Metal Price Spike Prediction via a Neurosymbolic Ensemble Approach

Nathaniel Lee [a,*], Noel Ngu [b], Harshdeep Singh Sahdev [b], Pramod Motaganahall [b], Al Mehdi Saadat Chowdhury [b], Bowen Xi [b] and Paulo Shakarian [b]

[a] *Arizona State University, Tempe, AZ, USA*
E-mail: nlee51@asu.edu
[b] *Arizona State University, Tempe, AZ, USA*
E-mails: nngu2@asu.edu, hsahdev@asu.edu, pmotagan@asu.edu, achowd43@asu.edu, bowenxi@asu.edu, pshak02@asu.edu

Abstract. Predicting price spikes in critical metals such as Cobalt, Copper, Magnesium, and Nickel is crucial for mitigating economic risks associated with global trends like the energy transition and reshoring of manufacturing. While traditional models have focused on regression-based approaches, our work introduces a neurosymbolic ensemble framework that integrates multiple neural models with symbolic error detection and correction rules. This framework is designed to enhance predictive accuracy by correcting individual model errors and offering interpretability through rule-based explanations. We show that our method provides up to 6.42 % improvement in precision, 29.41% increase in recall at 13.24% increase in F1 over the best performing neural models. Further, our method, as it is based on logical rules, has the benefit of affording an explanation as to which combination of neural models directly contribute to a given prediction.

Keywords: Neurosymbolic, unsupervised learning, time series analysis

1. Introduction

Global trends such as the energy transition from fossil fuels and the reshoring of manufacturing has led to an increased reliance on metals. As a result, significant metal price fluctuation can lead to outsized economic impacts. In this paper, we study the problem of predicting spikes in the prices of four critical metals: Cobalt, Copper, Magnesium, and Nickel. While prior work [2] [6] [4] have mainly focused on regression and prediction by a single model, we focus on the identification of price spikes, which are potentially more economically destabilizing. Further, we propose a neurosymbolic ensemble approach to enable the use of multiple models. The way in which models are ensembeled is guided by an extension of the error detection and correction rules (EDCR), a symbolic rule learning technique [5, 8] (Section 3). We find that our approach provides up to 6.42% improvement in precision, 29.41% increase in recall and 13.24% increase in F1 over the best performing neural models (Table 2) in addition to an explanation as to which models contribute to a prediction that we explore with an ablation study (Section 4.3).

[*]Corresponding author. E-mail: nlee51@asu.edu.

2. Related Work

There have been several existing works on metal price prediction that rely on single model approaches, such as Support Vector Regression (SVR) [1] and Artificial Neural Networks (ANNs) [12] to forecast general price trends by modeling non-linear relationships in metal price. While these models offer valuable insights, they may be limited in their ability to fully capture the complex patterns and variations in price movements. A multi-model approach can be more effective by combining the strengths of various models to more broadly capture price dynamics.

Several studies have explored hybrid neural models that combine different architectures to enhance predictive power. For example, LSTM-CNN and LSTM-GRU models have been used to capture both short-term fluctuations and long-term trends in price data [7]. Another study combined LSTM and ANN with the GARCH model to predict copper price volatility, demonstrating the effectiveness of integrating statistical and deep learning methods [3]. Additionally, research has proposed hybrid models like GA-ELM and PSO-ELM, which optimize the parameters of Extreme Learning Machines (ELM) using meta-heuristic algorithms such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), achieving more accurate copper price forecasts [9]. These hybrid models have shown to outperform standalone models by leveraging complementary strengths across architectures.

While hybrid models integrate architectures within a single model, our ensemble approach offers further flexibility by combining multiple independent models, allowing us to capture a wider range of patterns and achieve more robust predictions. Additionally, recent studies have shown that self-attention mechanisms improve the ability to capture long-range dependencies in time-series data, leading to better performance in forecasting key trends over extended periods [10]. These mechanisms allow models to focus on important temporal features, making them particularly effective in time-series forecasting. Drawing from these insights, we have incorporated self-attention mechanisms into our neurosymbolic ensemble to enhance the detection of significant patterns and fluctuations, particularly in identifying metal price spikes.

3. EDCR For Metal Price Prediction

In this paper, we define the Metal Price Spike Classification (MPSC) problem as the identification of significant shifts in price movements in the time series of metal commodity prices. Given a window of n days, we wish to predict if the price at the $(n+1)$th day is "anomalous" which can be defined as either an unsupervised or semi-supervised approach. For example, a spike can be a certain number of standard deviations above the moving average - hence this can be framed as a binary classification problem with two classes. In our experiments, spikes are characterized as values that exceed or fall below the rolling mean by two rolling standard deviations, with a window size of 20 days. We note that other formulations to define spike are possible and that our approach does not depend on the specific definition of spikes.

Next, we describe the error detection and correction rule (EDCR) framework of [8] to address the MPSC problem. Intuitively, given a trained model, f_θ, EDCR involves the learning of rules to detect and correct errors based on conditions under which the model is used. These conditions can be aspects of the samples themselves or the environment. A key intuition of this work is that *we have one primary model and use other neural models as conditions* - as such EDCR provides customized rules that specify the best way to combine these models. In this work, all the models are predicting the same binary class, as opposed to prior work where the primary model was always multi-class and the models for conditions were binary [8] or where classes at different levels of granularity of the same model are used for conditions [5].

3.1. EDCR Rule Learning Algorithms (recap of [8])

To effectively apply the EDCR framework for MPSC, we first need to understand the key rule-learning algorithms from [8]. These algorithms form the basis for detecting and correcting errors in the primary model's predictions. The EDCR framework consists of two rule types: error detection and error correction. Each class in the dataset has one detection rule and one correction rule, designed to identify when the model's prediction is incorrect and how to correct it.

Error Detection Rule Learning. The goal of error detection is to identify when a prediction made by the primary model f_θ is incorrect. Algorithm 1, DetRuleLearn [8], learns these rules by selecting conditions (denoted as DC_i, the detection conditions for class i) that maximize precision while ensuring the recall reduction remains within a given threshold. The learned rule flags misclassifications by the primary model f_θ, which predicts a class $pred_i(\omega)$, where ω is a sample.

$$error(\omega) \leftarrow pred_i(\omega) \wedge \bigvee_{j \in DC_i} cond_j(\omega)$$

Algorithm 1 DetRuleLearn [8]

```
    Require: Class $i$, Recall reduction threshold $\epsilon$, Condition set $C$
    Ensure: Subset of conditions $DC_i$
1: $DC_i := \emptyset$
2: $DC^* := \{c \in C \ s.t. \ NEG_{\{c\}} \leqslant \epsilon \cdot \frac{N_i P_i}{R_i}\}$
3: while $DC^* \neq \emptyset$ do
4:     $c_{best} = \arg\max_{c \in DC^*} POS_{DC_i \cup \{c\}}$
5:     Add $c_{best}$ to $DC_i$
6:     $DC^* := \{c \in C \setminus DC_i \ s.t. \ NEG_{DC_i \cup \{c\}} \leqslant \epsilon \cdot \frac{N_i P_i}{R_i}\}$
7: end while
8: return $DC_i$
```

Here, POS_{DC} is the number of samples where the conditions are met and the model's prediction is an error, indicating how well the detection rule identifies misclassifications. NEG_{DC} counts the samples where the conditions are met, but the prediction was correct. BOD represents all samples that satisfy the rule's conditions, regardless of whether the prediction was correct.

Error Correction Rule Learning. Once a misclassification is detected, Algorithm 2, CorrRuleLearn [8] algorithm reassigns the sample to the correct class. It does this by selecting condition-class pairs (denoted as CC_i, the prior assigned class and associated condition that allows us to reclassify the sample to i) to maximize both precision and recall. The correction rules are based on the output of other models or conditions, represented as $cond_q(\omega)$, which help correct the primary model's prediction.

$$corr_i(\omega) \leftarrow \bigvee_{q,r \in CC_i} (cond_q(\omega) \wedge pred_r(\omega))$$

3.2. EDCR Applied to MPSC

As with [5, 8] model f_θ, the conditions, and the EDCR rules are created based on the training data. We instantiate the logical language for our use case as follows: $assign_{spike}, assign_{no}$ meaning that f_θ assigned a given sample class *spike* (if a price spike is predicted) or *no* (for no spike predicted), $corr_{spike}$ meaning that the model's prediction should be corrected to class *spike* (note due to the binary nature of the problem, we do not use the corresponding $corr_{no}$ predicate in this work), and *cond* meaning that condition *cond* is true for the given sample. In this work, for example, $cond_{CNN1}$ would mean that a model "CNN (1)" (i.e., CNN variant 1 as per Table 4) – which is not f_θ – assigned a sample as having a price spike. We show two example rules obtained from our modified EDCR rule learner in Table 1. Notice that here our conditions are based directly on the classification decisions of other models.

Algorithm 2 CorrRuleLearn [8]

```
Require: Class $i$, Set of condition-class pairs $CC_{all}$
Ensure: Subset of condition-class pairs $CC_i$
1: $CC_i := \emptyset$
2: $CC'_i := CC_{all}$
3: Sort each $(c, j) \in CC_{all}$ from greatest to least by $\frac{POS_{\{(c,j)\}}}{BOD_{\{(c,j)\}}}$ and remove $\frac{POS_{\{(c,j)\}}}{BOD_{\{(c,j)\}}} \leqslant P_i$
4: for $(c, j) \in CC_{all}$ selected in order of the sorted list do
5:     $a := \frac{POS_{CC_i \cup \{(c,j)\}}}{BOD_{CC_i \cup \{(c,j)\}}} - \frac{POS_{CC_i}}{BOD_{CC_i}}$
6:     $b := \frac{POS_{CC'_i \setminus \{(c,j)\}}}{BOD_{CC'_i \setminus \{(c,j)\}}} - \frac{POS_{CC'_i}}{BOD_{CC'_i}}$
7:     if $a \geqslant b$ then
8:         $CC_i := CC_i \cup \{(c, j)\}$
9:     else
10:        $CC'_i := CC'_i \setminus \{(c, j)\}$
11:    end if
12: end for
13: if $\frac{POS_{CC_i}}{BOD_{CC_i}} \leqslant P_i$ then
14:     $CC_i := \emptyset$
15: end if
16: return $CC_i$
```

Table 1
Example EDCR Rule Learned for the MPSC Problem

Rule	Machine Learning Models
$corr_{spike}(w) \leftarrow assign_{no}(w) \wedge cond_{CNN1}(w)$	If base model f_θ identifies sample w as not having a spike (class *no*) and auxiliary model CNN1 classifies a certain day in the time series as a *spike*, then the overall ensemble prediction is *spike*.
$corr_{spike}(w) \leftarrow assign_{no}(w) \wedge cond_{RNN4}(w)$	If base model f_θ identifies sample w as not having a spike (class *no*) and auxiliary model RNN4 classifies a certain day in the time series as a *spike*, then the overall ensemble prediction is *spike*.

3.3. Modified EDCR Rule Learner for MPSC

In this paper, we extend the Algorithm 3, DetCorrRuleLearn of [8] with an overall approach suited for the MPSC problem called MPSCRuleLearn, which calls both DetRuleLearn and CorrRuleLearn as subroutines. In this adaptation, both the detection and correction rules lead directly to corrective actions due to the binary nature of the classification problem. This eliminates the need to separately define error detection and correction rules, simplifying the learning process.

Furthermore, Line 1 of Algorithm 3 is a modified condition set C where conditions are filtered out using two distinct approaches:

– **Top F1**: Condition selection based on the top n Training F1 scores, providing a direct measure of condition efficacy based solely on performance.
– **Threshold**: Includes conditions with Training F1 scores above a given threshold n, allowing for dynamic inclusion based on condition quality.

```
Algorithm 3 MPSCRuleLearn
Require: Recall reduction threshold $\epsilon$, Condition set $C$
Ensure: Subset of conditions $DC_i$
1: $C_f$ := Filtered Condition set C.
2: $CC_{all} := \{\forall_c \in C_f : (nospike, c)\}$
3: $DC = \text{DetRuleLearn}(nospike, \epsilon, C_f)$
4: $CC = \text{CorrRuleLearn}(spike, CC_{all})$
5: return{
       $\text{corr}_{spike}(w) \leftarrow pred_{no}(w) \wedge \bigvee_{cond \in DC} cond(w)$,
       $\text{corr}_{spike}(w) \leftarrow \bigvee_{cond \in CC}(pred_{no}(w) \wedge cond(w))$
   }
```

4. Experiments

In this section, we describe a set of experiments where we evaluated our approach on price spike prediction for four metals (Cobalt, Copper, Magnesium, and Nickel). We also describe our experimental setup as well as an ablation study.

4.1. Experimental Settings

Dataset and Setup In our experiments, we utilized datasets of daily open prices for four distinct metals—cobalt, copper, magnesium, and nickel—from [11], spanning the years 2020 to 2023. This range of metals was selected to assess the applicability of our results across diverse commodity markets. Each row in the dataset contains the open, high, and low prices of a metal on a specific day. The datasets contain 1039 samples of cobalt, 1008 samples of copper, 716 samples of magnesium, and 1006 samples of nickel. Datasets were split into training and test sets at a 60:40 ratio across all datasets. The datasets were first sorted by their date. Then, the first 60% of the dataset was placed into the training set while the last 40% was put into the test set.

Machine Learning Models In our experiments, we trained and fine-tuned a range of machine-learning models. Specifically, we employed Convolutional Neural Networks (CNN), CNN with Attention mechanism (ATTN), Long Short-Term Memory (LSTM) networks, and Recurrent Neural Networks (RNN). In total, there were 4 variants of RNNs and LSTMs, and 12 variants of CNN and ATTN. These variants are elaborated on in Table 4.

Hardware and Implementation Experiments were performed on a AMD Ryzen 9 7420HS CPU, Radeon 780M GPU using Python 3.11 with Tensorflow.

4.2. Model Evaluation

We evaluated how effectively the EDCR framework improved the performance of our foundational classification models: CNN, RNN, LSTM, and ATTN. Each model was trained using features extracted from the daily price data of various metals. After generating a pool of classification results with various model configurations and hyperparameters, these results were processed using the EDCR framework, using our modified Algorithm 3. The performance of the base models with the best precision, recall, and F1 on each metal dataset, as well as their incorporation of the Detection Correction rule using Top F1 selection, are shown in Table 2. For the Top F1 selection, we selected the top 200 conditions (there were 744 in total) based on F1 score to apply the EDCR corrections. Random baseline precisions when recall is set to 1 (e.g., predict spikes for all input) for metals Cobalt, Copper, Magnesium, and Nickel are 0.15, 0.14, 0.19, 0.14 respectively. The base models shown in the table for each metal represent the best precision, best recall, best F1 among all models trained. In general, a baseline model augmented with EDCR improved in all metrics over the baseline model - with the exception of high-precision baseline models. Compared to all trained models, recall and F1 were both significantly improved with the use of EDCR (of note F1 improved by over 10% in three of four metals). We note that our approach is designed to increase the recall of the *spike* class (while minimizing recall decrease to the non-spike class) - which is recall improvements seem to drive the results in most cases (and also why in only one of four metals precision improved significantly over the baseline). However, it is noteworthy that in all but one case, the top recall or top F1 model also improved or maintained precision over its non-EDCR counterpart. In addition to Top F1 selection, we also applied Threshold filtering, as shown in Table 3. For this method, we applied EDCR to rules where the F1 score was greater than 0.5. This approach generally showed similar improvements in recall and F1, with a few exceptions. Notably, for Copper and Nickel, the best precision models showed more significant increases in recall and F1 after applying EDCR with Threshold filtering.

Table 2

Model Evaluation Results for Cobalt, Copper, Magnesium and Nickel. Using the MPSCRuleLearn algorithm with Top F1 selection. The base models (no EDCR) with the best precision, recall and F1 are each underlined. The overall best performing models in terms of precision, recall, F1 across all models are bolded.

Cobalt			
Model Variant	Precision	Recall	F1
CNN (12)	0.82	<u>0.65</u>	<u>0.73</u>
ATTN (8)	**<u>0.94</u>**	0.19	0.32
CNN (12) (EDCR)	0.80 (-2.15%)	**0.85** (+29.41%)	**0.83 (+13.24%)**
ATTN (8) (EDCR)	**0.94** (0.0%)	0.19 (0.0%)	0.32 (0.0%)

Copper			
Model Variant	Precision	Recall	F1
ATTN (2)	0.78	0.74	<u>0.76</u>
CNN (3)	0.52	<u>0.78</u>	0.62
CNN (4)	**<u>0.83</u>**	0.52	0.64
ATTN (2) (EDCR)	0.79 (+0.5%)	0.76 (+2.33%)	**0.77** (+1.43%)
CNN (3) (EDCR)	0.53 (+0.69%)	**0.84** (+8.89%)	0.65 (+3.84%)
CNN (4) (EDCR)	**0.83** (0.0%)	0.52 (0.0%)	0.64 (0.0%)

Magnesium			
Model Variant	Precision	Recall	F1
CNN (1)	0.52	0.65	<u>0.58</u>
RNN (4)	0.20	<u>0.99</u>	0.33
CNN (2)	**<u>0.86</u>**	0.17	0.28
CNN (1) (EDCR)	0.53 (+3.19%)	0.79 (+21.74%)	**0.64** (+10.67%)
RNN (4) (EDCR)	0.20 (+0.87%)	**1.00** (+1.43%)	0.33 (+0.96%)
CNN (2) (EDCR)	**0.86** (0.0%)	0.17 (+0.0%)	0.28 (0.0%)

Nickel			
Model Variant	Precision	Recall	F1
ATTN (5)	0.59	0.48	<u>0.53</u>
CNN (6)	0.37	<u>0.57</u>	0.45
ATTN (8)	<u>0.66</u>	0.38	0.48
ATTN (5) (EDCR)	0.60 (+1.38%)	0.59 (+24.14%)	**0.60** (+12.85%)
CNN (6) (EDCR)	0.40 (+6.42%)	**0.69** (+20.0%)	0.50 (+11.38%)
ATTN (8) (EDCR)	**0.68** (+2.82%)	0.41 (+8.7%)	0.51 (+6.48%)

Table 3

Model Evaluation Results for Cobalt, Copper, Magnesium and Nickel. Using the MPSCRuleLearn algorithm with Threshold filtering. The base models (no EDCR) with the best precision, recall and F1 are each underlined. The overall best performing models in terms of precision, recall, F1 across all models are bolded.

Cobalt			
Model Variant	Precision	Recall	F1
CNN (12)	0.82	0.65	0.73
ATTN (8)	**0.94**	0.19	0.32
CNN (12) (EDCR)	0.85 (+3.0%)	**0.78** (+19.61%)	**0.81** (+11.63%)
ATTN (8) (EDCR)	**0.94** (0.0%)	0.19 (0.0%)	0.32 (0.0%)

Copper			
Model Variant	Precision	Recall	F1
ATTN (2)	0.78	0.74	0.76
CNN (3)	0.52	0.78	0.62
CNN (4)	0.83	0.52	0.64
ATTN (2) (EDCR)	0.79 (+0.5%)	0.76 (+2.33%)	**0.77** (+1.43%)
CNN (3) (EDCR)	0.53 (+0.69%)	**0.84** (+8.89%)	0.65 (+3.84%)
CNN (4) (EDCR)	**0.85** (+1.54%)	0.57 (+10.0%)	0.68 (+6.6%)

Magnesium			
Model Variant	Precision	Recall	F1
CNN (1)	0.52	0.65	0.58
RNN (4)	0.20	0.99	0.33
CNN (2)	**0.86**	0.17	0.28
CNN (1) (EDCR)	0.53 (+3.19%)	0.79 (+21.74%)	**0.64** (+10.67%)
RNN (4) (EDCR)	0.19 (-0.52%)	**1.00** (+1.43%)	0.33 (-0.2%)
CNN (2) (EDCR)	**0.86** (0.0%)	0.17 (+0.0%)	0.28 (0.0%)

Nickel			
Model Variant	Precision	Recall	F1
ATTN (5)	0.59	0.48	0.53
CNN (6)	0.37	0.57	0.45
ATTN (8)	**0.66**	0.38	0.48
ATTN (5) (EDCR)	0.61 (+2.59%)	0.56 (+17.24%)	**0.58** (+10.23%)
CNN (6) (EDCR)	0.37 (-0.12%)	**0.74** (+28.57%)	0.49 (+9.5%)
ATTN (8) (EDCR)	**0.66** (+0.21%)	0.44 (+17.39%)	0.53 (+10.49%)

4.3. Ablation Studies

We studied the effects of ablating different models from the condition set C used by the EDCR algorithm, where one model's output corrects another under different hyperparameters or architectures, impacting the precision and recall of the base models. Our findings indicate that excluding LSTM and RNN-based results from the condition set C generally has little to no effect on overall model performance, supporting our initial assumption that they would not contribute to improved overall performance of the ensemble. Specifically, our ablation analysis in Figure 1 on the CNN (1) model (left chart) for predicting Magnesium price spikes showed that removing CNN Attention from the conditions resulted in a drop in both Precision (-1%) and Recall (-13%) while removing other configurations of CNN only caused a drop in Recall (6%). Similarly, for the CNN (5) model (right chart) used in predicting Cobalt price spikes, the exclusion of both differently configured CNN and CNN Attention from the conditions led to moderate decreases in Recall (-19% and -18% respectively), with slight increases in Precision (+2%). It should be noted that in both these situations where a CNN was used as the base model and other CNN configurations were excluded, there was a performance decline, indicating that these variations could detect price spikes that the base model missed.

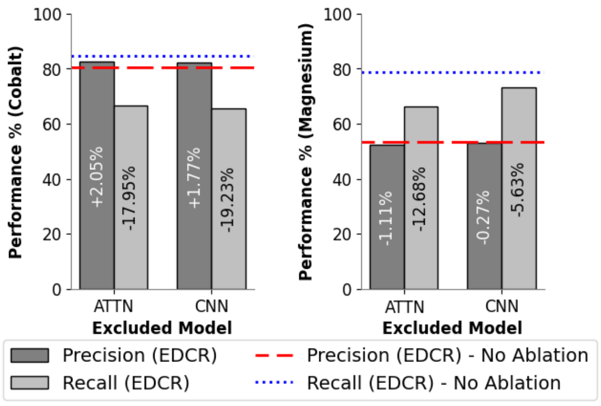

Fig. 1. Ablation analysis of EDCR on ATTN (1) and model (left) and CNN (5) model (right) performance metrics for predicting Magnesium and Cobalt price spikes respectively. The bar charts demonstrate varying Precision and Recall tradeoffs when different models are used as rules for the base model.

5. Conclusion

In this paper, we use a variant of error detection and correction rules (EDCR) to ensemble multiple neural models to predict spikes in the prices of various metals. Using this technique, we obtained significant improvements over a single model as we found through ablation studies that there is variation on the performance of individual models - particularly impacting recall. Further analysis into other metals–such as Lithium–is a direction for future work. Another direction is moving beyond the problem of this paper – using predicting spikes in metal prices (classification) - to use regression to predict the actual price. Additionally, a shortcoming of this paper is that the results from applying EDCR results are deterministic. As such, probabilistic semantics is another direction for future work.

6. Acknowledgment

Some of the authors are supported by ONR grant N00014-23-1-2580 and ARO grant W911NF-24-1-0007.

References

[1] G. Astudillo, R. Carrasco, C. Fernández-Campusano and M. Chacón, Copper Price Prediction Using Support Vector Regression Technique, *Applied Sciences* **10**(19) (2020). doi:10.3390/app10196648. https://www.mdpi.com/2076-3417/10/19/6648.
[2] K. Drachal, Forecasting prices of selected metals with Bayesian data-rich models, *Resources Policy* **64** (2019), 101528. doi:https://doi.org/10.1016/j.resourpol.2019.101528. https://www.sciencedirect.com/science/article/pii/S0301420719306713.
[3] Y. Hu, J. Ni and L. Wen, A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction, *Physica A: Statistical Mechanics and its Applications* **557** (2020), 124907. doi:https://doi.org/10.1016/j.physa.2020.124907. https://www.sciencedirect.com/science/article/pii/S0378437120304696.
[4] E. Kahraman and G. Ünal, Multiple Wavelet Coherency Analysis and Forecasting of Metal Prices, 2016. https://arxiv.org/abs/1602.01960.
[5] J.S. Kricheli, K. Vo, A. Datta, S. Ozgur and P. Shakarian, Error Detection and Constraint Recovery in Hierarchical Multi-Label Classification without Prior Knowledge, in: *Proc. CIKM*, 2024.
[6] M.K. Mohanty, P.K.G. Thakurta and S. Kar, Agricultural Commodity Price Prediction Model: A Machine Learning Framework, *Neural Computing and Applications* **35** (2023), 15109–15128. doi:10.1007/s00521-023-08528-7.

[7] T. Shi, C. Li, W. Zhang and Y. Zhang, Forecasting on metal resource spot settlement price: New evidence from the machine learning model, *Resources Policy* **81** (2023), 103360. doi:https://doi.org/10.1016/j.resourpol.2023.103360. https://www.sciencedirect.com/science/article/pii/S0301420723000685.

[8] B. Xi, K. Scaria, D. Bavikadi and P. Shakarian, Rule-Based Error Detection and Correction to Operationalize Movement Trajectory Classification, 2024. https://arxiv.org/abs/2308.14250.

[9] H. Zhang, H. Nguyen, X.-N. Bui, B. Pradhan, N.-L. Mai and D.-A. Vu, Proposing two novel hybrid intelligence models for forecasting copper price based on extreme learning machine and meta-heuristic algorithms, *Resources Policy* **73** (2021), 102195. doi:https://doi.org/10.1016/j.resourpol.2021.102195. https://www.sciencedirect.com/science/article/pii/S0301420721002099.

[10] J. Zhou, Z. He, Y.N. Song, H. Wang, X. Yang, W. Lian and H.-N. Dai, Precious Metal Price Prediction Based on Deep Regularization Self-Attention Regression, *IEEE Access* **8** (2020), 2178–2187. doi:10.1109/ACCESS.2019.2962202.

[11] Investing.com: Financial Markets News, Data, and Analysis, Accessed: 2024-09-05.

[12] U. Çelik and C. Başarır, The Prediction of Precious Metal Prices via Artificial Neural Network by Using RapidMiner, *Alphanumeric Journal* **5**(1) (2017), 45–54–. doi:10.17093/alphanumeric.290381.

7. Appendix

Table 4 contains the various hyperparameters that were applied across all variations of each classification model.

Table 4

Variations of specific hyperparameters across the classification models used in the experiments.

RNN and LSTM	
Variant	Layers
1	32
2	64
3	128
4	256

CNN and ATTN	
Variant	Filters, Kernel Size
1	32, 7
2	32, 5
3	32, 3
4	64, 7
5	64, 5
6	64, 3
7	128, 7
8	128, 5
9	128, 3
10	256, 7
11	256, 5
12	256, 3