Overstock prediction using machine learning in retail industry

K. Bernard Agbemadon
FEMTO-ST Institute, CNRS
Université de Franche-Comté (UFC)
Belfort, France
kodjo.agbemadon@univ-fcomte.fr

Raphaël Couturier
FEMTO-ST Institute, CNRS
Université de Franche-Comté (UFC)
Belfort, France
raphael.couturier@univ-fcomte.fr

David Laiymani
FEMTO-ST Institute, CNRS
Université de Franche-Comté (UFC)
Belfort, France
david.laiymani@univ-fcomte.fr

***Abstract*—Success in supply-chain relies, in large part, on good stock management. It is quite simple to guess that there will be an increase in demand for a type of product, or rather reluctance over a period of time, but it becomes complicated to know in advance the exact or optimal number of products to order to avoid stock-outs and at the same time overstocking.**

This article shows how transactional data can be used with Machine Learning to forecast demand in the retail industry. To train the machine learning models, a sample of 5,115,472 records of receipt data was obtained from the French branch of one of the largest Belgian supermarket chains's data warehouse. The results revealed that the machine learning models manage to learn the seasonality effects and allow to make better predictions.

***Index Terms*—supply chain, overstock prediction, retail industry, machine learning, deep learning**

I. Introduction

Forecasting demand is one of the biggest challenges in the retail industry, which essentially answers the question: what is the probability distribution, in the future, of the demand for a product or a product family, over a given time period? Among its numerous advantages, a predictive forecast is a crucial enabler for lower costs thanks to better planned inventory and fewer write-off items, as well as for a better customer experience by reducing out-of-stock situations [9]. Manufacturers, distributors, retailers and other businesses are constantly looking for more accurate predictions to reduce uncertainty in decision-making. Accurate demand forecasting [1], especially in retail, leads to well-informed purchasing and better inventory management, scheduling, capacity management and assortment planning decisions. A common definition of demand forecasting refers to the practice of estimating future customer demand over a predetermined period of time using historical data and other information. [9].

When companies use loyalty cards, thousands of attribute values are stored for each buyer. Those data include useful knowledge which is often buried in the large array of raw data. It should be noticed that these datasets mainly contain structured data that can be requested through SQL and semi-structured data such as Excel, JSON and CSV files. This study uses a private dataset from Colruyt France (the French branch of Colruyt which is one of the largest Belgian supermarket chain' data warehouse), a retail company with 90 supermarkets (700 to 1200 m²), mainly located in the Franche-Comté region, France. This dataset represents only in-store purchases, i.e. sales receipts. The dataset does not directly concern information related to the loyalty card and it focuses on two major product families, the dairy family which includes (milk, cheese, yogurt, ice cream, ...) and the fish family which includes products derived from fish.

The most often used methods to forecast demand try to identify seasonality and trend in time series of sales data and, in this way, try to create correlations between the variable of interest and other independent variables. However, these methods struggle to perform effectively when the dependent variable also depends on external variables. The techniques presented in this document manage to deal with these situations by using state-of-the-art machine learning techniques since it is now, widely accepted that Machine Learning techniques manages to perfectly extract hidden characteristics from raw data. Our approach, here, is to train machine learning and deep learning models on the dataset presented above, with a limited number of samples. We show that for the best scenario, the Autoformer model [30] gives the best results with a Mean Squared Error (MSE) equal to 0.43.

The remainder of the paper is structured as follows. Section II presents the context, regarding the definition of overstock and the different levels of difficulty that can be encountered. Section III explains the methodology, from data acquisition to evaluation of models. Section IV details the different models used and section V presents the obtained results and compares the different techniques used to tackle the problem. Finally, Section VI summarizes the conclusions drawn from the paper.

II. Context

A. The overstocking problem

Success in supply-chain relies, for a large part, on good stock management. It is quite simple to guess that there will be an increase in demand for a type of product, or rather

reluctance over a period of time, but it becomes complicated to know in advance the exact or optimal number of products to order to avoid stock-outs and at the same time overstocking. Overstock in retail industry, usually means having too much stock in a store that has not sold. One consequence of overstocking in some supermarkets is that the latter is obliged to make a special promotion called 'anti-waste' on products to make them more attractive. In anti-waste promotions, the company agrees to sell excess products at a loss just before their expiration date instead of throwing them away. This is often the case with short-life products. Even an anti-waste promotion cannot prevent some products from ending up in the trash because customers were unable to purchase all of the discounted products before their expiration date. Anti-waste is very different from traditional promotion which is more about attracting new customers.

B. Related works

During the past few years, there have been a lot of research in the field of demand forecasting. This section will provide an overview of the literature on time series forecasting and the application on product sales [13], [26]. Traditional times series models, Machine learning models and Deep learning models are among the relevant works. Predicting time series allows researchers to understand the changes in systems without having to design the exact parameters that influence them. After decades of study, time series models have made great progress and have been used for various projects in many application areas. The earliest data analysis methods can be traced back to 1970. Following Markov process [14], we can cite ARIMA [10], an auto-regressive model for recursively sequential forecasting. However, an autoregressive process is not sufficient to deal with non-linear and non-stationary sequences. Already in 1996, Ansuj et al. [5] used the AutoRegressive Integrated Moving Averages (ARIMA) model with interventions and the Artificial Neural Network (ANN) model to analyze sales data covering a 10-year period (1979 to 1989). Compared to the ARIMA model, the ANN model's predictions were more accurate. In 2001, Alon et al. [3] conducted a comparative research between conventional techniques and ARIMA models with ANN models on US aggregate retail sales data. Based on the empirical results, they were able to deduce that for different forecast periods and different forecast horizons, ANN performed best. In 2017, Aras et al. [6] provide a good overview of the literature and a comparative study on retail sales forecasting of "an international furniture company, which has operated in Turkey's retail sector for many years" between methods with different approaches like ARIMA and ARFIMA models, ETS (Error, Trend, Seasonal), Artificial Neural Networks (ANN) and Adaptive Network-based Fuzzy Inference System (ANFIS). According to them, it is almost impossible to know in advance which forecasting model will perform best for a given data set. No single model is best for all situations and circumstances. With the rise of deep neural networks over the past few years, recurrent neural networks (RNN) have been designed to better handle of sequential data. To address the gradient vanishing or exploration problem, RNNs such as LSTM and GRU [18] use a gated structure to control the flow of information. DeepAR [22] incorporates binomial likelihood into a sequential architecture for probabilistic prediction. Attention-based RNN [24] uses temporal attention to capture long-range dependencies. However, the recurrent model is not parallelizable and cannot to handle long dependencies. The recurrent model, on the other hand, is not parallelizable and cannot handle extended dependencies. Another useful family in sequential tasks is the temporal convolutional network [23]. However, because the kernel's receptive field is limited, the extracted features are always local, making long-term dependencies difficult to grasp.

Transformers-based architectures have shown their effectiveness in natural language processing NLP and computer vision tasks [15], [21], [28]. They are now also applied in time series forecasting [29] and also show their efficiency. For example in [25], an encoder-decoder architecture is used for sequence-to-sequence time series forecasting tasks. In transformers, the core layers are the self-attention and cross-attention mechanisms.

However, to the best of our knowledge, no existing work provides a clear answer as to the effectiveness of ML models for the overstock prediction problem.

III. Data Acquisition

This research includes Colruyt sales data from the years 2017 to 2022. Table I presents the structure of the dataset in its raw state.

TABLE I: A sample of the input dataset

product_ID	date	quantity	unit_price
229490008301	2015-02-02	738	1.2363
229490008301	2015-02-04	366	1.0368
229490008301	2015-02-05	521	1.1131

The product ID includes information about the category it belongs to. Therefore, the products were grouped into categories and two main categories were experimented in this study. These two main product families were the ones that usually generate the most anti-waste promotions. Namely the dairy family which includes (milk, cheese, yogurt, ice cream, ...) and the fish family which includes products derived from fish. Table II gives an overview of this dataset.

TABLE II: An overview of the dataset

Labels	Values
Number of sales	5,115,472
Number of articles families	2
Number of articles	150
Time period	2017 - 2022
Granularity	days

IV. Models

In the following, we present some machine learning approaches that have been proven to work for time series forecasting in general and that we used in our experimentations.

A. Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm [12]. Gradient Boosting represents a set of Machine Learning algorithms that can be used for predictive classification or regression modeling problems and that are built on decision tree models. XGBoost has become the gold standard method and often the key component of winning solutions for classification and regression problems in Machine Learning competitions. In this paper we tested the XGBoost Regressor API of the open-source Gradient Boosting implementation.

B. LSTM

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) that can learn order dependencies in sequence prediction problems [18]. In traditional Neural Networks, each input and output is independent, but in cases of predicting the next word in a sentence, the previous words are needed and therefore the previous words must be remembered. Similarly for time series, past values can influence the current value. This inspired the application of RNN for time series and the development of RNN-based architectures for time prediction. While introduced in the late 90's, LSTM models have become in last decade a viable and powerful forecasting technique for time-series. An LSTM rectifies a huge issue that recurrent neural networks suffer from: short term memory i.e. the inability to learn dependencies from long sequences. Using a series of 'gates' [16] each with its own RNN, a LSTM manages to keep, forget or ignore data points based on a probabilistic model.

C. Attention family

Transformer, Informer and Autoformer models are all fundamentally based on the mechanism of Attention. Bahdanau et al. [7] proposed the attention mechanism to address the bottleneck problem that arises when using a fixed-length encoding vector, as the decoder would have limited access to the information provided by the input. This is thought to be especially problematic for long and/or complex sequences, whose dimensionality would be forced to be the same as for shorter or simpler sequences. Among these three architectures, the Transformer was the first to be developed [28] and has shown its effectiveness in many areas (natural language processing, computer vision...) In the following, the specificity of the Autoformer model regarding to the rest of Attention family will be explained.

D. Autoformer

Autoformer is a decomposition architecture that embed the series decomposition block as an inner operator, which can progressively aggregate the long-term trend part from intermediate prediction [30]. Besides, there is an Auto-Correlation [30] mechanism to conduct dependencies discovery and information aggregation at the series level, which contrasts clearly from the previous models of the attention family. See figure 1 for an Autoformer illustration.

E. Hyper-parameters

For the XGBoost, we have used the Scikit-learn [20] library and its hyper-parameters tuning function which considers the cross validation, to automate the search for the best configuration. We provided this function with the splitting of the time series which is common to all the tested models. The search ranges were $[0.005, 0.05]$ for the learning rate, $[5, 30]$ for the maximum depth, $[50, 1000]$ for the number of estimators. The configurations then converged to a learning rate of 0.03, a maximum depth of 100.

The LSTM model as well as the other attention-based models took advantage of an automatic adjustment of the learning rate and an early stopping system. The best of LSTM models, contained 8 layers of LSTM with 16 hidden states and a dropout equal to 0.25. The search ranges were $[1, 10]$ for the number of layers, $[4, 32]$ for the hidden state dimension, $[0.2, 0.5]$ for the dropout. The learning rate started at 0.1 and could be adapted during training, it could go down to 10^{-10}.

The attention-based models, because of their similarity to each other, were able to benefit from the same search ranges overall. Some of them still gave better results on some ranges than on others. The search ranges were $[4, 35]$ for the number of heads, $[2, 5]$ for the number of encoding and decoding layers, $[10, 25]$ for the moving average, $[1, 3]$ for the attention factor. The activation functions Rectified Linear Unit (ReLU) [2] and Gaussian error linear units (GELU) [17] have been tested. The models gave better results with the GELU activation function.

V. Results and discussion

A. Cross Validation & Forward chaining

Cross-validation [8] and Forward-chaining [4] are used to evaluate the models in this paper. Cross-validation [8] is frequently used in the evaluation of regression and classification models. Applying it to the time-series or other naturally ordered data adds some complexity because of the chronology of events. To accurately simulate the real world forecasting environment, in which we stand in the present and forecast the future, the forecaster must withhold all data about events occurring chronologically after the events used to fit the model. Rather than using k-fold cross-validation, we use hold-out cross-validation for time series data, in which a subset of the data (split temporally) is reserved for validating model performance. As shown in Figure 4, the test set data follows the training set chronologically. Similarly, the validation set follows the training subset chronologically.

Recall that the dataset used to evaluate the performance of the tested forecasting models contains daily purchases of 150 products divided into 2 main categories.

B. Discussion

Root-mean-square deviation (RMSE) and Mean absolute error (MAE) are used as measurement tools in this paper to measure the reliability of the various prediction models [11]. In the remainder, category A represents the Fish category, while category B represents the Dairy category.

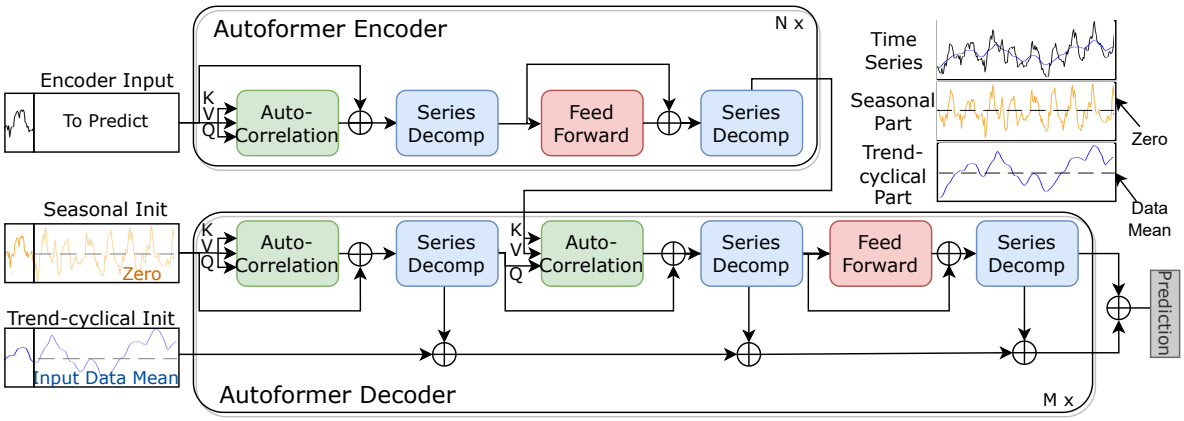
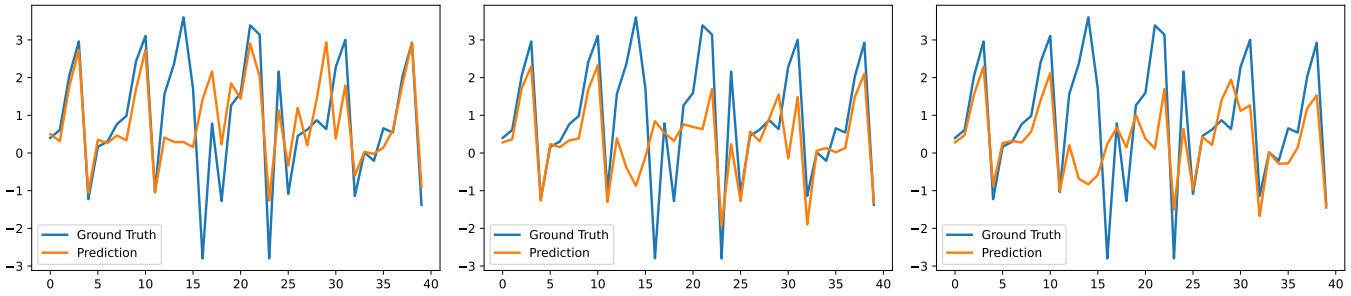


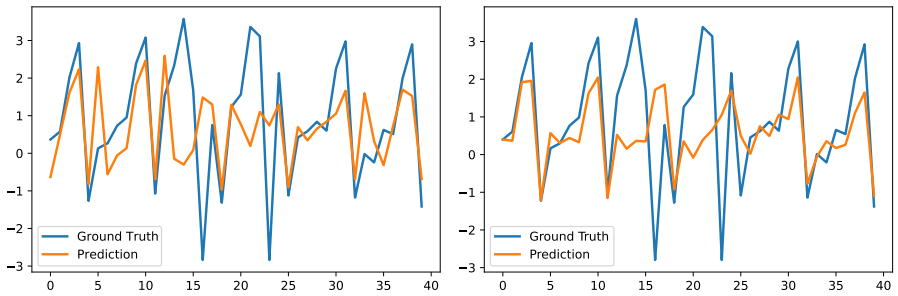
Fig. 1: Autoformer architecture. The encoder eliminates the long-term trend-cyclical part by series decomposition blocks (blue blocks) and focuses on seasonal patterns modeling. The decoder accumulates the trend part extracted from hidden variables progressively. The past seasonal information from encoder is utilized by the encoder-decoder Auto-Correlation (center green block in decoder).

TABLE III: Results on two datasets with predicted length as 1, 7, 30, 60. (Highest values in bold)

Models		Autoformer		Informer		Transformer		XGboost		LSTM	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Fish	1	**0.43**	**0.36**	0.50	0.42	0.54	0.43	0.67	0.47	0.59	0.45
	7	**0.52**	0.40	0.53	0.39	**0.52**	**0.39**	0.60	0.45	1.08	0.77
	30	**0.53**	0.44	0.69	0.48	**0.53**	**0.38**	0.71	0.50	1.67	0.99
	60	**0.55**	0.45	0.83	0.58	0.56	**0.40**	0.70	0.50	1.79	1.03
Dairy Lait	1	**0.43**	0.37	0.44	0.39	0.47	**0.36**	0.68	0.46	0.54	0.44
	7	**0.48**	**0.37**	0.49	0.38	**0.48**	0.38	0.60	0.46	0.98	0.73
	30	0.52	0.45	0.64	0.46	**0.48**	**0.36**	0.72	0.51	1.52	0.94
	60	**0.51**	0.43	0.78	0.56	0.53	**0.39**	0.75	0.53	1.64	0.98

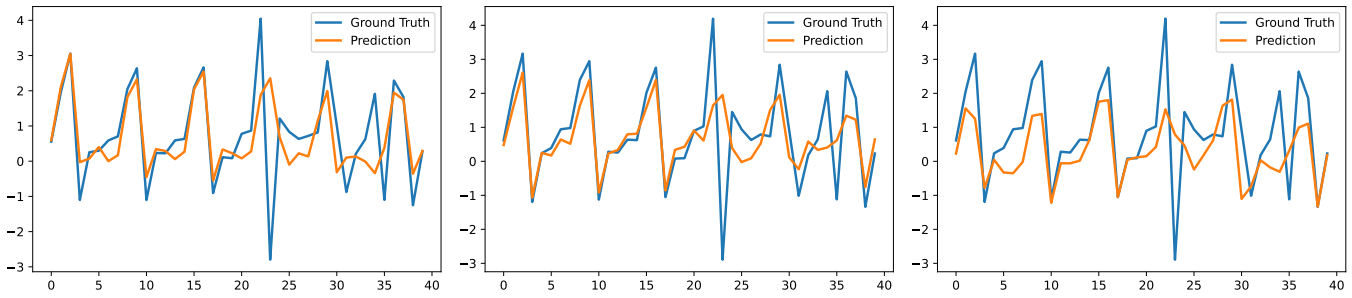


(a) Autoformer (b) Informer (c) Transformer

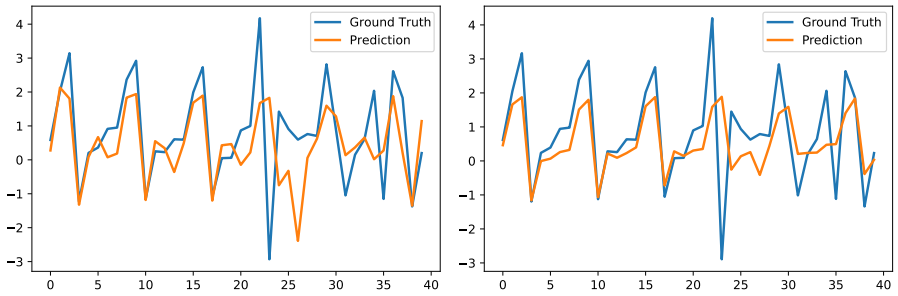


(d) XGboost (e) LSTM

Fig. 2: Product category A dataset



(a) Autoformer (b) Informer (c) Transformer



(d) XGboost (e) LSTM

Fig. 3: Product category B dataset

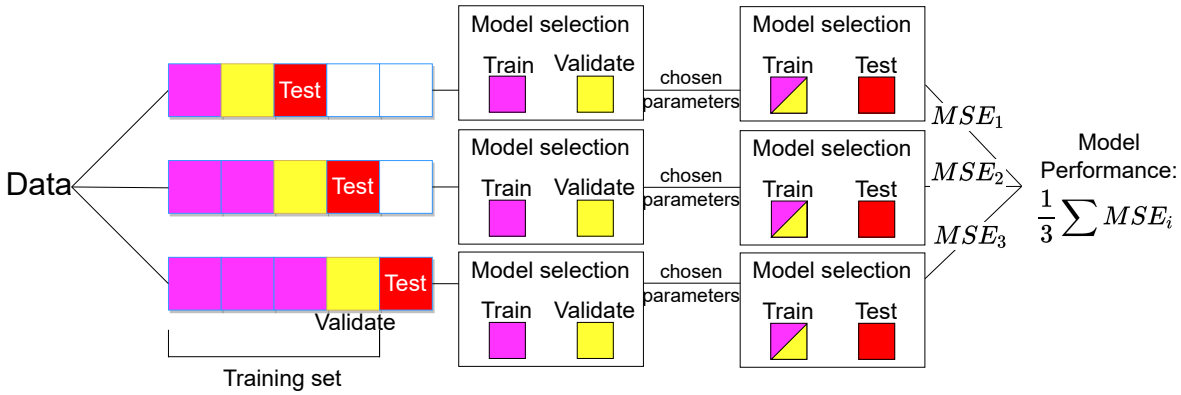


Fig. 4: Forward-chaining illustration. Creating many train/test splits and average the errors over all the splits to produce a better estimate of model prediction error.

1) LSTM: The LSTM was developed to learn order dependence in sequence prediction problems. And the results in table III show that LSTM captured the patterns but encounter difficulties against noise. All of the tested models were able to understand the seasonality in the data on a global scale. However, when trends are combined with seasonality, the LSTM encounters difficulties, as shown in Figure 2. On the product category A, the LSTM model reaches its best prediction with $MSE = 0.43$ and $MAE = 0.36$. The number of epochs has been limited to 15 for all the deep learning models. With an Early stopping [19] and Auto adjust learning rate [27] configuration enabled.

2) XGBoost: The results show that using 100 estimators with a squared error as objective function, the XGBoost model was able to capture the seasonality almost as the LSTM did, but when it comes to long term prediction, the XGBoost outperforms the LSTM, as shown in Table III. The GridSearchCV class of the sk-learn library has been used to automate the model selection. Thus, ranges of hyper-parameters have been provided to the GridSearchCV, which then returns the configuration that works best based on desired evaluation technique. The best prediction ($MSE = 0.67\%$, $MAE = 0.47\%$) was achieved with 100 estimators and a maximum depth of 9.

3) Autoformer vs Attention family: It was determined that Autoformer globally outperformed the other models in this study. Precisely because with the series decomposition blocks, Autoformer can aggregate and refine the trend-cyclical part from series progressively. It was also designed to facilitates the learning of the seasonal part, especially the peaks and troughs.

This verifies the necessity of the decomposition architecture. The best results of the Autoformer, Informer and Transformer were obtained with 8 heads when predicting on 1 day, and 30 when predicting on 7 to 60 days, with 3 encoding layers and 2 decoding layers, an ADAM optimizer, a moving average equal to 13, an attention factor equal to 3. This confirms the observation made in [30] — datasets with an obvious periodicity tend to perform better with a high factor.

VI. Conclusion

In this paper we have seen the application of Machine Learning models to predict overstocking. We addressed the overstocking prediction problem, which can be formulated as a demand forecasting problem. We compared 5 prediction approaches, including Deep Learning approaches such as LSTM, Autoformer, Informer, Transformer and a Machine Learning approach, namely XGBoost. LSTM and XGBoost were taken up and re-adapted for the regression problem and then compared to attention-based models. The best results with the MSE metric were generally observed with the Autoformer model and the best results with the MAE metric were generally observed with the Transformer model. In our future works we plan to use other external and public data such as weather to increase accuracy of the forecasting. Preliminary results are very encouraging and confirm the superiority of the Autoformer model.

Acknowledgements

This work was done as a part of a CIFRE (N 2019/1139) project with Colruyt France, funded by the Ministry of Higher Education and Research of France, managed by the Association Nationale de la Recherche et de la Technologie (ANRT) and was partially supported by the EIPHI Graduate School (contract "ANR-17-EURE-0002").

References

[1] L. Aburto and R. Weber. Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing*, 7(1):136–144, 2007.

[2] A. F. Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[3] I. Alon, M. Qi, and R. J. Sadowski. Forecasting aggregate retail sales:: a comparison of artificial neural networks and traditional methods. *Journal of retailing and consumer services*, 8(3):147–156, 2001.

[4] M. E. Aminanto, T. Ban, R. Isawa, T. Takahashi, and D. Inoue. Threat alert prioritization using isolation forest and stacked auto encoder with day-forward-chaining analysis. *IEEE Access*, 8:217977–217986, 2020.

[5] A. P. Ansuj, M. Camargo, R. Radharamanan, and D. Petry. Sales forecasting using time series and neural networks. *Computers & Industrial Engineering*, 31(1-2):421–424, 1996.

[6] S. Aras, İ. Deveci Kocakoç, and C. Polat. Comparative study on retail sales forecasting between single and combination methods. *Journal of Business Economics and Management*, 18(5):803–832, 2017.

[7] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[8] C. Bergmeir and J. M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

[9] J.-H. Böse, V. Flunkert, J. Gasthaus, T. Januschowski, D. Lange, D. Salinas, S. Schelter, M. Seeger, and Y. Wang. Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment*, 10(12):1694–1705, 2017.

[10] G. E. Box and D. A. Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526, 1970.

[11] T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014.

[12] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[13] T. Chou and M. Lo. Predicting credit card defaults with deep learning and other machine learning models. *International Journal of Computer Theory and Engineering*, 10(4):105–110, 2018.

[14] D. A. Darling and A. Siegert. The first passage problem for a continuous markov process. *The Annals of Mathematical Statistics*, pages 624–639, 1953.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[16] J. Gonzalez and W. Yu. Non-linear system modeling using lstm neural networks. *IFAC-PapersOnLine*, 51(13):485–489, 2018.

[17] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[19] H. Liang, S. Zhang, J. Sun, X. He, W. Huang, K. Zhuang, and Z. Li. Darts+: Improved differentiable architecture search with early stopping. *arXiv preprint arXiv:1909.06035*, 2019.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[21] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 13937–13949. Curran Associates, Inc., 2021.

[22] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[23] R. Sen, H.-F. Yu, and I. S. Dhillon. Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[24] D. Song, H. Chen, G. Jiang, and Y. Qin. Dual stage attention based recurrent neural network for time series prediction, Feb. 23 2021.

[25] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

[26] W. Tian, Y. Zhang, Y. Li, and H. Zhang. Probabilistic demand prediction model for en-route sector. *International Journal of Computer Theory and Engineering*, 8(6):495–499, 2016.

[27] Q. Tong, G. Liang, and J. Bi. Calibrating the adaptive learning rate to improve convergence of adam. *Neurocomputing*, 2022.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[29] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun. Transformers in time series: A survey, 2022.

[30] H. Wu, J. Xu, J. Wang, and M. Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34, 2021.