Proceedings of the of the 17th Conference on Computer Science and Intelligence Systems pp. 425–429
DOI: 10.15439/2022F295
ISSN 2300-5963 ACSIS, Vol. 30

Predicting the Costs of Forwarding Contracts Using XGBoost and a Deep Neural Network

Łukasz Podlodowski, Marek Kozłowski
National Information Processing Institute
Laboratory of Natural Language Processing
al. Niepodległości 188b 00-608 Warsaw, Poland
Email: lpodlodowski@opi.org.pl; mkozłowski@opi.org.pl

***Abstract*—This article presents an application of an XGBoost and deep neural network ensemble as a solution for a task assigned at the FedCSIS 2022 Challenge: Predicting the Costs of Forwarding Contracts. We demonstrate that prediction quality can be improved by combining the two approaches. We present a neural network architecture based on three independent flows. We then discuss the influence of long short-term memory units on the risk of overfitting. Finally, we show that the static XGBoost model can complement a neural network that processes dynamic data.**

***Index Terms*—XGBoost; deep neural network; LSTM; Costs of Forwarding Contracts**

I. Introduction

COST estimations are an imperative factor in business decisions in the transport sector. Researchers have demonstrated that due to a variety of dependencies, the price estimation of shipments in the shipping industry is frequently complex. Based on these studies, shipment pricing methods can be classified into two major classes: scenario-based pricing methods and algorithmic pricing. This article focuses on the former.

We concentrate on the logistical problem of predicting the costs of executing forwarding contracts. Our work relates closely to the FedCSIS 2022 Challenge: Predicting the Costs of Forwarding Contracts[1], which was organised in association with the Conference on Computer Science and Information Systems (https://fedcsis.org/). The competitors were tasked with designing high-quality methods for predicting the costs associated with forwarding contracts based on many features that describe key contract data and planned routes.

The competitors' task involved developing a predictive model that assessed the actual costs of individual orders as accurately as possible. Such models will be used in the future to support freight forwarders in the selection of profitable contracts.

II. Related work

The need for transportation arises from the need to move goods from one place to another in line with consumer demand. Sea transportation is one of the cheapest and oldest modes of transportation of goods. During the last twenty years, seabourne trade has accounted for approximately 80% of the world's and 90% of developing countries' total trade volume. According to recent statistics, since the 1980s, global seaborne trade has increased in size almost threefold, and containerised trade has increased also [1]. The goods delivered to major sea ports are next distributed by road or by rail. The global weight of loads in ports and the density of road/railway link networks continues to increase steadily. This means that the problem of optimal planned deliveries has become a significant challenge.

The article of Joo, Min, and Smith [2] presents an entire framework for benchmarking freight rates based on current and historical data. This is one of the first articles to examine shipping cost differentials between different shippers and to determine their causes.

The authors of [3] address the research gap of optimal spot shipment price calculation based on current shipment demand and available shipping capacity. The authors gathered data from various sources to generate a shipping dataset for 2016–2018. They use regression and correlation analysis to quantify their research outcomes.

A fuzzy regression forecasting model was introduced in [4] to forecast demand by examining the current international air cargo market. The difficulty of such forecasting derives primarily from individuals' differing perceptions of their socioeconomic environments and their competitiveness when evaluating risk. The main purpose of using fuzzy regression is to resolve these uncertainties and specificities while accommodating individuality.

Many long-term and long-distance transportation services are offered now via various types of auction; firms in the sector must deliver competitive prices if they want to win the competitions. The article of Nataraj et al [5] explores the application of forecasting and statistical learning methods to enhance the competitiveness of firms when applying for tenders. The authors use time series analysis to: (i) forecast the long-term cost of logistics services; and (ii) construct 'risk-aware' intervals for the prices to be offered in bids.

Yang and Mehmed [6] adopt an artificial neural network to forecast shipping freight rates. The key objective of their work is to improve the forecasting accuracy of traditional time series analysis. They evaluate the accuracy of their forecasting models using the mean square error with historical data. The authors used two different dynamic artificial neural network models, NARNET and NARXNET, and compared their per-

[1] https://knowledgepit.ml/fedcsis-2022-challenge/

formance. The experimental results suggest that, in general, NARXNET outperforms NARNET in all forecast horizons. This reveals the importance of the information contained in forward freight agreements in improving forecasting accuracy.

In 2022, Adrian Viellechner defended his PhD dissertation [7] on the application of machine learning models to transportation forecasting. First, Viellechner analyses the container shipping industry to predict delays of vessels. In his second study, he presents how machine learning methods can be applied to predict spot rates in container shipping. With accuracy of 89%, his forecasts support the decision-making of various shipping players in the negotiation of transportation contracts. By proposing prediction solutions that feature high accuracy, robustness, and applicability in practice, Viellechner's dissertation demonstrates that machine-learning-based solutions enhance effectiveness in transportation.

III. Problem description

The problem involves predicting the current costs of individual orders using detailed information, such as the type of order, the basic characteristics of the shipped goods (e.g. dimensions, special requirements), and the expected route that a driver would traverse. Using machine learning approaches, our goal is to create an accurate regressive method for predicting the costs associated with forwarding contracts; one that is based on contract data and planned routes. We evaluate the regression models using the root mean square error measure.

IV. Data

The accuracy of the forwarding contract cost prediction method depends heavily on the quality and quantity of the training data.

The datasets made available to the competitors contained six years of order history that appeared on the transport exchange, along with details such as the type of order, the basic characteristics of the shipped goods (e.g. dimensions, special requirements), and the expected route that a driver would traverse. More details about competition and data are presented in [8].

V. Data processing

A. Data preprocessing

The initial step of the data's preprocessing involved translating categorical features into one-hot encoding. We set the minimum threshold for any categorical value at 2,500; if the value occurred fewer times than this threshold, we translated it as 'unknown'. We used a regex-based method to extract minimal and maximal temperatures. When information about temperature was unavailable, we filled it in with a constant value of 35. We assumed that high temperatures did not increase contract costs, and that 35 was a neutral value. Since the dataset does not contain this value in the temperature column, it allows the model to handle situations when the absence of temperature data is relevant for undisclosed reasons. Our future experiments showed temperature information increased prediction quality. Based on date information associated with journey start dates, we matched fuel prices and added them as an additional column to each row.

We will refer to the dataset outlined above as our base set. We adopted a different approach for the neural network and XGBoost models to use the additional data included in the sequences of steps corresponding to each row from the base set. For XGBoost, we aggregated additional information, such as how many times an external fleet was used in the transport process, or how many times a specific country was visited. For the neural network, we removed some columns from the sequence data, such as cargo hold height and width, and treated this data as a fixed-size window of twelve steps. Approximately 0.4% of the sequences included in training set were longer, which forced us to cut off the sequences' tails to a fixed twelve-length size. Approximately 99.6% of sequences were shorter; to solve this difficulty, we padded the sequences with zero vectors. We choose sequences size respect to balance between computation complexity and prediction quality. We found out based on the cross-validation that lower sequence size increased prediction error.

Our first experiments revealed large differences between the cross-validation results and the preliminary scores. This led us to the conclusion that our models were overfitted. These differences disappeared, however, after we disabled the shuffling of the data during the cross-validation procedure. We supposed that the dates included in the training set could lead to the model becoming overfitted.

We assumed that date relations with target values would suppress the recognition of other patterns. To avoid this situation in our subsequent experiments, we prepared an additional set with the date information removed to make it more difficult for the model to build a relation between start times and predicted expenses.

VI. Prediction models

We used two different models: XGBoost and a deep neural network(DNN). We intended to compare the approaches and to verify whether the two models could be complementary in the regression problem embedded in sparse space. Significant differences exist in the training data prepared for each model. XGBoost naturally models static data and cannot handle dynamic data directly. For this reason, all information from the sequences is aggregated then presented as scalar values for each row from the base set. This method introduces the risk of eliminating important information from the data included in the order of steps in a sequence.

The deep neural network can handle dynamic data using a recurrent approach. We used popular long short-term memory units, which are known for their high efficiency and have proved their worth in a wide range of fields, including machine translation [9], financial market forecasting [10] and air quality prediction [11].

Long short-term memory is commonly used with one-hot encoding in various problems, such as electric load forecasting [12] and the construction of intelligent agents that play video games [13]. We used long short-term memory to enable the

models to extract important information from the order of sequences, encode it in a static space, and use it with the static data included in the base set.

We encoded the date as a number of minutes starting from 1st January 2016. We allowed XGBoost to operate on date information; we removed this information from the dataset of the deep neural network, however. For the final prediction, we used the strategy of split data set into 75% train and 25% test. For XGBoost this split was done randomly but for a DNN we used the oldest 25% of data as a validation set.

A. XGBoost

From a theoretical perspective, tree-based methods should naturally fit to data with mixed domain values. This suggests that any ensemble of trees is an effective choice for concatenate categorical features with other domains, such as the distance between two cities or temperature. Tree-based methods are limited to static data, which could lead to the loss of important information from the sequences. We accepted that risk and selected XGBoost as our base model.

Our first experiments were based on a fourfold cross-validation method. We achieved an impressive root mean square deviation of 0.1324. However, our preliminary result, evaluated on a small part of the test set, was 0.4047, which was significantly worse. In this paper we call this model XGBoost CV-optimized. We observed that the root mean square deviation based on cross-validation increased to 0.158 when we disabled data shuffling. This suggests that the data contained some undisclosed information associated with time, which led to overfitting. From this point, we based each step only on 75% of the data; we used the remainder as a validation set for the training procedure. The experimental values are presented in Table I.

To prevent overfitting, we explored the possibility of increasing the model's regularisation parameters. We decreased subsampling to 0.8, which forced the model to randomly skip some rows in training iterations. We also increased the regularisation λ parameter, which corresponds to L2 regularisation. This approach forced our model to become more conservative and less likely to overfit. Adjusting both parameters decreased the gap between the final score and the cross-validation result. Our experiments failed to reveal any opportunity to increase the α parameter, which corresponds to L1 regularisation without significantly lowering the quality of the model's predictions.

To find a reason for the high error of XGBoost CV-optimized predictions, we compared feature importance tables of our models. We recognized that the CV-optimized model increased the importance of specific categorical values. The biggest difference was the importance of the 'unknown' *prim ferry line* value generated by the procedure described in V. Importance for this value increased from 0.0001 to 0.08. We also noticed that XGBoost CV-optimized tended to be more confident on every localization feature corresponding to country code like *route start country* or *first unload country*.

B. Deep neural network

Our second model was based on a neural network. We decided to allow fully-connected layers to encode the inputs before we concatenated them into fully-connected layers operating on the whole data. The input was divided into three independent flows:

1) localisation data: static data from the base set, including all information about geolocalisation associated with route starts and ends, as well as first loadings and last unloadings. These vectors contained latitudes, longitudes, and country codes.

2) general data: static data from the base set, including all information not included in the localisation data.

3) sequence data: vectors that represent fixed-size sequences of route steps. Applying long short-term memory units enabled us to handle dynamic data and to transform sequences to static fully-connected rectifier layers.

To prevent overfitting and the local minima trap, we added small Gaussian noise layers to the inputs. These layers guarantee the training process to be out of balance, which prevents the local minima trap.'The amount of information in a weight can be controlled by adding Gaussian noise and the noise level can be adapted during learning to optimise the trade-off between the expected squared error and the information in the weights' [14]. Adding noise to the input also enabled us to control the degree of fitting to the data. Additionally, we used dropout with $p = 0.4$ to each layer, which decreased the probability of the model's overfitting [15]. The network architecture is presented in Figure 1.

For the training network, we used the Adam optimiser [16]. Adam automatically adapts parameters to training, which enabled us to avoid the arduous process of training tuning on the mixed domains. We repeated the training process after the competition ended to gain greater insight into the deep neural network's results. For this training, we used all of the testing data as a validation set. We present a chart of root mean square errors during the training process in Figure 2. We can observe that the training process is highly chaotic initially, and that the validation root mean square errors rapidly reduce to $0.158 - 0.16$ before stabilising. It is noteworthy that the chart begins from the twentieth epoch. The best result achieved in this experiment was 0.1564.

To analyse the influence of the sequential data flow processed using long short-term memory units, we prepared an additional deep neural network with similar architecture, but without the sequential data flow. The error observed in the training process is presented in Figure 3. We made two key observations associated with the overfitting risk:

- The deep neural network with long short-term memory tends, as it continues to train, to enlarge the gap between training and the validation data much faster than the network without sequential data flow. This could be the result of the network's increased capacity, or could suggest that information about the order of the sequences misdirects the training process. Both reasons lead easily to overfitting.

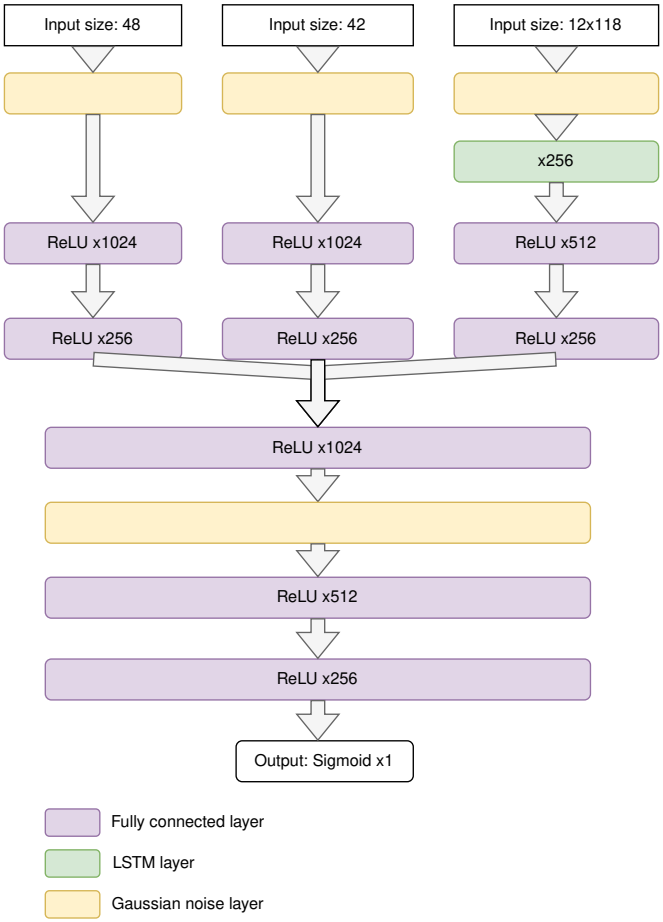

Fig. 1. The architecture of the deep neural network used as one of the solutions to a defined problem.

- The deep neural network without long short-term memory tends to gain and lose the optimisation target in each epoch in an increasingly synchronised manner. The shapes of the training and validation error curves correlate more closely on this network.

We conclude that long short-term memory would extract important information from the dataset and increase prediction quality but should be used carefully because of the overfitting problem.

VII. Results

To generate the final prediction, we used an ensemble of the XGBoost and deep neural network models as the arithmetic mean of both predictions. As illustrated in Table I and Figure 4, this ensemble reaches the best result of 0.1498. This root mean square error value is significantly better than the results of the individual models, demonstrating clearly that the deep neural network and XGBoost extracted different information from the complementary data. In the training process, we used the splitting data strategy described in VI. This strategy lets us have better control over the overfitting.

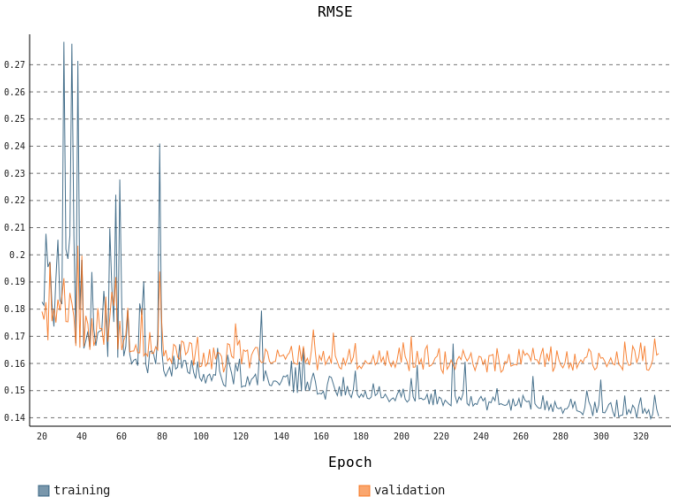

Fig. 2. The root mean square errors during the deep neural network's training process. The blue series represents error on the training data; the orange series represents error on the whole validation set—which corresponds directly to the final scores of the competition.

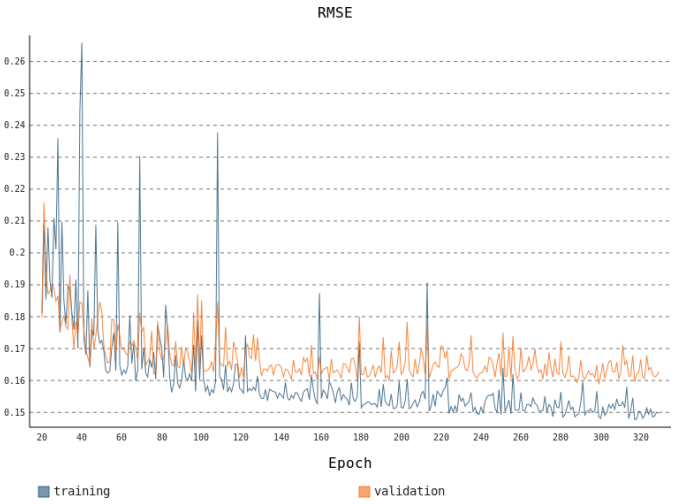

Fig. 3. The root mean square errors during the training process of the neural network without sequential data flow. The blue series represents error on the training data; the orange series represents error on the whole validation set—which corresponds directly to the final scores of the competition.

Our cross-validation-optimised XGBoost model reaches a root mean square error of only 0.404, which is significantly worse than the other models. We failed to identify precisely which factor caused the overfitting. Future experiments that focus on the analysis of smaller parts of the dataset should be performed to gather detailed information on this problem.

VIII. Conclusions

This article presents our solution to the FedCSIS 2022 Challenge: Predicting the Costs of Forwarding Contracts. During our experiments, we identified an overfitting problem, which had a significant impact on subsequent steps. We prepared two approaches: XGBoost and a deep neural network. We demonstrated that the application of a long short-term memory

TABLE I
EXPERIMENTAL RESULTS: PREDICTION QUALITY ROOT MEAN SQUARE ERRORS BASED ON CROSS-VALIDATION AND PRELIMINARY SCORES

Model	CV	Preliminary	Final score
XGBoost	0.153	0.1522	0.15252
XGBoost CV-optimized	**0.1324**	0.4047	0.40394
Deep Neural Network	0.1581	0.1585	0.15815
Ensemble	-	**0.1500**	**0.14978**

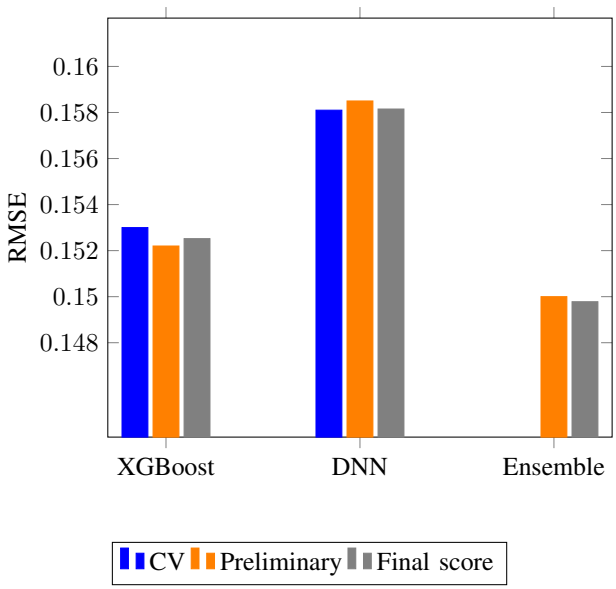

Fig. 4. Root mean square errors based on three experiments: fourfold cross-validation (blue), preliminary scores, and final scores.

layer to sequential data improves the results; however, it also increases the probability of overfitting. We demonstrated that combination of the deep neural network and XGBoost is potentially complementary, significantly increases prediction quality, and serves as a possible solution for mixing dynamic data with the static XGBoost approach.

REFERENCES

[1] Z. H. Munim and H.-J. Schramm, "Forecasting container shipping freight rates for the far east–northern europe trade lane," *Maritime Economics & Logistics*, vol. 19, no. 1, pp. 106–125, 2017.

[2] S.-J. Joo, H. Min, and C. Smith, "Benchmarking freight rates and procuring cost-attractive transportation services," *The International Journal of Logistics Management*, 2017.

[3] A. Ubaid, F. Hussain, and J. Charles, "Modeling shipment spot pricing in the australian container shipping industry: case of asia-oceania trade lane," *Knowledge-based systems*, vol. 210, p. 106483, 2020.

[4] T.-Y. Chou, G.-S. Liang, and T.-C. Han, "Application of fuzzy regression on air cargo volume forecast," *Quality & Quantity*, vol. 45, no. 6, pp. 1539–1550, 2011.

[5] S. Nataraj, C. Alvarez, L. Sada, A. Juan, J. Panadero, and C. Bayliss, "Applying statistical learning methods for forecasting prices and enhancing the probability of success in logistics tenders," *Transportation Research Procedia*, vol. 47, pp. 529–536, 2020.

[6] Z. Yang and E. E. Mehmed, "Artificial neural networks in freight rate forecasting," *Maritime Economics & Logistics*, vol. 21, no. 3, pp. 390–414, 2019.

[7] A. M. Viellechner, "The new era of predictive analytics in container shipping and air cargo," Ph.D. dissertation, WHU-Otto Beisheim School of Management, 2022.

[8] A. Janusz, A. Jamiołkowski, and M. Okulewicz, "Predicting the costs of forwarding contracts: Analysis of data mining competition results," in *Proceedings of the 17th Conference on Computer Science and Intelligence Systems, FedCSIS 2022, Sofia, Bulgaria, September 4-7, 2022*. IEEE, 2022.

[9] G. Tiwari, A. Sharma, A. Sahotra, and R. Kapoor, "English-hindi neural machine translation-lstm seq2seq and convs2s," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020. doi: 10.1109/ICCSP48568.2020.9182117 pp. 871–875.

[10] A. H. Bukhari, M. A. Z. Raja, M. Sulaiman, S. Islam, M. Shoaib, and P. Kumam, "Fractional neuro-sequential arfima-lstm for financial market forecasting," *IEEE Access*, vol. 8, pp. 71 326–71 338, 2020. doi: 10.1109/ACCESS.2020.2985763

[11] J. Wang, J. Li, X. Wang, J. Wang, and M. Huang, "Air quality prediction using ct-lstm," *Neural Computing and Applications*, vol. 33, pp. 1–14, 05 2021. doi: 10.1007/s00521-020-05535-w

[12] A. Janusz, T. Tajmajer, and M. Świechowski, "Helping ai to play hearthstone: Aaia'17 data mining challenge," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2017. doi: 10.15439/2017F573 pp. 121–125.

[13] K. H. Kim, B. Chang, and H. K. Choi, "Deep learning based short-term electric load forecasting models using one-hot encoding," *Journal of IKEEE*, vol. 23, no. 3, pp. 852–857, 2019.

[14] G. E. Hinton and D. v. Camp, "Keeping neural networks simple," in *International Conference on Artificial Neural Networks*. Springer, 1993, pp. 11–18.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: http://dl.acm.org/citation.cfm?id=2627435.2670313

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980