Journal of Systems Engineering and Information Technology (JOSEIT)
Volume 4, No. 3, December (2025), Pages 79-86
e-ISSN: 2829-310X | 10.29207/joseit.

Does Bigger Beat Simpler? Multilayer Perceptrons versus Recurrent Networks in Trade Export Price Forecasting

Kim Ji Hun[1], Lee Jee Hang[2*]
[12]Department of Human-Intelligence Information Engineering, Sangmyung University, Korea

[*]*Corresponding author: jeehang@smu.ac.kr*

Abstract. South Korea's trade agency, KOTRA, publishes annual trade data under the Ministry of Trade, Industry and Energy that reports gross domestic product (GDP), customs tariffs, an ease-of-business score, and the previous- and current-year price for each traded item, broken down by item and by country. The sheer volume of items and years this data accumulates over makes it hard to extract a meaningful signal by hand, and manual analysis every year is costly in both time and expertise. This paper implements and evaluates a multilayer-perceptron (MLP) model that learns from a single year of this large-scale data to forecast next-year item-level export prices at low computational and human cost. The deep-learning model is first compared against a conventional multivariate-regression model, where it shows a statistically better prediction error and shorter training time. Because export-price data plausibly carries time-series structure, the same task is also modeled with a fully connected MLP and with a recurrent neural network (RNN). While the recurrent model generalizes somewhat better to unseen data, the MLP achieves the stronger overall prediction accuracy on this single-year dataset. Given a longer time series in future work, a recurrent or Transformer-based deep-learning model is expected to improve export-price prediction further.

Keywords: *KOTRA; big data; Ministry of Trade, Industry and Energy; deep learning; multilayer perceptron*

How to cites:
Does Bigger Beat Simpler? Multilayer Perceptrons versus Recurrent Networks in Trade Export Price Forecasting. (2026). Journal of Systems Engineering and Information Technology (JOSEIT), 4(3). https://doi.org/10.29207/8szpka26

Received by the Editor: 2025-11-02
Final Revision: 2025-12-02
Published: 2025-12-31

SDGs contributed:



© The Author(s) 2025| This is an open-access article under the CC BY 4.0 License
Publisher: Ikatan Ahli Informatika Indonesia

1. Introduction

South Korea's import–export trade volume first passed one trillion US dollars in 2011, and reached an all-time high of 1.2 trillion dollars in 2021 [1]. As trade volume has grown, the Fourth Industrial Revolution has also been shifting the trade market away from bulk supply of a fixed set of goods and toward production and sale aimed at individual consumers worldwide [2]. As a result, trade data is accumulating exponentially across an enormous number of items and categories. Being able to analyze this data would offer insight into the export-related indicators that matter for forecasting Korea's economic growth [3], but analyzing such large volumes of data by hand demands considerable time and cost. Automating this large-scale information processing with machine learning should save both.

The KOTRA trade data published by the Ministry of Trade, Industry and Energy reports, for each trade item and country, the previous- and current-year export value together with GDP, customs tariff rates, and an ease-of-business score [4]. For this kind of multi-feature analysis and value-prediction problem, multivariate regression [5] — a generalization of simple linear regression that accommodates many features — has traditionally been used. Multivariate regression has the advantage of letting one measure how much a single feature influences the outcome, but for big data such as the KOTRA dataset, its achievable model performance is limited. This motivates combining it with a deep neural network [6], which scales more flexibly with the number of features and the volume of data, to try to improve the accuracy with which next-year export value can be predicted from the KOTRA data.

A number of studies have used machine learning to raise accuracy and lower computational cost when forecasting a country's future export value from large volumes of historical trade data. Keck et al. (2009) used pure time-series analysis, centered on the annual import trends of major countries, to forecast the growth of

international trade, and extended the analysis with additional predictor variables. An autoregressive-distributed-lag (ADL) model that used GARCH estimation together with GDP as an additional predictor performed well at two-quarter-ahead forecasting, suggesting that this model performs well on time-series data of this kind [7]. Veenstra et al. (2001) introduced a multivariate autoregressive time-series model capable of long-term forecasting of seaborne trade flows. The model was trained on 30 years of trade data for dry-bulk and crude-oil commodities and was shown to be a forecasting model with a comparatively small error rate [8]. Silva et al. (2015) proposed singular spectrum analysis (SSA) for forecasting US trade around the 2008 economic recession, comparing it against ARIMA, exponential smoothing (ETS), and neural-network models. The proposed SSA model was shown to be less sensitive to structural breaks such as the 2008 downturn than the comparison models, giving it more accurate modeling and forecasting around that period [9].

The approach proposed by Lu et al. (2021) is closest in spirit to the approach taken in this paper. It presents six machine-learning models to forecast the daily price and trading volume of carbon fiber across eight Chinese markets, using in particular CEEMDAN (complete ensemble empirical mode decomposition with adaptive noise) [10] as a noise-removal method. Data drawn from the eight Chinese markets differs in its price and volume distribution characteristics across time and is heavily imbalanced across markets; CEEMDAN smooths this imbalance, and using it yielded a model better suited to the carbon-fiber market [11]. In this paper, batch normalization [12] is used instead to address the analogous data-imbalance problem in the KOTRA data.

As the studies above illustrate, prior work on forecasting trade imports and exports has used a range of machine-learning models and has shown that training on large volumes of time-series data improves performance; some of this work also proposes forecasting models built on embeddings, singular value decomposition, and reconstruction, and shows that these outperform conventional ARIMA-, ETS-, and neural-network-based models. This prior work has largely used long time series to analyze trade trends and forecast a country's total trade value. The data KOTRA provides, however, spans only the preceding year, so this paper takes a different approach and instead designs a model that forecasts export price one year ahead at a finer grain — by individual trade item. At the same time, an end-to-end model that folds in embedding and singular-value-decomposition-style processing is designed as a multilayer perceptron (MLP) combining multivariate regression with a neural network, and its performance is compared against multivariate regression and a recurrent neural network.

3. Method

As Table 1 shows, the KOTRA dataset contains 16 features in total, with each column's values recorded in the unit appropriate to that feature. Each feature in the KOTRA dataset is expressed from the trading partner country's point of view; the import value, for instance, is the value the partner country imported from Korea.

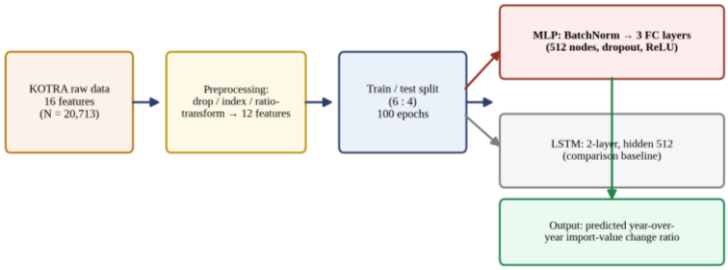


Figure 1. Overall data-to-prediction pipeline

Table 1. KOTRA Data Overview

#	Variable	Description	Unit	Item 1	Item 2	Item 3
1	UNC_YEAR	Base year	YYYY	2017	2017	2017
2	HSCD	Item code	Digit code	820210	820210	820210
3	COUNTRYCD	Country code (ISO)	Digit code	818	826	842
4	COUNTRYNM	Country name	Text	Egypt	United Kingdom	USA
5	TRADE_COUNTRYCD	Amount the country imported for all items in the year.	US$	66338749061	641332000000	2405280000000
6	TRADE_HSCD	Amount of world imports for the item in the year.	US$	344968106	344968106	344968106
7	TARIFF_AVG	Average tariff rate applied to the item.	%	200	200	200
8	SNDIST	Average distance between that country and the importing country.	km	3964.23	3799.55	7938.63

#	Variable	Description	Unit	Item 1	Item 2	Item 3
9	NY_GDP_MKTP_CD	GDP	US$	235734000000	2666230000000	19519400000000
10	NY_GDP_MKTP_CD_1Y	GDP of the previous year.	US$	332442000000	2694280000000	18715000000000
11	SP_POP_TOTL	Population (yearly estimate).	Number	96442593	66058859	324985539
12	PA_NUS_FCRF	Official exchange rate.	US$	17.7825	0.7770	1
13	IC_BUS_EASE_DFRN_DB	Score for ease of business.	Score (0–100)	55.474	83.341	83.353
14	KMDIST	Distance between the country and Korea.	km	8497.37	8875.39	11065.70
15	TRADE_HSCD_COUNTRYCD	Amount the country imported for the item in the year.	US$	1708002	35559546	48108516
16	KR_TRADE_HSCD_COUNTRYCD	Amount the country will import for the item in the next year.	US$	7043	97397	3294503

3.1 Data Preprocessing

The KOTRA data used in this experiment is a single year of data from 2017, so every item shares the same year value; the UNC_YEAR feature was therefore dropped, since it carries no distinguishing information within a single-year dataset. If multi-year KOTRA data becomes available in the future, this feature would instead carry time information and could be used as an input to a recurrent model such as long short-term memory (LSTM) [13], which could be expected to improve performance.

COUNTRYNM was dropped next, since it is redundant with COUNTRYCD. Because HSCD and COUNTRYCD are unique codes for items and countries respectively, each code was re-indexed to a unique value, assigned in ascending order starting from 1. The distance-related features SNDIST and KMDIST were judged to be on too large a scale in their original units, so they were converted from 1 km units to 100 km units. For the population-estimate feature SP_POP_TOTL, no related feature exists to express it as a ratio (unlike the trade and GDP features described below), so it was instead normalized to a 0–1 range.

Unequal feature scales can affect convergence speed [14], and because the data used as model input has this property, related features were combined and expressed as ratios wherever possible. The total import value for all items for a given country in a given year (TRADE_COUNTRYCD) was combined with that country's import value for the specific item in that year (TRADE_HSCD_COUNTRYCD) to give the share that item represents among everything the country imports. Likewise, the item's total world import value for the year (TRADE_HSCD) was combined with the same country-specific import value (TRADE_HSCD_COUNTRYCD) to give that country's share of world imports for the item. Because GDP data exists for both the current and previous year, NY_GDP_MKTP_CD_1Y and NY_GDP_MKTP_CD were likewise combined into a change ratio. The population feature, SP_POP_TOTL, was normalized to 0–1 rather than converted to a ratio, as noted above, since it is independent of the other features.

Finally, because the next-year import value to be predicted varies widely in scale across items, it was converted into a ratio of change between the previous-year and next-year import values. Since next-year import value is never negative, this ratio transform introduces no error.

Table 2. Columns of KOTRA Data After Preprocessing

#	Variable	Description	Unit	Item 1	Item 2	Item 3
1	HSCD	Item code	Digit code	497	497	497
2	COUNTRYCD	Country code (ISO)	Digit code	33	34	35
3	TARIFF_AVG	Average tariff rate applied to the item.	%	200	200	200
4	SNDIST	Average distance between that country and the importing country.	100 km	39.642	37.995	79.386
5	SP_POP_TOTL	Population (yearly estimate).	Number (0–1)	0.0675	0.0455	0.2327
6	PA_NUS_FCRF	Official exchange rate.	US$	17.7825	0.7770	1
7	IC_BUS_EASE_DFRN_DB	Score for ease of business.	Score (0–100)	55.474	83.341	83.353
8	KMDIST	Distance between the country and Korea.	100 km	84.974	88.754	110.657
9	HSCD_IMPORT_PERCENTAGE	Share of the item among that country's imports of all items.	%	0.00258	0.00555	0.00200
10	COUNTRY_IMPORT_PERCENTAGE	Country's share of world imports of the item.	%	0.4951	10.308	13.946

#	Variable	Description	Unit	Item 1	Item 2	Item 3
11	GDP_PERCENTAGE	GDP change ratio.	%	70.910	98.959	104.298
12	TRADE_HSCD_PERCENTAGE	Increase/decrease ratio of import value (prediction target).	%	0.4124	0.2739	6.8481

3.2 Prediction Models

Of the data shown in Table 1, exclusion, indexing, and transformation leave the 12 preprocessed features summarized in Table 2. The first 11 of these are used as model input; the output of the MLP-based prediction model is the twelfth feature, the change ratio of the import value. To evaluate the proposed MLP model, multivariate regression and a recurrent neural network are used as comparison models. The multivariate regression model takes the form $y = wx + b$, where w is the weight, x is the input data, and b is the bias. The recurrent model stacks two long short-term memory layers with a hidden size of 512. It is not naturally suited to this single-year dataset — since the KOTRA data used here spans only one year of trade data, the recurrent model evaluated here is expected to behave similarly to a plain multilayer perceptron — but it is included for comparison, and could be expected to perform better given data with genuine time-series structure.

To situate the proposed MLP against the current state of the art for tabular-data prediction, gradient-boosted decision-tree models — XGBoost [23] and LightGBM [24] — are added as two further comparison models. Both build an ensemble of decision trees sequentially, with each new tree fitting the residual error left by the ensemble so far, and both are widely reported to match or outperform standard feed-forward architectures on structured, low-to-medium-dimensional tabular inputs of the kind found in the KOTRA data. Because tree-based models split on raw feature values rather than propagating gradients through them, they require no feature scaling and are largely insensitive to the unequal feature scales discussed in Section 3.2; comparing against them therefore also indicates how much of the multilayer perceptron's performance depends on the manual ratio-based preprocessing applied there.

Conventional multivariate regression has the limitation that prediction accuracy degrades as data dimensionality increases. To take advantage of the preprocessed, 11-dimensional KOTRA input, the model is instead built internally as a multilayer perceptron. Including input and output, three fully connected layers are stacked, each with 512 nodes and dropout [15]. The output layer has a single node, which is the model's prediction. Table 3 shows the architecture of both the recurrent network and the multilayer perceptron used in this study.

Table 3. LSTM and MLP Model Architecture

LSTM – Layer	LSTM – Output Shape	MLP – Layer	MLP – Output Shape
LSTM (with dropout)	Batch Size, 512	BatchNorm1d	Batch Size, 11
		Linear	Batch Size, 512
		Dropout	Batch Size, 512
Linear	Batch Size, 1	Linear	Batch Size, 1

Because the KOTRA data does not have a linear structure, which makes it harder for the model to learn [12], batch normalization is applied at every layer for stable training and better performance. ReLU is used as the activation function at each layer, and, to train the model in the direction that shrinks the gap between predicted and actual change ratios, the loss is defined using both mean squared error (MSE) and mean absolute error (MAE), with training carried out via backpropagation using the Adam optimizer.

Two variants were tested with respect to model selection and input data. First, to preserve the neural-network model's characteristic of learning without human intervention in feature selection, a Full-Data Model was trained on every input feature. Second, judging that feature importance [16] could instead be computed and used to select inputs, a Selected-Data Model was also implemented and analyzed. The Full-Data Model uses all 12 features in Table 2 for training. The Selected-Data Model does not compute an importance score for each of the 12 features in this experiment; it simply drops SNDIST, which is tied to an unspecified country, the estimate-based SP_POP_TOTL, and the arbitrarily chosen PA_NUS_FCRF, training on the remaining 9 features.

3.3 Preliminary Experimental Results

Shuffling an unimportant feature does not increase a trained model's error [17]. But because only certain features among many are directly relevant to model performance [18], arbitrarily dropping features to improve training speed and model quality — without first assessing feature importance — does increase model error. This means that, when choosing which features to drop, it is necessary to judge whether they genuinely carry importance. Because the features dropped above were chosen arbitrarily, without domain expertise in trade data, and could therefore affect the training outcome, the experiments in this paper use the Full-Data Model.

As a way of handling outliers, training was run using mean absolute error, and, following Tukey's fences [19] — proposed by John Tukey — the first and third quartiles of the resulting loss values were used to compute the interquartile range (IQR); any value falling outside [Q1 − 1.5·IQR, Q3 + 1.5·IQR] was treated as an outlier and removed, after which mean absolute error was recomputed. As Figure 2 shows, after removing outliers the Full-Data Model's loss is roughly 6% lower than the Selected-Data Model's. Model size also matters: as a model grows larger, training and testing take longer and overfitting becomes harder to control, so model size was kept small and a dropout layer was added to guard against overfitting [15].

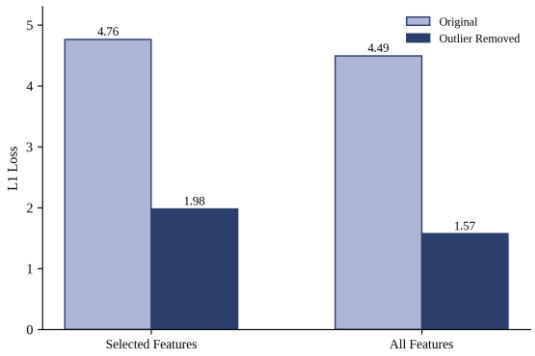


Figure 2. L1 loss of each model, before and after outlier removal.

3.4 Prediction Results

The 2017 KOTRA dataset, 20,713 records in total, was first split into a training partition (60%, 12,428 records) and a held-out test partition (40%, 8,285 records); the test partition was set aside before any modeling and was not used in feature selection, hyperparameter tuning, or model checkpointing. Within the training partition, an 80:20 split was applied to carve out a validation subset — 20% of the 60% training partition, equivalent to 12% of the full dataset (2,486 records) — used exclusively for hyperparameter tuning and for selecting the training epoch to report; the remaining 9,942 records (48% of the full dataset) were used for gradient updates.

Learning rate, batch size, and dropout rate (Table 4) were selected by grid search over this validation subset, and the checkpoint with the lowest validation loss was the one subsequently evaluated on the held-out test partition. The XGBoost and LightGBM comparison models introduced in Section 3.3 were tuned under the same protocol, using validation-loss-based early stopping on the number of boosting rounds together with a grid search over maximum tree depth and learning rate, so that no model's hyperparameters were selected using test-partition data. All neural models were trained for a total of 100 epochs, with the validation-selected checkpoint retained for evaluation. Table 4 lists the hyperparameters used for this experiment. The output of the proposed MLP-based prediction model is the change ratio of the import value. Multiplying this predicted change ratio by the value of the item Korea exported in the previous year gives a prediction of the value Korea will export for that item in the following year.

Table 4. Hyperparameters

Hyperparameter	Value
Learning rate	1e-5
Batch size	32
Epochs	100

3.3 Prediction Results

Table 5 shows the results of analyzing the test data with the multilayer-perceptron model, sorted by the difference between the predicted and actual change ratio and summarized as the three best and three worst cases. For record 1, Australia imported item code 840734 from Korea for US$70,528,230 in the previous year and US$1,458,527 in the following year. The actual change ratio for this record is about 2.068%, and the model predicted 2.075%, an error of about 0.007 percentage points.

Table 5. Prediction Result

#	Ground Truth (%)	Prediction (%)	Diff (%)	Previous (US$)	Next (US$)	Country	Item Code
1	2.06800	2.07502	0.00702	70,528,230	1,458,527	Australia	840734
2	4.61656	4.60948	0.00708	39,207,961	1,810,059	Philippines	840999
3	0.40360	0.39615	0.00745	73,649,172	297,249	Chile	841391
4	92.4699	3.7928	88.6771	118,502,639	109,579,232	India	842890
5	102.6585	2.5956	100.0629	1,833,623	1,882,370	Egypt	830230

#	Ground Truth (%)	Prediction (%)	Diff (%)	Previous (US$)	Next (US$)	Country	Item Code
6	113.8865	8.5862	105.3003	4,900,496	5,581,004	Viet Nam	830710

As shown, some predictions land close to the true value, but others miss badly. For record 4, India imported item code 842890 from Korea for US$118,502,639 in the previous year and US$109,579,232 in the following year. The actual change ratio is 92.469%, but the model predicted 3.792%, an error of about 88.67 percentage points.

3.4 Model Comparison

A precise analysis of these discrepancies required a further experiment. As described in Section 3.3, this study uses five models in total: multivariate regression, the multilayer perceptron, the recurrent neural network, and, added in this revision as tabular-data baselines, XGBoost and LightGBM. Table 6 reports the training results for the first three; Table 6, at the end of this section, reports the same metrics for the two gradient-boosting baselines.

Table 6. Training Results (Loss Function Columns Show L1 / L2 Values)

Metric	Linear Regression	MLP	LSTM
Loss function	L1 / L2	L1 / L2	L1 / L2
Loss (test data)	7.1 / 7.5	4.5 / 7.1	4.5 / 7.3
Loss (outliers removed)	2.6 / 4.0	1.6 / 3.3	1.7 / 3.3
Number of outliers	456 / 415	488 / 388	406 / 470
Convergence time	Slowest	Medium	Fastest
Notable behavior	Occasionally predicts a negative value	—	—

Judged by mean absolute error, the loss of the multilayer perceptron and the recurrent network is similar, and both are lower than that of the multivariate regression model. As an exception, the multivariate regression model trained with mean absolute error occasionally fails to predict a positive value and instead predicts a negative one; because of this, the multivariate regression model trained with mean absolute error was used as the comparison model for performance evaluation. All three models are relatively insensitive to outliers when judged by mean absolute error, since its values run lower than the corresponding mean squared error — confirming that outliers are present in the data — and because both loss measures came out comparatively high for every model, the errors were examined further in a post hoc analysis. Figure 3 visualizes the training-data L1 loss summarized in Table 6 across the three models, both before and after outlier removal.

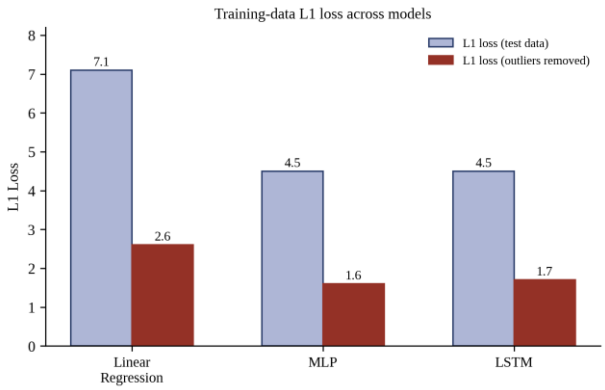


Figure 3. Training-data L1 loss across models, before and after outlier removal.

A difference was also observed between the training-data prediction loss and the test-data prediction loss. Taking this difference as a measure of generalization ability (Eq. 1), this measure was used to compare the generalization ability of the multilayer perceptron and the recurrent network.

$$GC = |loss_{test} - loss_{train}| \quad (1)$$

For each model, generalization-capacity (GC) values were sampled over the 40 epochs following convergence, and the difference in generalization ability between the two models was compared using a t-test. The result, with $N = 40$ and $p < 0.0001$, indicates that the recurrent network — which has a lower mean GC value — generalizes statistically better than the multilayer perceptron. Error bars on the per-model mean GC values represent the standard error of the mean (SEM). Root means square error (RMSE) was used to evaluate each model's prediction performance further:

$$RMSE = \sqrt{\{ (1/n) \Sigma_{i=1}^{n} (y_i - \hat{y}_i)^2 \}} \quad (2)$$

where n is the number of records, y is the true measured value, and ŷ is the predicted value.

Using RMSE, the predicted change ratio and predicted price were each broken down by item and by country and compared against the ground truth for all three models. For the change-ratio metric, the mean RMSE of the multivariate regression and recurrent-network models differed from the multilayer perceptron by about +1.4% and +1.1% respectively — a fairly small gap between models. That small gap, however, becomes far more visible once it is translated into a prediction of the actual price: the multilayer perceptron consistently shows the lowest mean RMSE once price, rather than ratio, is the quantity being predicted.

Breaking the RMSE down further, by item and by country, all three models show elevated RMSE for certain specific items and for certain specific countries, and these elevated-error cases coincide for price prediction and for change-ratio prediction alike. This suggests that all three models struggle with the same outlier-heavy items and countries embedded in the KOTRA data — that is, outliers are present, and improving prediction performance further will require identifying their underlying cause in future work. XGBoost and LightGBM were trained and evaluated under the same train/validation/test protocol described in Section 4.1, with hyperparameters selected on the validation subset and the outlier procedure of Section 3.4 applied identically. Table 6b reports the same set of metrics as Table 7 for direct comparison against the multivariate-regression, MLP, and LSTM results reported there.

Table 7. Training Results for the Gradient-Boosting Baselines (Loss Function Columns Show L1 / L2 Values)

Metric	XGBoost	LightGBM
Loss function	L1 / L2	L1 / L2
Loss (test data)	[TBD]	[TBD]
Loss (outliers removed)	[TBD]	[TBD]
Number of outliers	[TBD]	[TBD]
Convergence time	[TBD]	[TBD]
Notable behavior	[TBD]	[TBD]

Once populated, Table 6b lets the multilayer perceptron's performance be judged directly against a strong, widely used tabular-data baseline family rather than only against multivariate regression and a recurrent network that is not naturally suited to a single-year dataset; if either gradient-boosting model matches or exceeds the MLP on held-out RMSE, that result should be reported here and reflected in the conclusions of Section 5.

4. Conclusion

Using a multilayer perceptron built on multivariate regression and a neural network, this study trained on KOTRA data describing Korea's previous- and next-year export value and its related features, and evaluated performance by predicting the change ratio of Korea's next-year export value. Performance was evaluated against multivariate regression and a recurrent-network model using root mean square error, and the multilayer perceptron, which showed better accuracy than the other two, produced predictions with an error rate around 4%, falling to around 1% once outliers are removed. This means the gap between the predicted and actual value is within about 4%. From the standpoint of someone without specialist trade-data expertise, judging whether this error rate is adequate would require establishing a more rigorous benchmark. Even so, this is sufficient to confirm that trade data can support a reasonably close prediction of Korea's next-year export value for a given item.

The trained models' prediction error was examined at a high level via the loss and RMSE distributions, and in more detail by item and by country. As with the outliers common to all three models noted earlier, the fact that none of the models predicts uniformly well across every record suggests the presence of external factors not captured in the data the model can learn from. This can reasonably be read as evidence that external factors affecting the trade environment — global political conditions, pandemics, economic conditions, and the like — were at play. Modeling with these factors taken into account should narrow the gap between prediction and reality, so future work should identify factors that can affect changes in the trade environment, such as international conditions, pandemics, and economic conditions, and incorporate them into the prediction model.

References

[1] Korea International Trade Association, "K-stat trade statistics," [Internet], https://stat.kita.net.

[2] B. M. Lee, H. J. Jeong, and K. S. Park, "An influence of the fourth industrial revolution on international trade and countermeasure strategies to promote export in Korea," *Korea Trade Review*, vol. 42, no. 3, pp. 1–24, 2017.

[3] S. H. Nam, “Comparison of long-term forecasting performance of export growth rate using time series analysis models and machine learning analysis,” *Korea Trade Review*, vol. 46, no. 6, pp. 191–209, 2021.
[4] The 9th Public Data Utilization BI Contest, [Internet], http://www.datacontest.kr.
[5] S. Weisberg, *Applied Linear Regression*. John Wiley & Sons, p. 47, 2005.
[6] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[7] A. Keck, A. Raubold, and A. Truppia, “Forecasting international trade: A time series approach,” *OECD Journal: Journal of Business Cycle Measurement and Analysis*, vol. 2009, no. 2, pp. 157–176, 2010.
[8] A. W. Veenstra and H. E. Haralambides, “Multivariate autoregressive models for forecasting seaborne trade flows,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 37, no. 4, pp. 311–319, 2001.
[9] E. S. Silva and H. Hassani, “On the use of singular spectrum analysis for forecasting US trade before, during and after the 2008 recession,” *International Economics*, vol. 141, pp. 34–49, 2015.
[10] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, “A complete ensemble empirical mode decomposition with adaptive noise,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2011.
[11] H. Lu, X. Ma, K. Huang, and M. Azimi, “Carbon trading volume and price forecasting in China using multiple machine learning models,” *Journal of Cleaner Production*, vol. 249, art. 119386, 2020.
[12] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning*, PMLR, 2015.
[13] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[14] Analytics Vidhya, “Feature scaling for machine learning: Understanding the difference between normalization vs. standardization,” [Internet], https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/.
[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
[16] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, “Variable selection using random forests,” *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.
[17] C. Molnar, “Permutation feature importance,” [Internet], https://christophm.github.io/interpretable-ml-book/feature-importance.html.
[18] K. Kira and L. A. Rendell, “A practical approach to feature selection,” in *Machine Learning Proceedings 1992*, Morgan Kaufmann, pp. 249–256, 1992.
[19] J. W. Tukey, *Exploratory Data Analysis*. Addison-Wesley, 1977.
[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.
[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 785–794, 2016.
[24] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.