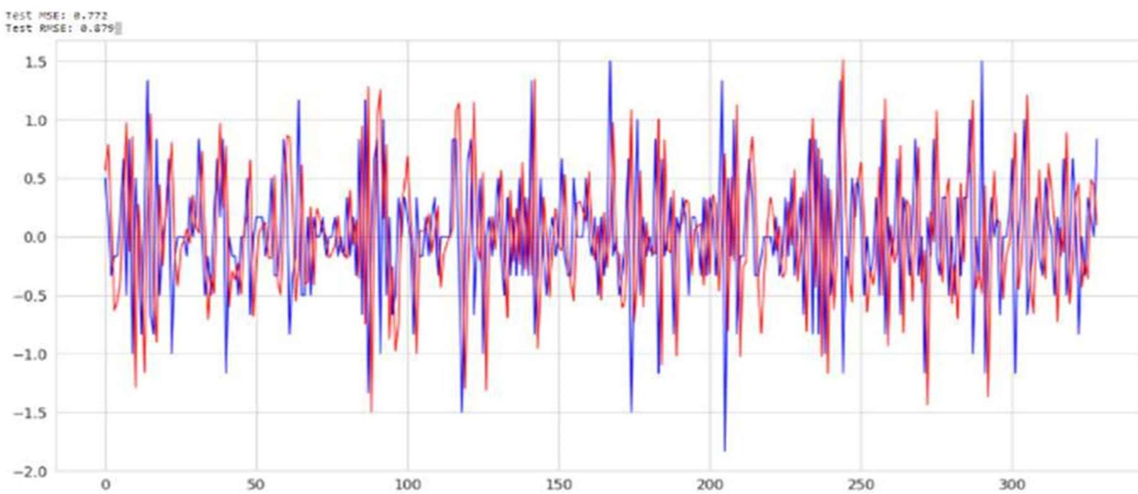

Figure 18. SVR (0.03) Predicted values Vs real values

Results of the different models

· Cumulative Delays : Direct, Indirect and Cumulative

- VAR

We proceeded to a series of tests on the dataset before applying our model:

- Causality with Granger causality
- Co-integration
- Stationarity

Next, we will calculate for each series the following evaluation metrics:

We notice that the RMSEs values for each series are very high.

Given the dissatisfaction of the results provided by the VAR model, we will try with the deep learning approach with the LSTM model.

- LSTM

First, to include the nature of the failures and the series of the train in loss, we must make a categorical data encoding. We will use Label Encoding; this approach is straightforward and consists of converting each column's value into a column number.

Next, we have to resort to the normalization of numerical data to alleviate the concern of our data's different orders of magnitude. After that, we divide our data into a training set with 70% of the total dataset and a test set with 30%.

The activation function we used is the one used by the default model. The model is trained 500 times with a batch size equivalent to 32 and compiled with the "ADAM" optimizer.

Finally, to evaluate our model, we have to calculate the following metrics, and we notice that the RMSE is very low compared to the results obtained before, with a value of 0.006.

Model	**LSTM**
RMSE	0.006
MAE	0.003

Table 3. LSTM Results

· **Classification of the nature of failures :**

In order to determine the nature of the failure that will occur, we will proceed the classification of failures by nature.

- KNN

In order to choose the best value of k, we compare the accuracy of the model for each value of k, to choose the hyperparameter that gives the maximum accuracy. In this case the best value of k is 18 with an accuracy of 0.604.

After having made the predictions with the KNN model of parameter 18, we recalculate the accuracy to obtain a value of 61.3%.

- SVM

Now we move on to applying the SVM model, which we will test with different types of kernels to choose the one that gives the best results.

After training the SVM models with different types of kernels, we move on to the evaluation; we notice that the RBF kernel gives the best results with an accuracy score of 57.84% and an F1 score equal to 47.25%:

Kernel	**Accuracy**	**F1**
Polynomial	42.73%	25.58%
RBF	56.84%	46.25%
Sigmoid	36.41 %	29.23 %

Table 4. SVM Results

- Random Forest
The performance results of the Random Forest model give as accuracy score 62.2%.

- Voting
A Voting classifier is a machine learning model that trains on a set of many models and predicts an output class based on the highest probability of the class chosen as output. To apply this method to our database, we use the three classifiers seen before: k-Nearest Neighbors, Random Forest and SVM. The results of the Voting model, which includes three classifiers, give an *'accuracy'* score for the test data of 61.73%, and for the training data of 64.66%.

Model	**KNN**	**RF**	**SVM**	**Voting Classifier**
Accuracy	61,3%	62,2%	57.84%	64.66%

Table 5. Classification of failures by Nature

The table below presents the different accuracy values obtained in the problem of classification of the nature of failures, using other methods and several algorithms:

6. Discussion of the Results

In this section we will discuss the results obtained on the simulation section presented before.

So after the realization of the models, we obtained the first result in favor of the SVR model in the prediction of the number of breakdowns per day, LSTM for the prediction of delays, and Voting in the classification of the nature of railway breakdowns.

- Number of breakdowns per day
For a distribution that varies between 0 and 32 breakdowns per day, we notice that the error between the actual values and the one predicted in most of the models does not even reach 1, i.e., The model is wrong in its predictions with less than one breakdown.

In terms of conclusion, the data augmentation method minimizes the error with a very minimal value compared to raw data.

For a distribution that varies between 0 and 32 breakdowns per day, we notice that the error between the fundamental values and the one predicted in most of the models does not even reach 1, i.e., The model is wrong in its predictions with less than one breakdown.

In terms of conclusion, the data augmentation method minimizes the error with a very minimal value compared to raw data.

Thus, for the SVR model, we notice that it gives better coordination between the test data and the predicted data, especially with a gamma value equal to 0.03

- Direct, indirect and cumulative lag
We notice that the values obtained by the LSTM model are acceptable and even promising compared to those obtained with the VAR model, and it reaches an accuracy of 92%.

- Classification of the nature of failures
The results show that the Voting Classifier gives the highest value of test accuracy with 61.73%; we can explain by combining the three models for a learning set.

7. Conclusion

Unlike simpler classiûcation and regression problems, time series problems add the complexity of order or time dependence between observations. The contribution of our work lies in exploiting the techniques used in the cited research and trying other new ones to compare the different results as well as working with time series containing other categorical characteristics and doing the same with multivariate time series, whose output will be multivariate and not only the input as in the research work. Moreover, the originality of the work is presented through the combination of different approaches in the same work, namely the prediction of univariate and multivariate time series and classification.

IoT-based predictive maintenance helps monitor, maintain and optimize assets for better availability, utilization, and performance. It also provides better visibility into assets via real-time monitoring, predicting machine failures, and identifying parts that need to be replaced. For future work, we will integrate connected devices, sensors, and big data technologies to improve its maintenance operations.

For future work, we will adopt the integration of connected devices, sensors and big data technologies to improve its maintenance operations. IoT-based predictive maintenance helps to monitor, maintain and optimize assets for better availability, utilization and performance. It also provides better visibility into assets via real-time monitoring, which can predict machine failures and identify parts that need to be replaced.

References

[1] Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., Francisco, RdP., Basto, J.P. & Alcalá, S.G.S. (2019) A systematic literature review of machine learning methods applied to predictive maintenance. *Computers and Industrial Engineering*, 137, 106024 [DOI: 10.1016/j.cie.2019.106024].

[2] Kaffash, S., Nguyen, A.T. & Zhu, J. (2021) Big data algorithms and applications in intelligent transportation system: A review and bibliometric analysis. *International Journal of Production Economics*, 231, 107868 [DOI: 10.1016/j.ijpe.2020.107868].

[3] Botte, M. & D'Acierno, L. (2018) Dispatching and Rescheduling Tasks and Their Interactions with Travel Demand and the Energy Domain: *Models and Algorithms. Urban Rail Transit*, 4, 163–197 [DOI: 10.1007/s40864-018-0090-8].

[4] Brownlee, *J. Deep Learning for Time Series Forecasting*, p. 574.

[5] Plaud, A., Nguifo, E. & Charreyron, *J. Classification des séries temporelles multivariées par l'usage de Mgrams*, p. 12.

[6] Bashir, F. & Wei, H. (2018) Handling missing data in multivariate time series using a vector autoregressive model-imputation (VAR-IM) algorithm. Neurocomputing, 276, 23–30 [DOI: 10.1016/j.neucom.2017.03.097].

[7] M.S. says (2021). LSTM | Introduction to LSTM | Long Short Term Memor », Analytics Vidhya, mars 16. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/ (consulté le juin 29, 2021).

[8] Sun, Q., Jankovic, M.V., Bally, L. & Mougiakakou, S.G. (2018) Predicting blood glucose with an LSTM and bi-LSTM based deep neural network. In: 14th Symposium on Neural Networks and Applications (NEUREL), Belgrade, November 2018, pp. 1–5 [DOI: 10.1109/NEUREL.2018.8586990].

[9] Shen, G., Tan, Q., Zhang, H., Zeng, P. & Xu, J. (2018) Deep learning with gated recurrent unit networks for financial sequence predictions. Procedia Computer Science, 131, 895–903 [DOI: 10.1016/j.procs.2018.04.298].

[10] Barbour, W., Martinez Mori, J.C., Kuppa, S. & Work, D.B. (2018) Prediction of arrival times of freight traffic on US railroads using support vector regression, [août 2018]. *Transportation Research Part C*, 93, 211–227 [DOI: 10.1016/j.trc.2018.05.019].

[11] Hua, G., Zhu, L., Wu, J., Shen, C., Zhou, L. & Lin, Q. (2020) Blockchain-based federated learning for intelligent control in heavy haul railway. IEEE Access, 8, 176830–176839 [DOI: 10.1109/ACCESS.2020.3021253].

[12] Wu, Y., Ianakiev, K. & Govindaraju, V. (2002). *Improved k-nearest neighbor classiÿcation. Pattern Recognition*, 8.

[13] Jiang, C., Huang, P., Lessan, J., Fu, L. & Wen, C. (2019) Forecasting primary delay recovery of high-speed railway using multiple linear regression, supporting vector machine, artificial neural network, and random forest regression. Canadian Journal of Civil Engineering, 46, 353–363 [DOI: 10.1139/cjce-2017-0642].

[14] Brownlee, J. (2020). How to develop voting ensembles with python. Machine Learning Mastery. https://machinelearningmastery.com/voting-ensembles-with-python/ (consulté le juill. 01, 2021).

[15] Huang, P., Wen, C., Fu, L., Peng, Q. & Tang, Y. (2020) A deep learning approach for multi-attribute data: A study of train delay prediction in railway systems. *Information Sciences*, 516, 234–253 [DOI: 10.1016/j.ins.2019.12.053].

[16] Wen et al. “2020-A Predictive Model of Train Delays on a Railway li.pdf.

[17] F. Francis et M. Mohan. ARIMA Model Based Real Time Trend Analysis for Predictive Maintenance, in (2019) 3rd Inter national conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, juin 2019, pp. 735–739 [DOI: 10.1109/ICECA.2019.8822191].

[18] Ordóñez, C., Sánchez Lasheras, F., Roca-Pardiñas, J. & Juez, F.JdC. (2019) A hybrid Arima–SVM model for the study of the remaining useful life of aircraft engines. *Journal of Computational and Applied Mathematics*, 346, 184–191 [DOI: 10.1016/j.cam.2018.07.008].