Terbit online pada laman web jurnal : https://publikasi.mercubuana.ac.id/index.php/fasilkom/

Collabits Journal

Vol 1, No 1 (2024) ISSN :

Transfer Learning Implementation on Bi-LSTM with Optimizer to Improve Non-Ferrous Metal Prices Prediction

Adji Pratomo[1], Muhammad O. Jatmika[2], Bedine Kerim[3], Rahmat Budiarto[4*]

[1]Faculty of Computer Science, Universitas Mercu Buana, Jakarta, 11650, Indonesia, *41518210036@student.mercubuana.ac.id*
[2]Faculty of Computer Science, Universitas Mercu Buana, Jakarta, 11650, Indonesia, *41518210001@student.mercubuana.ac.id*
[3]College of Computing and Information, Albaha University, Albaha, 65779-7738, Saudi Arabia, *bkerim@bu.edu.sa*
[4]College of Computing and Information, Albaha University, Albaha, 65779-7738, Saudi Arabia, *rahmat@bu.edu.sa*

Abstract

Over the past few years, the implementation of renewable energy or go-green has intensified along with the rapid development of its technology and increasing uncertainty of natural conditions that cause the prices of non-ferrous metals such as copper, aluminum, nickel, etc. used as main components for developing renewable energy devices, e.g.: battery, experience instability price in the commodity futures market. Economic players who trade metals in the futures market certainly need to be careful and must evaluate the state of the world economy. This study proposes a prediction engine as a combination of Bidirectional Long-Short Term Memory (BiLSTM), with three optimization algorithms, i.e.: Adam, Root Mean Squared Propagation (RMSProp), and Stochastic Gradient Descent (SGD), and transfer learning to make model training better. Experiments on four historical data on nickel, lead, aluminum and copper prices in the commodity futures market are conducted. The selected features are: open price, close price and volume price. Twelve models will be created to find the model that best predicts the metal prices. The top 3 models with the best performance were selected, they are: model 4 RMSProp with R2 value of 0,99029 and MSE 0,00076 as the first ranking, model 3 Adam with R2 value of 0,98877 and MSE 0,00074 as the second ranking, and model 4 Adam with value of R2 0,98522 and MSE 0,00115 as the third ranking.

Keywords : *Metals commodity, Price Prediction, BiLSTM, Optimizer, Transfer Learning*

1. Introduction

Over the past few years, the implementation of renewable energy or go-green has intensified along with the rapid development of its technology and increasing uncertainty of natural conditions that cause the prices of raw metal materials such as copper, aluminium, nickel, etc. used as main components for developing renewable energy devices, e.g.: battery, experience instability price in the commodity futures market. Economic players who trade metals in the futures market (as buyer or also as seller) certainly need to be careful and must evaluate the state of the world economy, thus, need sophisticated tool to observe the dynamics of the price's fluctuation as well as to predict the prices in the future to optimize the profit or to minimize the loss if any.

Time Series Forecasting/Prediction is one area in machine learning that focuses on time series attributes, more specifically, on sequential time series data analysis, then predicting the future outcomes based on previous available data. Prediction using deep learning approach such as BiLSTM on big data and relatively huge number of epochs definitely requires long processing time and consumes intensive computing resources during the model development stage, i.e.: for training as well as testing of the model, and when the dataset changes, the developed model needs to be retrained again since the model does not keep the interconnection weights or weight values in each neuron. Therefore, a prediction model that can be reused without losing its interconnection weights and need shorter training time is required.

To address the above mentioned issues, this study proposes a prediction engine as a combination of BiLSTM with three optimization algorithms, i.e.: Adam, RMSProp, and SGD, and transfer learning to make model training better. Then experiments on four historical datasets, i.e.: nickel, lead, aluminum and copper prices datasets in the commodity futures market are carried out. The use of four different datasets makes the developed models more robust.

This study contributes towards the development of robust model to be used as accurate prediction engine that can deal with big data. In addition, the model assists the traders in the metals commodity futures market to minimize the business risk. A good prediction of metals business prospect, will attract more capital ventures to invest in the advancement of renewable energy/ go-green technology.

2. Literature Review

Prediction in commodity trading is often used as a tool in planning, whether it is done by individuals or companies. One method that can be used for prediction is data mining [1]. Previous studies [2] define data mining as the process of extracting and processing data into very important and useful information that may not have been known before. With different conditions of data when prediction, there is no one method that can provide absolute prediction accuracy and perfect performance. The data mining approach every so often used to make predictions is Deep Learning [3-4]. One of the Deep Learning algorithms that are widely used in prediction is Neural Network [7-8]. In deep learning there are three types of neural networks that form the basis for most models, namely Artificial Neural Networks (ANN) [9], Convolutional Neural Networks (CNN) [10-12], and Recurrent Neural Networks (RNN) [13- 15], and can be used in time series forecasting. The neural network models and architectures that can be used for time series forecasting have different performances depending on how the model is built and what dataset is used for training the model.

Several related studies that have been carried out by previous researchers have tried to predict time series data using several methods from the knowledge branch of machine learning [16]. Khoshalan et al. [17] carried out research on copper price prediction using Gene Expression Programming (GEP), Artificial Neural Network (ANN), Adaptive Neuro-Fuzzy Inference System (ANFIS), and ant colony optimization algorithm (ANFIS-ACO). The result is that the four methods have satisfactory performance in predicting copper prices. Afrianto et al. [18] conducted an experiment on stock price prediction with public sentiment factor using BiLSTM and produced the smallest MSE value of 0.094 and the smallest RMSE value of 0.306. Adhinata and Rakhmadani [19] conducted a study on forecasting the daily increase in COVID-19 cases with the LSTM model, the experimental results showed that the deep learning approach resulted in MSE values of 0.0308, RMSE 0.1758, and MAE 0.13. Ramadan et al. [20] conducted an experiment on prediction sea level height using RNN and LSTM. The results for the 14- day prediction get a correlation coefficient R2 of 0.97 and an RMSE value of 0.036. Jaseena and Kovoor
[21] used the EWT-BiLSTM and BiLSTM models for wind speed prediction which resulted in the EWT-BiLSTM model being superior to the BiLSTM model.

Koshiyama et al. [22] created an architecture called QuanNet that can study market trends and use it to learn market-specific trading strategies that excel using transfer learning methods in making specific strategies for predicting global market trends. Ye and Dai [23] proposed a transfer learning-based hybrid algorithm, namely Online Sequential Extreme Learning Machine with Kernels (OS-ELMK), and ensemble learning, (TrEnOS-ELMK), compared with many existing time series prediction methods, the newly proposed algorithm considers old data and can effectively utilize the latent knowledge implicit in the data for prediction.

3. Proposed Method

This study integrates BiLSTM with optimization algorithms (Adam, RMSProp, and SGD). Fine-Tuning and Transfer Learning are independently employed for enhanced training. MSE and R2 score serve as performance metrics. Figure 1 depicts the research workflow. Publicly available data from id.investing.com (508 rows, 6 columns) during the period from October 1, 2019, to October 1, 2021, is utilized for experimentation.

Preprocessing the Data: The data preprocessing stage is the stage where the data will go through several steps to prepare the data to be ready for use in the next stage, the stages are: Data Cleaning to clean data values such as filling in missing values; Data Reduction choosing the features to be used; Data Transformation such as making dataset X and Y where X has 2 features (Open & Volume prices) while Y has 1 feature (Close price); scaling the data values with a scale of 0 to 1; converting data values into *numpy* arrays; splitting the data for training and for testing data with 80%:20% ratio for Model 1, Model 2 and Model 3, while for Model 4, the data is split into 75%:25% ratio. The use of different data ratio is to verify whether transfer learning can produce good performance compared to fine tune approach even the training data portion is lesser.

1. *Model Creation*: At this stage the BiLSTM model that combined with the three optimizers will be created and will be used as a prediction model.
2. *Creating weight storage models*: The next stage is weight storage model that will be used to store the interconnection weights between neurons or nodes
3. *Training and saving the models*: At this stage, the model will be trained using the split dataset with the number of training adjusting the epoch. Then the model that has been trained and has interconnection weights is stored and will later be used again in prediction the next metal price.
4. *Testing the models:* Here, the models are validated using the split data and then the testing result is displayed graphically.
5. *Prediction*: Then the next stage is running experiments on predictions using the testing data.

6. *Performance metrics calculation*: Performance matrix will be measured to assess the performance of the prediction models.

7. *Comparison*: In the last stage, a comparison of the models' performance will be carried out to determine the best model.

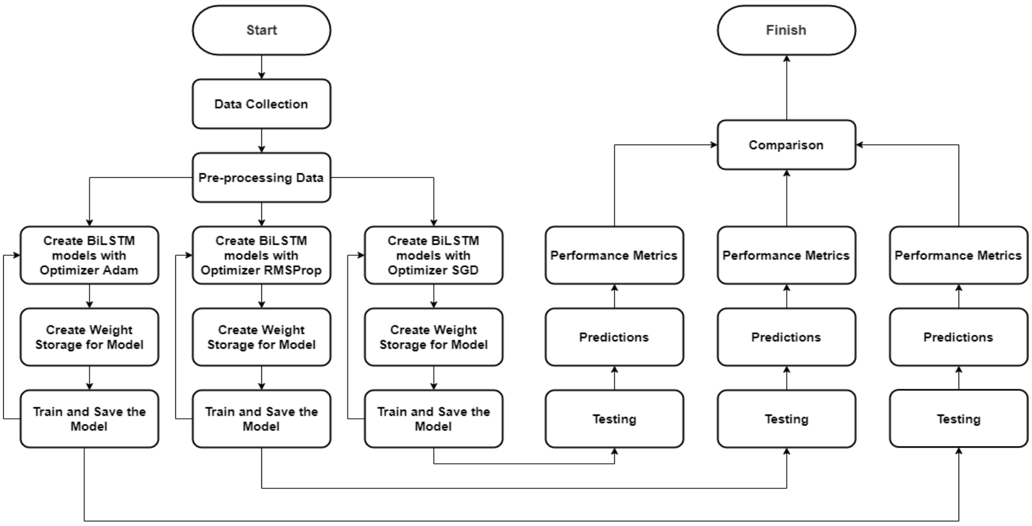


Figure 1: Overall research workflow

3.1 BiLSTM Model

Bidirectional Long Short-Term Memory (BiLSTM) is the process of making each neural network has a sequence of information from both directions backwards (future to past) or forward (past to future). In Bidirectional, inputting flows from both directions makes BiLSTM different from ordinary LSTM and makes the BiLSTM model able to perform better in utilizing the information. Figure 2 shows how the BiLSTM works and Table 1 shows the structure model parameters and their values applied in the experiments.

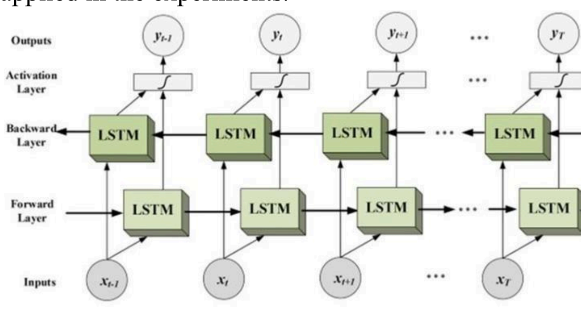


Figure 2: Architecture of BiLSTM [18]

Parameter	Value
Model	Sequential ()
Input	2 nodes
1st Bidirectional LSTM	64 nodes
2nd Bidirectional LSTM	32 nodes
3rd Bidirectional LSTM	16 nodes
Batchnormalization	
Dropout	0.2
Dense	1
Model.Compile	MSE, Optimizer

Table 1: Model structure parameter [18], [19]

3.2 Optimizer

This study considers three optimizer algorithms as follows :

- *Adam Optimizer Algorithm* is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to iteratively update the weights based on the training data.
- *RMSprop Optimizer Algorithm* is similar to the stochastic gradient descent algorithm with momentum. RMSprop optimization limits oscillations in the vertical direction. This can increase the learning speed and the algorithm can take larger steps in the horizontal direction that converges more quickly.

SGD Optimizer Algorithm is an optimization algorithm that is often used in machine learning applications to find the most suitable model parameters between the predicted output and the actual output. Figure 3 illustrates the four models creation with Adam optimizer experiments stages. Models creation using other two optimizers is conducted similarly. As we can see in Figure 3, Model 1 is developed from the scratch. Model 2 is created from the obtained Model 1 through fine-tune approach. Next step is the creation of Model 3, also using fine tune approach. It means that each layer is trained, using its corresponding dataset. Lastly, Model 4 is created through transfer learning approach. This strategy is taken after observing the dynamic and fluctuation of the metals prices data from time to time

3.3 Fine Tuning

Fine-tuning is a way of implementing or utilizing transfer learning to make better adjustments to improve the performance and accuracy of the previously trained network [24]. Specifically, fine- tuning is the process of taking a trained model for one given task and then tuning or modifying the model to make it perform a second similar task.

3.4 Transfer Learning

Transfer learning is the enhancement of learning in a new task through the transfer of knowledge from related tasks that have been learned. Transfer learning is concerned with issues such as multi-tasking learning and concept shifting and is not exclusively a field of study for deep learning [25]. Transfer learning aims to improve target training performance in the target domain by transferring knowledge contained in different sources but related domains. However, transfer learning is popular in deep learning, given the enormous resources required to train deep learning models or the large and challenging datasets on which deep learning models are trained. Two common approaches are as follows:

1. Developed Model Approach:
   (a) Select Source Task → (b) Develop Source
   Model → (c) Reuse Model → (d) Tune Model.

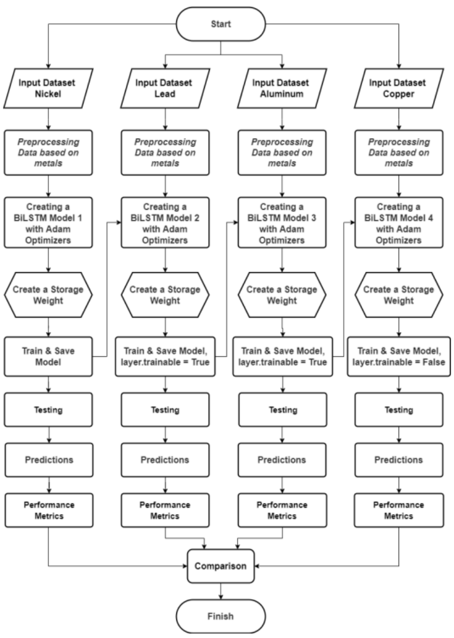


Figure 3: Models creation witAdam optimizer experiments stages

2. Pre-trained Model Approach:
   (a) Select Source Model → (b) Reuse Model → (c) Tune Model

3.5 Performance Metrics

Two metrics are selected to measure the performance of the proposed models, i.e.: MSE and R2 Score. Mean Squared Error (MSE) or Mean Squared Deviation (MSD) is used to measure the difference in the mean squared between the estimated value and the actual value [26]. R2 Score is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, R2 or r-squared indicates how well the data fit the regression model [27].

3.6 Dataset

Four datasets on Nickel, Lead, Aluminum, and Copper are taken from the id.investing.com portal. Each dataset consists of 508 rows and 6 columns. Figure 4 shows the four datasets.

Four models are created, one model for each dataset, i.e.: Model 1 for Nickel price dataset, Model 2 for Lead price dataset, Model 3 for Aluminum price dataset, and Model 4 for Copper price dataset. This sequence is based on the trend of each dataset. In Nickel and Lead price datasets, the price data movement is still fairly stable, however, for Aluminum and Copper the prices fluctuation is sharp enough as shown in Figure 5. Then each model is combined with three different optimizers, thus 12 models will be created.

	A	B	C	D	E	F
1	date	close	open	high	low	volume
2	2019-10-1	17.220	17.250	17.325	17.195	239750
3	2019-10-2	17.485	17.525	17.505	17.485	187440
4	2019-10-3	17.625	17.630	17.630	17.610	117020
507	2021-9-29	18.342	18.600	18.555	18.555	21270
508	2021-9-30	17.936	18.170	18.100	18.100	19540
509	2021-10-1	17.971	18.100	18.080	18.100	11970

Nickel (Model 1)

	A	B	C	D	E	F
1	date	close	open	high	low	volume
2	2019-10-1	2.096	2.130	2.122	2.096	98220
3	2019-10-2	2.100	2.095	2.110	2.090	66660
4	2019-10-3	2.129	2.108	2.130	2.106	46370
507	2021-9-29	2.140	2.158	2.160	2.158	7540
508	2021-9-30	2.093	2.087	2.087	2.078	16970
509	2021-10-1	2.141	2.127	2.128	2.126	9940

Lead (Model 2)

	A	B	C	D	E	F
1	date	close	open	high	low	volume
2	2019-10-1	5.686	5.640	5.686	5.601	257720
3	2019-10-2	5.678	5.662	5.702	5.662	202320
4	2019-10-3	5.662	5.657	5.665	5.655	191830
507	2021-9-29	9.155	9.220	9.215	9.215	29880
508	2021-9-30	8.937	9.030	9.025	9.021	38370
509	2021-10-1	9.128	9.100	9.099	9.100	33670

Aluminum (Model 3)

	A	B	C	D	E	F
1	date	close	open	high	low	volume
2	2019-10-1	1.740	1.719	1.740	1.719	473950
3	2019-10-2	1.705	1.723	1.722	1.706	484720
4	2019-10-3	1.718	1.712	1.718	1.710	329770
507	2021-9-29	2.912	2.937	2.935	2.930	36670
508	2021-9-30	2.859	2.870	2.866	2.867	60390
509	2021-10-1	2.857	2.881	2.880	2.881	38560

Copper (Model 4)

Figure 4. Illustration of Datasets for the experiments

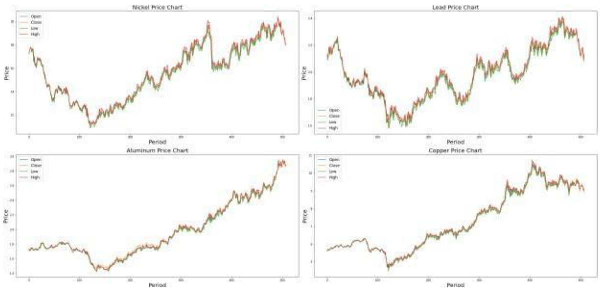


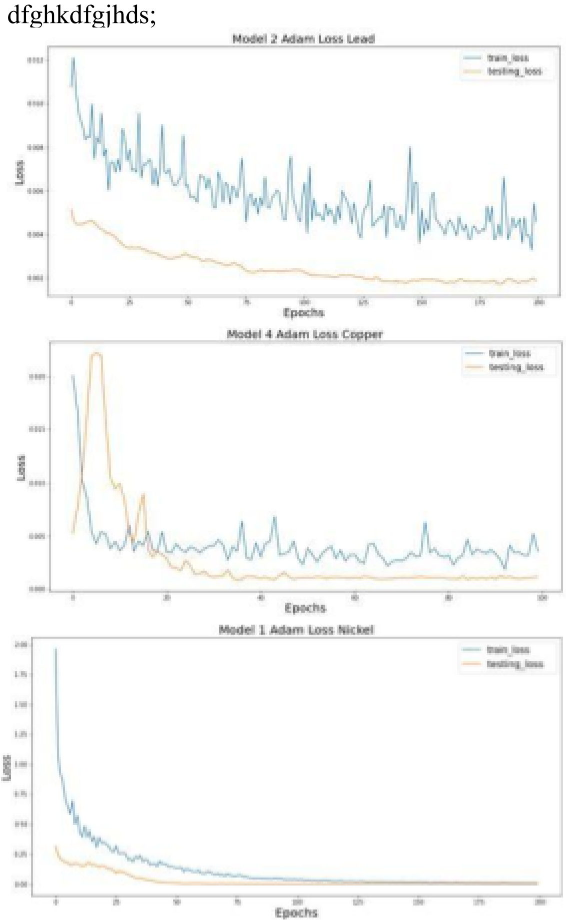
Figure 5. Trends of price movement of the four metals

4. Experimental Result and Discusion

For the implementation purpose, this study uses Intel Core i7-8750H CPU @ 2.20GHz (12 CPUs), RAM 8 GB, running Windows 10 and Python version 3.7.13 programming language, with Google Collaboratory supporting apps and libraries, including: Sys, Pandas, MatplotLib, Seaborn, Tabulate, MinMaxScaler, train_test_split, mean_squared_error, r2_score. Whereas for model creation, the following libraries are utilized: Sequential, LSTM, Dense, Dropout, Bidirectional, Batchnormalization, Input, ModelCheckpoint, CSVLogger, and Load_model.

4.1 BiLSTM + Adam Optimizer

Four models for this arrangement are: Model 1 Adam, Model 2 Adam, Model 3 Adam, and Model 4 Adam. Figure 6 and Figure 7 show the loss and prediction of the models, respectively. It can be seen dfghkdfgjhds;



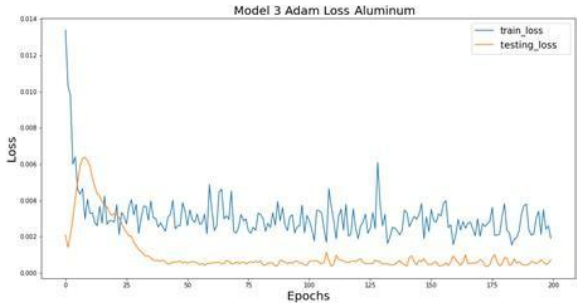


Figure 6. Loss values of BiLSTM+Adam optimizer

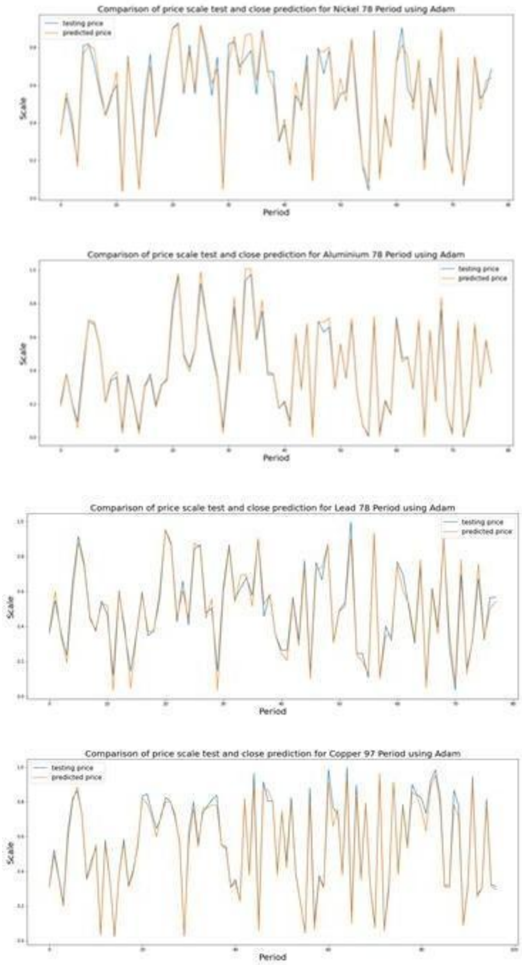


Figure 7. Accuracy of BiLSTM+Adam optimizer

4.2 BiLSTM + RMSProp Optimizer

Four models for this arrangement are: Model 1 RMSProp, Model 2 RMSProp, Model 3 RMSProp, and Model 4 RMSProp. Figure 8 and Figure 9 show the loss and prediction of the models, respectively. Figure 8 shows the convergence of the four models. Model 2 RMSProp fluctuates a lot compared to the other three models. Model 4 RMSProp achieves the best performance.

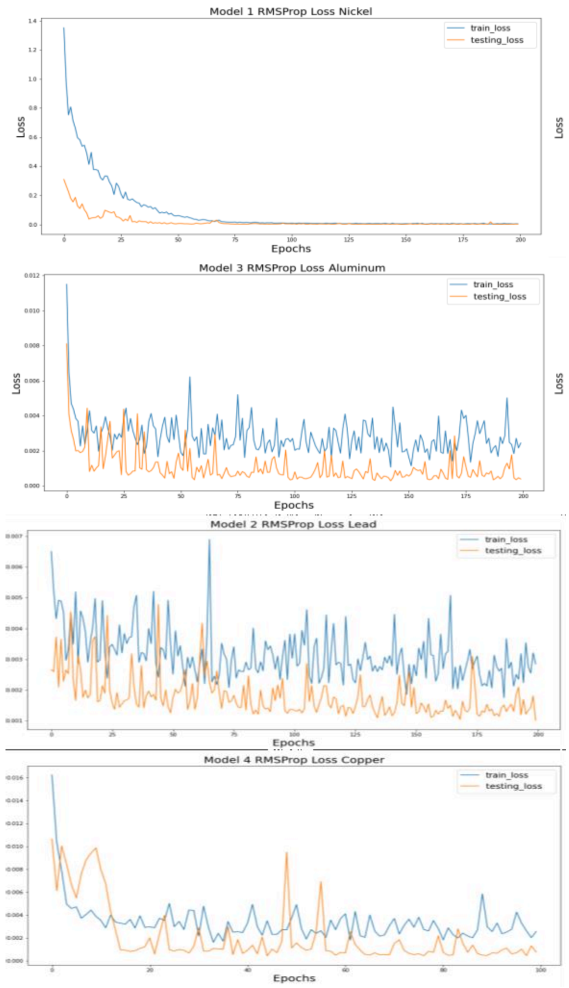
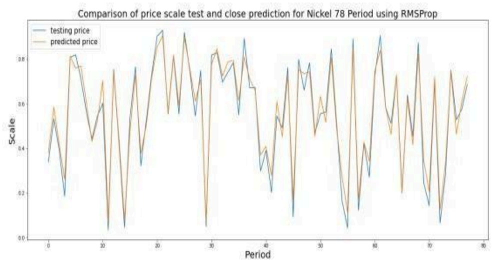
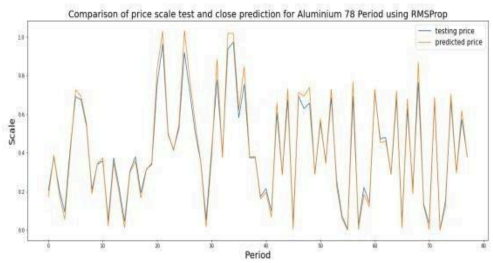
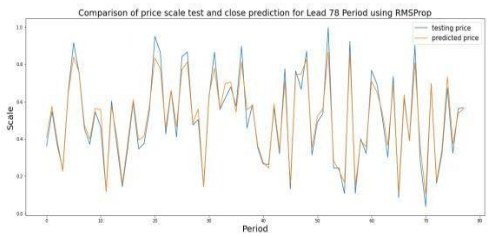


Figure 8. Loss values of BiLSTM+RMSProp optimizer







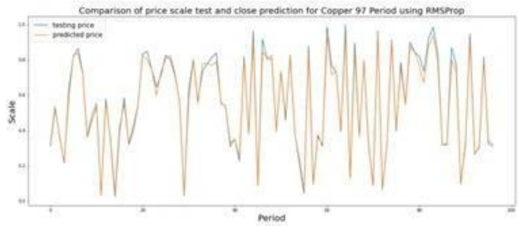


Figure 9. Accuracy of BiLSTM+RMSProp optimizer

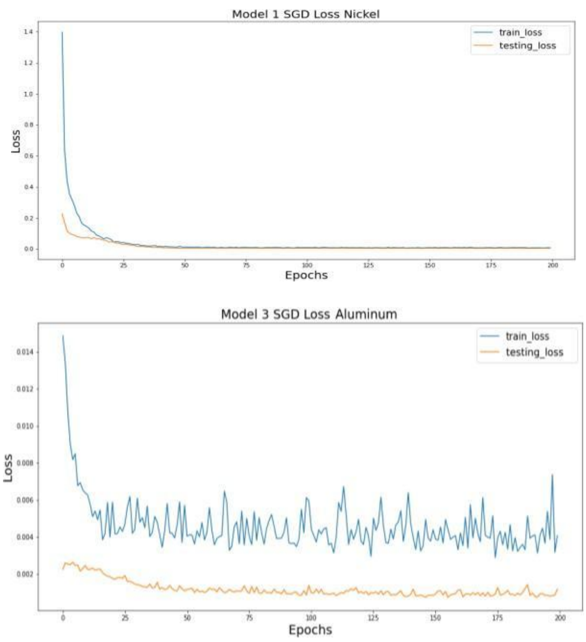
4.3 BiLSTM + SGD Optimizer

Four models for this arrangement are: Model 1 SGD, Model 2 SGD, Model 3 SGD, and Model 4 SGD. Figure 10 and Figure 11 show the loss and prediction of the models, respectively. Figure 10 shows the convergence of the four models. Model 2 SGD is similar to Model 2 RMSProp fluctuates a lot compared to the other three models. Model 3 RMSProp achieves the best performance.

Table 2 summarizes overall models' performance. It can be seen that in general, the combination of BiLSTM with the three optimizers improves the prediction performance in terms of R2, MSE. Implementation on other combinations only provides better number of epoch to achieve convergence, and shorter processing time.

Model	R2 Score	MSE	Epoch	Time (Sec)
Model 1 Adam	0.96234	0.00236	200	619
Model 2 Adam	0.96823	0.00184	200	411
Model 3 Adam	0.98877	0.00074	*200*	*412*
Model 4 Adam	0.98522	0.00115	100	211
Model 1 RMSProp	0.96823	0.00184	200	411
Model 2 RMSProp	0.96823	0.00184	200	411
Model 3 RMSProp	0.97488	0.00166	200	412
Model 4 RMSProp	**0.99029**	**0.00076**	**100**	**213**
Model 1 SGD	0.93801	0.00388	200	613
Model 2 SGD	0.94805	0.00301	200	396
Model 3 SGD	0.98234	0.00117	*200*	*413*
Model 4 SGD	0.97771	0.00174	100	213

Table. Overall Models' performance



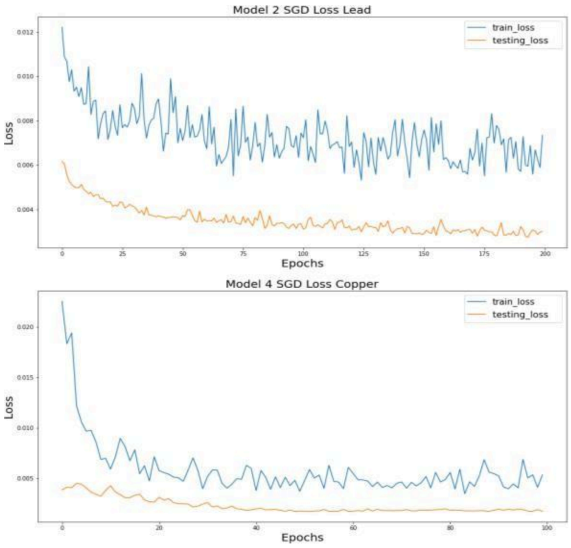


Figure 10. Loss values of BiLSTM+SGD optimizer

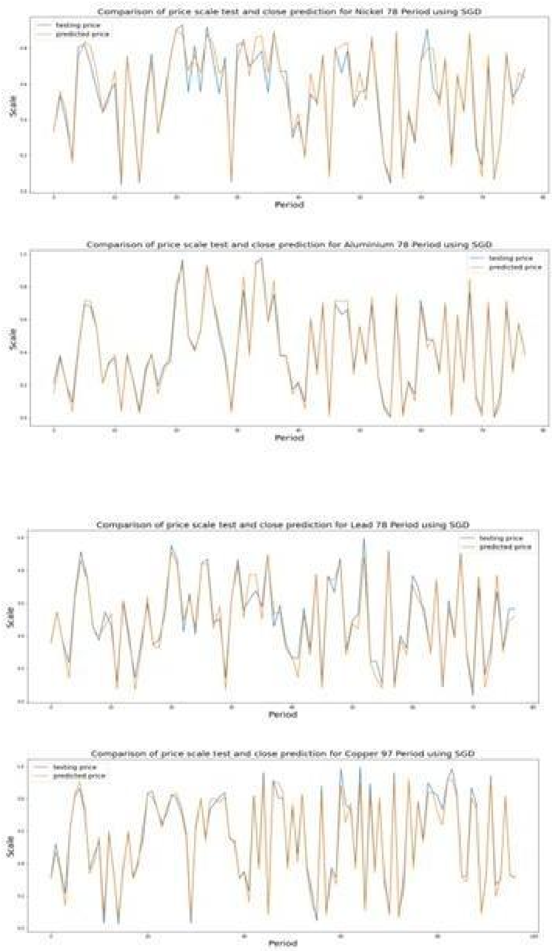


Figure 11. Accuracy of BiLSTM+SGD optimizer

5. Discusion and Comparisson

As shown in Table 2, all models are able to predict the four metal's price very well, particularly Model 3 and Model 4 as they are created in sequence number 3 and number 4. From observation of Model 4's performances, it can be seen that Model 4 RMSProp (model using Copper dataset combined with RMSProp optimizer) achieves the best performance among other 11 models. On the other hand, for other combinations, Model 3s (Model 3 Adam and Model 3 SGD) perform better compared to their Model 4s. Nevertheless, Model 3s and Model 4s need less processing time compared to Model 1s and Model 2s. As discussed in Section 3.2 on model creation, neuron weights of Hidden layer of Model 1 resulted from its training stage is used as basis for Model 2 creation using fine tune approach. This way makes neurons at Hidden Layer of

Model 2 has their initial weights, then when it is trained again using data for Model 2 will produce better performance (in terms of R2 Score, MSE and processing time) compared to Model 1. Same procedure happens to Model 3 creation. Whilst, Model 4 is purely created using transfer learning approach, without retraining but directly uses the neurons' weights from Model 3. Table 3 shows the top three models resulted from the experiments. We observe that the best performance for BiLSTM+RMSProp models, BiLSTM+Adam models, and BiLSTM+SGD models achieved by Model 4 RMSProp, Model 3 Adam, and Model 4 Adam.

Table 2. The 3 top models

Model	R2 Score	MSE
Model 4 RMSProp	0.96234	0.00236
Model 3 Adam	0.96823	0.00184
Model 4 Adam	0.98877	0.00074

Table 3. The 3 top models

Unfortunately, not so many works on metal prices prediction in commodity futures market are available in the literatures. Nevertheless, Table 4 shows performance comparison with some existing models

Commodity	Method	Ref.#	R2 Score	MSE
Nickel	BiLSTM+RMSProp	Shao et al. [28]	0.96823	0.00184
	LSTM21+PSO		N/A	0.03
Lead	BiLSTM+RMSProp	Lu et al. [29]	0.96823	0.00184
	ARIMA		0.9703	N/A
Aluminum	BiLSTM+Adam	(Mysen & Thornton [30]	0.98877	0.000741
	XGBosst		N/A	64.62
Copper	BiLSTM+RMSProp	Khoshalan et al. [17]	0.99029	0.000756
	ANN		0.98100	N/A

Table 4. Performance comparison

6. Conclusion

This study has proposed a prediction engine as a combination of BiLSTM with three optimization algorithms, i.e.: Adam, RMSProp, and SGD, and transfer learning to make model training better. Through a series of experiments, this study has shown that the implementation of transfer learning approach on the combination of BiLSTM and RMSProp optimizer improves all aspects of the prediction performance, i.e.: R2, MSE, number of epoch as well as processing time. Implementation on other combinations only provides better number of epoch to achieve convergence, and shorter processing time. The best performance with 0.990287 of R2 Score and MSE value of 0.000756 was achieved for model that is created using Copper dataset and implementation of transfer learning. The transfer learning approach reduces the prediction processing time and at the same time still provides excellent prediction results while the number of data training and testing is less. The proposed model is expected will assist metals trading players to secure their business, which in turn will support the advancement of renewable energy technology. Combinations of other prediction engines with several optimizer algorithms are considered as future study. Besides, as this study does not perform model validation, thus, validation of the model is considered also as one of future work.

REFERENCES

. [1] G. Schuh, G. Reinhart, J. Prote, F. Sauermann, J. Horsthofer, F. Oppolzer, F., & D. Knoll. "Data mining definitions and applications for the management of production complexity," *Procedia CIRP*, vol. 81, pp. 874–879. https://doi.org/10.1016/j.procir.2019.03.217.

[2] Dedi, & A.Cherid, "Data mining data processing prospective of Indonesian migrant workers (PMI) with application of K-Means clustering and K- nearest neighbor (KNN) classification: case study of PT. SAM," *B.Sc Thesis, Mercu Buana University, Indonesia*. https://repository.mercubuana.ac.id/61695/

[3] R. Law, G. Li, D.K.C Fong & X. Han, "Tourism demand forecasting: A deep learning approach," *Annals of Tourism Research.*, 75, 410–423. https://doi.org/10.1016/j.annals.2019.01.014.

[4] J.M. Stokes, K. Yang, S. Kyle, W. Jin, A. Cubillos-Ruiz, N.M. Donghia, C.R. MacNair, S. French, L.A Carfrae, Z. Bloom-Ackermann, V.M. Tran, A. Chiappino-Pepe, A.H. Badran, I.W. Andrews, E.J. Chory, G.M. Church, E.D. Brown, T.S. Jaakkola, R. Barzilay & J. C. Collins, "A deep learning approach to antibiotic discovery," *Cell*, vol. 180, pp. 688 – 702, 2020. https://doi.org/10.1016/j.cell.2020.01.021

[5] C.Xu, J. Ji & P. Liu, "The station-free sharing bike demand forecasting with a deep learning approach and large-scale datasets," *Transportation Research Part C: Emerging Technologies*, vol. 95, ppl 47–60. https://doi.org/10.1016/j.trc.2018.07.013.

[6] O.I Abiodun, A. Jantan, A.E. Omolara, K.V. Dada, N.A.E. Mohamed & H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, e00938. https://doi.org/10.1016/j.heliyon.2018.e00938.

[7] D. Rosadi, D. Arisanty & D. Agustina, "Prediction of forest fire using neural networks with backpropagation learning and exreme learning machine approach using meteorological and weather index variables," *Media Statistika.*, vol. 14, ppl 118–124, 2022. https://doi.org/10.14710/medstat.14.2.118-124

[8] I.T. Rahayu, N. Nurhasanah & R. Adriat, "Prediction of dengue hemorrhagic fever cases based on weather parameters using back propagation neural networks (case study in Pontianak city)," *Jurnal Pendidikan Fisika Indonesia*, vol. 15, pp. 114–121, 2019. https://doi.org/10.15294/jpfi.v15i2.19633.

[9] P. Jiang, Z. Liu, X. Niu& L. Zhang, "A combined forecasting system based on statistical method, artificial neural networks, and deep learning methods for short-term wind speed forecasting," *Energy*, vol. 217, 119361. https://doi.org/10.1016/j.energy.2020.119361.

[10] I. Kurniawan, L.S. Silaban& D. Munandar, "Implementation of convolutional neural network and multilayer perceptron in predicting air temperature in Padang," *RESTI*, vol. 4, pp. 2–7, 2020. https://doi.org/10.29207/resti.v4i6.2456.

[11] R.F. Rahmat, Dennis, O.S. Sitompul, S. Purnamawati & R. Budiarto, "Advertisement billboard detection and geotagging system with inductive transfer learning in deep convolutional neural network" *Telkomnika*, vol. 17, pp. 2659–2666, 2019. http://doi.org/10.12928/telkomnika.v17i5.11276.

[12] F. Makinoshima, Y. Oishi, T. Yamazaki, T. Furumura& F. Imamura. "Early forecasting of tsunami inundation from tsunami and geodetic observation data with convolutional neural networks," *Nature Communications*, vol. 12, pp. 1–10, 2021. https://doi.org/10.1038/s41467-021-22348-0.

[13] M.A. Faishol, E. Endroyono, & A. N. Irfansyah, "Predict urban air pollution in surabaya using recurrent neural network – long short term memory," *JUTI*, vol. 18, no. 102, 2020. http://dx.doi.org/10.12962/j24068535.v18i2.a 988.

[14] F. Granata & F. D. Nunno, "Forecasting evapotranspiration in different climates using ensembles of recurrent neural networks,"

Agriculture and Water Management, vol. 255,107040, 2021. https://doi.org/10.1016/j.agwat.2021.107040.
[15] K.E. ArunKumar, D.V. Kalaga, C.M.S. Kumar, M. Kawaji, M & T. M. Brenza, "Forecasting of COVID-19 using deep layer Recurrent Neural Networks (RNNs) with Gated Recurrent Units (GRUs) and long short-term memory (LSTM) cells," Chaos, Solitons & Fractals, vol. 146, 110861, 2021. https://doi.org/10.1016/j.chaos.2021.110861.
[16] T. Ahmad, & H. Chen, "A review on machine learning forecasting growth trends and their real-time applications in different energy systems," Sustainable Cities and Society, vol. 54, 102010, 2020. https://doi.org/10.1016/j.scs.2019.102010.
[17] H.A. Khoshalan, J. Shakeri, I. Najmoddini & M. Asadizadeh, "Forecasting copper price by application of robust artificial intelligence techniques," Resources Policy, vol. 73, 102239, 2021. https://doi.org/10.1016/j.resourpol.2021.102239.
[18] N. Afrianto, D.H. Fudholi & S. Rani. « Stock price prediction using BiLSTM with public sentiment factor," RESTI, vol. 6, ppl 41–46, 2022. doi: 10.29207/resti.v6i1.3676.
[19] F.D. Adhinata & D. P. Rakhmadani, "Prediction of Covid-19 daily case in indonesia using long short term memory method. Teknika, vol. 10, ppl 62–67, 2021. https://doi.org/10.34148/teknika.v10i1.328.
[20] A.W. Ramadhan, D. Adytia, D. Saepudin, S. Husrin & A. Adiwijaya, "Forecasting of sea level time series using RNN and LSTM case study in Sunda strait," Lontar Komputer, vol. 12, no. 130, 2021. https://doi.org/10.24843/LKJITI.2022.v13.i02
[21] K.U. Jaseena & B. C. Kovoor, "Decomposition-based hybrid wind speed forecasting model using deep bidirectional LSTM networks," Energy Conversion and Management, vol. 234, 113944, 2021. https://doi.org/10.1016/j.enconman.2021.113944.
[22] A. Koshiyama, S. Flennerhag, S.B. Blumberg, N. Firoozye& P. Treleaven, "QuantNet: transferring learning across systematic trading strategies, 2020. http://arxiv.org/abs/2004.03445.
[23] R. Ye & Q. Dai, "A novel transfer learning framework for time series forecasting," Knowledge-Based System, vol. 156, pp. 74–99, 2018. https://doi.org/10.1016/j.knosys.2018.05.021.
[24] A. Mustafid, M.M. Pamuji, & S. Helmiyah,. "A comparative study of transfer learning and fine-tuning method on deep learning models for wayang dataset classification," IJID, vol. 9, pp. 100–110, 2020. https://doi.org/10.14421/ijid.2020.09207.
[25] Y. Gao, Y. Ruan, C. Fang & S. Yin. "Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data," Energy and Buildings, vol. 223, 110156, 2020. https://doi.org/10.1016/j.enbuild.2020.110156.
[26] M.S. Acharya, A. Armaan& A. S. Antony, "A comparison of regression models for prediction of graduate admissions. ICCIDS 2019, pp. 1–5, 2019. https://doi.org/10.1109/ICCIDS.2019.8862140
[27] J. Kumar, R. Goomer & A. K. Singh. "Long short term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters," Procedia Computer Science, vol. 125, pp. 676–682, 2018. https://doi.org/10.1016/j.procs.2017.12.087.
[28] B. Shao, M. Li, Y. Zhao & G. Bian, "Nickel price forecast based on the lstm neural network optimized by the improved pso algorithm," Mathematical Problems in Engineering, 2019, Article ID 1934796. https://doi.org/10.1155/2019/1934796.
[29] L.L. Lu, X. Ma, Y.X. Wang, & G. B. Yu, « Lead price forecasting based on ARIMA model," Advanced Materials Research, vol. 488-489, pp. 1582-1586, 2012.
[30] S.J. Mysen & E. M. Thornton, "Forecasting the price of aluminum using machine learning empirical comparison of machine learning and statistical methods," 2021, Master thesis, Norwegian School of Economics. https://openaccess.nhh.no/nhh-xmlui/bitstream/handle/11250/2985394/masterthesis.pdf?sequence=