Accredited SINTA 2 Ranking
Decree of the Director General of Higher Education, Research, and Technology, No. 158/E/KPT/2021
Validity period from Volume 5 Number 2 of 2021 to Volume 10 Number 1 of 2026

Published online at: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 8 No. 5 (2024) 644 - 650 e-ISSN: 2580-0760

Machine Learning Methods for Forecasting Intermittent Tin Ore Production

Nabila Dhia Alifa Rahmah[1*], Budhi Handoko[2], Anindya Apriliyanti Pravitasari[3]
[1,2,3]Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Bandung, Indonesia
[1]nabila17035@mail.unpad.ac.id, [2]budhi.handoko@unpad.ac.id, [3]anindya.apriliyanti@unpad.ac.id

Abstract

Effective production forecasting is important for resource planning and management in the mining industry. Tin ore production from Cutter Section Dredges (CSD) may fluctuate due to a variety of factors, in which there are periods when the production is zero. This study compares various combinations of machine learning-based classification and forecasting to predict future tin ore production values, which have not been found in previous studies. The presence of zero values in the forecast in the next day's tin ore production forecast is addressed by combining classification and forecasting techniques. Random Forest and CatBoost classification techniques are used to determine the next day's CSD production operating status. Then, for each time point when the CSD is operational, a forecasting model is created using CatBoost and Bi-LSTM. This study's findings show that a serial combination of the Random Forest classification method and CatBoost forecasting can produce accurate tin ore production forecasts for the selected CSD (RMSE = 0.271, MAE = 0.179, MAE = 0.730, F1-score = 0,80). This study demonstrates how a serial combination of classification and forecasting models can improve the accuracy and efficiency of production forecasting for intermittent time series data.

Keywords: forecasting; classification, machine learning; mining; CatBoost

How to Cite: N. D. A. Rahmah, B. Handoko, and A. A. Pravitasari, "Machine Learning Methods for Forecasting Intermittent Tin Ore Production", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 8, no. 5, pp. 644 - 650, Oct. 2024.
DOI: https://doi.org/10.29207/resti.v8i5.5990

1. Introduction

Tin ore production plays a crucial part in the mining industry as it provides raw materials needed for a variety of industries. Effective forecasting of tin ore production aims to provide an overview of future trends in tin ore production and may aid in resource planning and management.

Offshore tin ore production using Cutter Section Dredges (CSDs) varies throughout the year due to changing weather, equipment and operational factors, and other field conditions, as well as unpredictable docking times that can result in zero production on some days. Up to 30.3% of the production data used in this study contain zero values.

Previous studies have been done on the prediction and forecasting of various mining outcomes, employing a variety of machine-learning techniques that combine historical production data with specific operational data for each mine type [1] - [8]. However, discussions on production forecasting, particularly for tin ore, are difficult to find.

Several studies have also examined forecasting data with a large number of zero values, which is common in the context of intermittent demand forecasts. [9], [10] take an interesting approach to overcome this problem by splitting it into two tasks: classifying the presence/absence of demand and predicting the magnitude of demand, which can provide more accurate results than the classical Croston method.

Deep learning methods, such as LSTM, are commonly used in time series data analysis due to their ability to process data in sequences and remember past information. However, other machine learning methods, such as gradient boosting, are more interpretable [11] and also applicable to make predictions with zero values, as seen in this study [9]. Random forest and gradient boosting are methods that can be used for time series data processing and classification.

Received: 13-08-2024 | Accepted: 02-10-2024 | Published Online: 15-10-2024

This study aims to build a model to forecast the tin ore production results of a chosen CSD for the following day. The model is built using a variety of machine learning methods and divided into two serial phases: classification and forecasting. Random Forest and CatBoost classification models are used to determine the CSD's operational status, addressing the zero values. The CatBoost and Bi-LSTM forecasting models are then used to predict tin ore production at the time step when the classification results indicate that the CSD is operational. As a result, the classification process must take precedence over the forecasting process.

In this study, the combined models are built using a combination of decision tree-based methods (Random Forest and CatBoost) and neural network techniques (Bi-LSTM). This is an extension of previous studies [9], [10] and allows for the comparison of combined models to determine the most effective configuration.

Separating intermittent data forecasting tasks into classification and regression tasks allows for easier evaluation and optimization for each of the tasks [10]. The classification phase results in this study are evaluated using a variety of metrics, including precision, recall, and F1 score. Meanwhile, the forecasting results for the task are evaluated using RMSE, MAE, and MASE. This combined approach, utilizing historical production data as the input, is expected to yield accurate forecasting results.

2. Research Methods

The flowchart in Figure 1 illustrates the stages of analysis for this study, while Figure 2 provides an overview of the combined methods used in the study . Inspired by [9], [10], we employ a machine learning-based methodology and follow a similar procedure. The forecasting model is built using various machine learning methods and divided into two serial tasks: classification and forecasting. Random Forest and CatBoost classification models are used to determine the CSD's operational status for the next day (operational or non operational/zero productions), while the CatBoost and Bi-LSTM forecasting models are used to forecast the size of the production value if the CSD is determined to be operational in the particular timestep. In the classification phase, the Random Forest classifier and CatBoost classifier models are compared, which are two types of decision tree ensembles that perform well in classification. During the forecasting phase, the Bi-LSTM and CatBoost models are compared. The Bi-LSTM model excels at processing sequence data such as time series data, whereas CatBoost is a type of gradient boosting that has been used in previous similar studies. All of the analyses in this study were carried out on Google Colab with Python 3.10.

2.1 Research Data and Feature Engineering

This study makes use of daily production data on tin ore production (measured in Ton of Sn) from CSD A, which was acquired from Company X. The data spans 1449 days within the timespan of 2020-2023, in which 439 days (30.3% of the data) are zeroes. In addition to the production data, several features are created to aid in the classification and forecasting process, using machine learning methods inspired by previous studies and tailored to the context of this research data. The CSD's operating status is a binary feature that describes when the ship is operational (has production value) and when it is non-operational (zero production). Discrete-time features (year, month, date, and day) are also included [9], [10]. In addition, several features related to the characteristics of intermittent data [9] were added, including time since the last CSD operation (number of days since the last non-zero production value) and time since the last CSD operation (number of days since the last zero production value). This study utilizes up to 14 days of lagged data to forecast the next day's tin ore production. Table 1 summarizes the variables used in each of the study's phases.

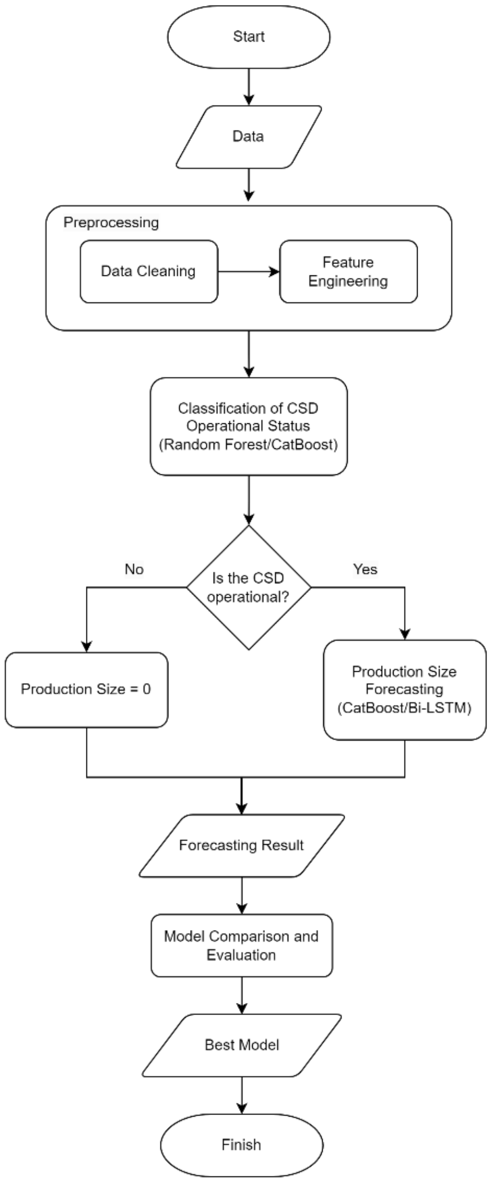

Figure 1 Flowchart of Research Plan

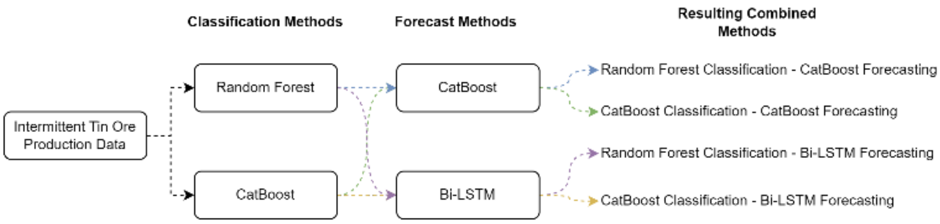

Figure 2 Flowchart of Combined Methods

Table 1. Features Used for Each Study Phase

Phase	Machine Learning Algorithm	Data
Classification	Random Forest	Historical Production Data + Additional Features
	CatBoost	
Forecasting	CatBoost	Historical Production Data (Non Zero) + Additional Features
	Bi-LSTM	Historical Production Data

2.2 Classification and Forecasting Model Development

Random Forest is one of the most popular machine learning algorithms in classification tasks that can provide good prediction results [12]. It is first developed by Breiman [13]. Random Forest is an ensemble learning which is built on decision trees as the basis. Each decision tree in the Random Forest model is formed by selecting attributes at random to determine the splitting of the tree, as well as taking subsamples of the data using bootstrap [13], [14]. When combining the tree models, Random Forest employs a bagging (bootstrap aggregation) algorithm [15]. In this study, Random Forest Classifier and Random Forest Regressor were implemented using scikit-learn [14] to perform classification tasks.

Random Forest can limit the training time and avoid *overfitting* [12], deal with *outliers* and is not dependent on the distribution of the data [16], and is also applicable to a large number of input features [12]. This makes Random Forest to be one of the easiest machine-learning algorithms to implement. However, the Random Forest model's hyperparameters can have an impact on the accuracy of the predictions [12]. In this study, a grid search technique was used to determine the best hyperparameters to apply to the dataset. Table 2 shows the selected random forest architecture used in the study.

Table 2. Random Forest Architecture

Hyperparameter	Value
Number of *Decision Tree* (n_estimators)	100
Max Depth of *Decision Tree*	5
Max features for split *node*	None
Min sample for *leaf*	2

Gradient Boosting, like Random Forest, is a machine-learning algorithm that consists of an ensemble of multiple decision trees. However, when combining the results of different decision trees into its ensemble, the Gradient Boosting model takes a different approach, known as "boosting". The boosting algorithm is built by systematically combining multiple weak learners into a strong learner [17], with each tree learning from the previous tree's errors to improve output prediction results [18]. This algorithm can also be applied to classification and regression tasks [19].

This research focuses on the CatBoost algorithm [20], [21], which is a development of the gradient-boosting algorithm with decision trees as its base predictor [21], [22]. CatBoost has advantages when processing categorical data and can reduce overfitting, producing competitive results and with faster processing times than other gradient boosting algorithms [20], [21]. CatBoost works well with heterogeneous, noisy, and complex data [19], [22].

The two main principles of the CatBoost algorithm include the processing of categorical data with target statistics and innovation in overcoming gradient bias or prediction shift with ordered boosting [20], [21]. In addition, CatBoost uses a combination of different category features as additional features to be able to capture more complex patterns that are taken with a greedy approach to the splitting of the decision tree, which can also include numerical features [20].

In this study, the CatBoost classification and forecasting method is implemented in Python using the CatBoost package. Table 2 shows the CatBoost model architecture after hyperparameter tuning.

Table 3. CatBoost Architecture

Model	*Hyperparameter*	Value
Classification of CSD's Operational Status	Number of iterations	500
	Max depth of *Decision Tree*	6
	Max combinations of features (*max_ctr_complexity*)	2
	Learning rate	0,1
Forecasting of Production Magnitude	Number of iterations	500
	Max depth of *Decision Tree*	5
	Max combinations of features (*max_ctr_complexity*)	2
	Learning rate	0,03

Long Short-Term Memory (LSTM) is a neural network algorithm introduced by Sepp Hochreiter and Jürgen Schmidhuber [23]. LSTM is an improvement to the Recurrent Neural Network (RNN) that is applicable to sequence data, able to take information from previous timesteps into account. LSTM is an RNN architecture that has a special unit to overcome the problem of vanishing gradients. LSTM has a memory cell that can store data for a long time, with three gates that regulate the entry and exit of information from each cell [24]. This keeps LSTM from the vanishing or exploding of error from backpropagations, overcoming the general vanishing gradient problem in RNNs and increasing accuracy [25]. The Forget Gate enables LSTM to control memory cell resetting by determining which information is stored or deleted when no longer needed [24]. Meanwhile, the input gate determines whether the memory cell is updated by controlling the information entering the cell [24]. Finally, the output gate controls the information leaving the cell [24], [25].

Bidirectional LSTM, also known as Bi-LSTM, is an advancement of LSTM in which the sequence data is processed in two directions using forward and backwards hidden layers that are then connected to the output layer [17]. This yields information from before and after a particular point in the sequence data being processed [17]. With its numerous benefits, Bi-LSTM has been shown to outperform LSTM in a variety of applications.

In general, the process in Bi-LSTM is similar to LSTM, but it also involves bidirectional information processing, specifically with forward and backwards hidden layers [17]. In deep bidirectional LSTM, two LSTMs are applied to the input data, the first to the input sequence (forward layer), which is then reversed (backward layer), increasing the ability to understand long-term dependencies in the sequence data [26].

The input to the Bi-LSTM model is formed as a 3-dimensional array using the sliding window method and is normalized with the min-max scaler prior to entering the model. The Bi-LSTM method is implemented in Python with TensorFlow [27] dan Keras [28]. Table 4 shows the architecture and hyperparameter tuning design for the LSTM and Bi-LSTM models, while Figure 2 illustrates the training loss.

Table 4. Bi-LSTM Architecture

Hyperparameter	Value
Number of layers	2
Number of units	64, 64
Activation function	ReLU
Optimizer	Adam
Dropout	0,1
L2 Regularizer	0,1
Batch	32
Learning rate	0,00015992
Epoch	22 (Early Stopping)

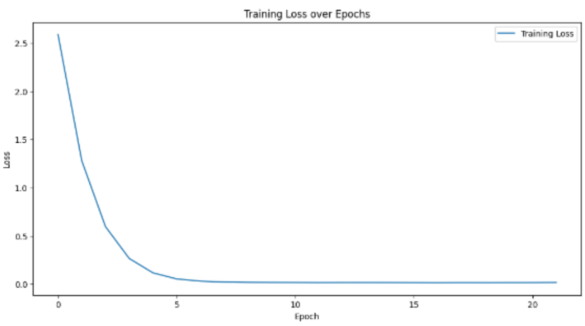

Figure 3 Training Loss and Epoch of the Bi-LSTM Model

2.3 Model Evaluation

Several metrics are used to assess the CSD's operational status classification model, which comprises recall/sensitivity, precision, and F1-score. These metrics are based on the values of TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative), indicating the agreement between the class obtained from the classification result and the actual class [29].

Meanwhile, the metrics Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Squared Error (MASE) are used to assess the final forecasting result of the serial combination of classification and forecasting method in this study. RMSE and MAE are scale-dependent metrics, therefore the resulting value has the unit of the actual data (Ton of Sn), while MASE is a scale-free error. The MASE is a useful metric for evaluating forecast results on sparse time series data [10], as it does not have to concern about having zero values in the denominator, and will have an infinite value unless the historical value is always the same [9]. The scaled error (q_t) and MASE is calculated as Formula 1 and 2 [30]:

$$q_t = \frac{e_t}{\frac{1}{n-1}\sum_{i=2}^{n}|Y_i - Y_{i-1}|} \quad (1)$$

$$MASE = mean(|q_t|) \quad (2)$$

3. Results and Discussions

3.1 Result for the Serial Combination of CatBoost Classification and Bi-LSTM Forecasting Model

The Serial Combination of CatBoost Classification Model - Bi-LSTM Forecasting Model is applied to forecast the test data. Table 4 and Figure 4 describes the evaluation of the CatBoost classifier when predicting the CSD's operational status, while Figure 5 compares the forecasting results from the 1-day-ahead model applied to the test data as compared to the actual data. This model does a good job of identifying zero data patterns, but it produces flat forecasts on data that has been identified as having a production value ($RMSE = 0{,}255, MAE = 0{,}159, MASE = 0{,}649$).

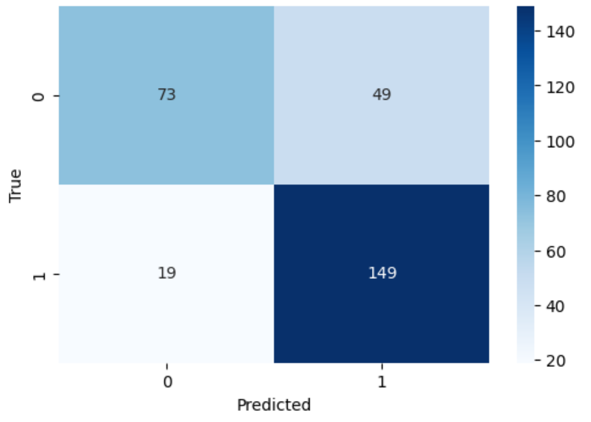

Figure 4 Confusion Matrix for the CatBoost Classification Model on Test Data

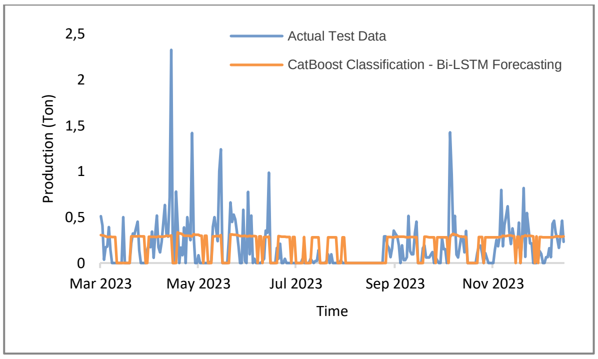

Figure 5 Forecasting Result of the Serial Combination of CatBoost Classification – Bi-LSTM Forecasting Model on Test Data

3.2 Result for the Serial Combination of CatBoost Classification and CatBoost Forecasting Model

The Serial Combination of CatBoost Classification Model - CatBoost Forecasting Model is applied to forecast the test data and then compared to the actual test data, as illustrated in Figure 6. As with the previous model, this model does a good job of identifying zero data patterns, but it can also provide good forecasts on data that has been identified as having a production value $(RMSE = 0{,}277\,, MAE = 0{,}188,\ MASE = 0{,}769)$. This result fit the data better than The Serial Combination of CatBoost Classification Model - Bi-LSTM Forecasting Model.

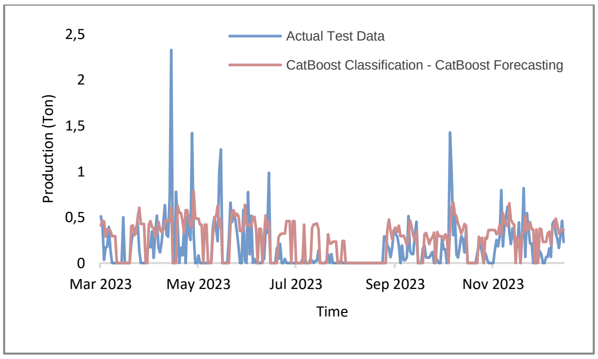

Figure 6. Forecasting Result of the Serial Combination of CatBoost Classification – CatBoost Forecasting Model on Test Data

3.3 Result for the Serial Combination of Random Forest Classification and Bi-LSTM Forecasting Model

The Serial Combination of Random Forest Classification Model - Bi-LSTM Forecasting Model is also applied to forecast the test data. Table 4 and Figure 7 describes the evaluation of the Random Forest classifier when predicting the CSD’s operational status, while Figure 8 compares the forecasting results from the model for the test data to the actual data. The Random Forest classification model also does a good job of identifying zero data patterns, but the final result of the serial combination model produces flat forecasts on data that has been identified as having a production value $(RMSE = 0{,}252, MAE = 0{,}150,\ MASE = 0{,}614)$.

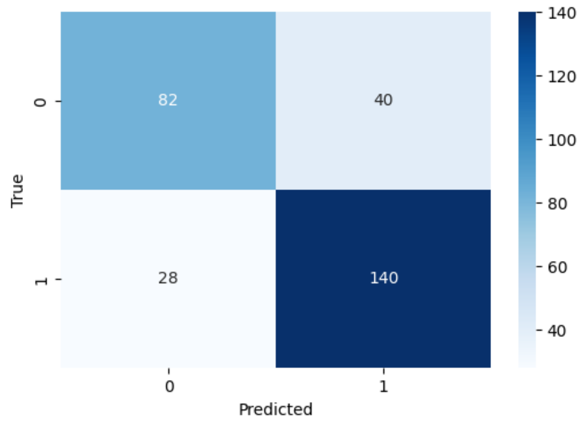

Figure 7 Confusion Matrix for the Random Forest Classification Model on Test Data

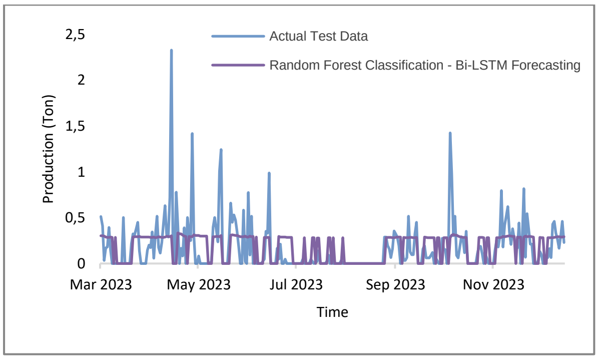

Figure 8. Forecasting Result of the Serial Combination of Random Forest Classification – Bi-LSTM Forecasting Model on Test Data

3.4 Result for the Serial Combination of Random Forest Classification and CatBoost Forecasting Model

At last, the Serial Combination of Random Forest Classification Model - CatBoost Forecasting Model is also applied to forecast the test data. Figure 9 compares the forecasting results from the model for the test data to the actual data. The Random Forest classification model does a good job at both identifying zero data patterns and also producing fitting forecast value for the production magnitudes $(RMSE = 0{,}271\,, MAE = 0{,}179,\ MASE = 0{,}730)$. This result is comparable to the Serial Combination of CatBoost Classification Model - CatBoost Forecasting Model.

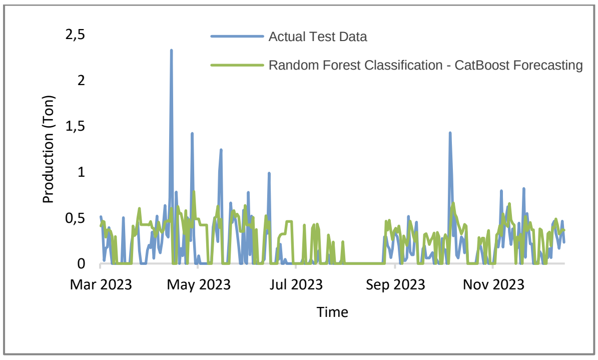

Figure 9. Forecasting Result of the Serial Combination of Random Forest Classification – CatBoost Forecasting Model on Test Data

Table 5. Metrics Comparison of All Serial Combination Methods

No	Classification Model	Forecasting Model	Evalution on Testing Data					
			Precision (1)	Recall (1)	F1-Score (1)	RMSE	MAE	MASE
1	CatBoost	Bi-LSTM	0,75	0,89	0,81	0,255	0,159	0,649
2	CatBoost	CatBoost	0,75	0,89	0,81	0,277	0,188	0,769
3	Random Forest	Bi-LSTM	0,78	0,83	0,80	0,252	0,150	0,614
4	Random Forest	CatBoost	0,78	0,83	0,80	0,271	0,179	0,730

3.5 Discussions

Finally, Table 5 summarizes the accuracy of all serial combination methods' forecast as compared to the actual test data, as can also be observed graphically on Figure 5-9.

The Serial Combination of Random Forest Classification and Bi-LSTM Forecasting Model achieves the best results in metric evaluation. However, when examining the graph form more closely in Figure 8, we can see that the model tends to produce nearly flat forecast results for tin ore production at time steps identified as having production value.

Alternatively, the Serial Combination of CatBoost Classification and CatBoost Forecasting Model and the Serial Combination of Random Forest Classification and CatBoost Forecasting Model both produce good results in identifying periods when the CSD is non-operational (zero production value) as well as in the forecasting of the tin ore production magnitude. Among the two models, the Random Forest Classification and CatBoost Forecasting Model produces result that is best suited to the data, as can be seen in Figure 9. Therefore, the model is chosen as the best model for the next day forecast of the daily tin ore production of the CSD.

The findings from this study show that using decision tree-based classification methods (Random Forest and CatBoost) prior to the forecasting step can help distinguish between the CSD's non-operational and operational periods, which aligns well with the previous study [10]. However, when the classification methods are combined with the neural network-based forecasting method (Bi-LSTM), the resulting models tend to produce forecast results that are flat over the operational period and can not reflect the fluctuations of the data. In contrast, combinations of the classification methods with the CatBoost forecasting method, which is also based on a decision tree, produce forecast results that are more responsive to data fluctuations. This result is consistent with the previous study by [9] that shows that combining classification and forecasting methods utilizing LightGBM, also based on a decision tree, can produce favorable results for intermittent data. However, it is worth noting that the methods used in this study are still unable to produce accurate forecasts for outliers.

4. Conclusions

This study attempts to forecast one day-ahead tin ore production of a selected CSD with highly varying data and intermittent operational periods (30.3% of the data has zero values). A proposed solution to this problem is to utilize a serial combination of classification and forecasting models using Machine Learning algorithms. The findings of this study show that the Serial Combination of Random Forest Classification and CatBoost Forecasting Model can provide good results in forecasting tin ore production from the CSD for the next day while also accurately identifying periods when the CSD is non operational or has zero values with $RMSE = 0{,}271$, $MAE = 0{,}179$, $MASE = 0{,}730$, and $F1 - score = 0{,}80$. In general, the proposed serial combinations of classification and forecasting models are able to predict the next day's CSD operational status and forecast the tin ore production size with adequate results. This demonstrates that the method used in this study is capable of capturing the dynamics of fluctuating tin ore production, including overcoming challenges such as a high number of zero-value productions. The presence of the classification method differentiates between zero and non-zero values, allowing for more accurate predictions. To improve the accuracy of the proposed method in future research, additional data and related operational variables can be added to consideration. Likewise, more complex machine learning and deep learning methods can also be utilized to improve forecasting performance, chosen based on the characteristics of the data. For example, the combination of decision tree-based classification methods and neural network forecasting methods can be tried on a more homogenous or non-fluctuating dataset. In the presence of outliers, future research can also look into appropriate methods to deal with extreme values in addition to accommodating zero values. This research has shown the potential for the proposed serial combination model to be considered in practical applications with an aim to support operational decisions in related fields.

Acknowledgements

We would like to thank Company X and Universitas Padjadjaran for their support of this research. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. The author declares no conflict of interest.

References

[1] Y. Choi, H. Nguyen, X. N. Bui, T. Nguyen-Thoi, and S. Park, "Estimating Ore Production in Open-pit Mines Using Various Machine Learning Algorithms Based on a Truck-Haulage System and Support of Internet of Things," *Natural Resources Research*, vol. 30, no. 2, pp. 1141–1173, Apr. 2020, doi: 10.1007/s11053-020-09766-5.

[2] D. Fan, H. Sun, J. Yao, K. Zhang, X. Yan, and Z. Sun, "Well production forecasting based on ARIMA-LSTM model considering manual operations," *Energy*, vol. 220, Apr. 2021, doi: 10.1016/j.energy.2020.119708.

[3] F. Abdullayeva and Y. Imamverdiyev, "Development of oil production forecasting method based on deep learning," *Statistics, Optimization and Information Computing*, vol. 7, no. 4, pp. 826–839, 2019, doi: 10.19139/soic-2310-5070-651.

[4] C. P. Obite, A. Chukwu, D. C. Bartholomew, U. I. Nwosu, and G. E. Esiaba, "Classical and machine learning modeling of crude oil production in Nigeria: Identification of an eminent model for application," *Energy Reports*, vol. 7, pp. 3497–3505, Nov. 2021, doi: 10.1016/j.egyr.2021.06.005.

[5] B. M. Negash and A. D. Yaw, "Artificial neural network based production forecasting for a hydrocarbon reservoir under water injection," *Petroleum Exploration and Development*, vol. 47, no. 2, pp. 383–392, Apr. 2020, doi: 10.1016/S1876-3804(20)60055-6.

[6] S. Wang, Z. Chen, and S. Chen, "Applicability of deep neural networks on production forecasting in Bakken shale reservoirs," *J Pet Sci Eng*, vol. 179, pp. 112–125, Aug. 2019, doi: 10.1016/j.petrol.2019.04.016.

[7] C. Fan, N. Zhang, B. Jiang, and W. V. Liu, "Using deep neural networks coupled with principal component analysis for ore production forecasting at open-pit mines," *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 16, no. 3, pp. 727–740, Mar. 2024, doi: 10.1016/j.jrmge.2023.06.005.

[8] W. Kaleem, S. Tewari, M. Fogat, and D. A. Martyushev, "A hybrid machine learning approach based study of production forecasting and factors influencing the multiphase flow through surface chokes," *Petroleum*, vol. 10, no. 2, pp. 354–371, Jun. 2024, doi: 10.1016/j.petlm.2023.06.001.

[9] X. Zhuang, Y. Yu, and A. Chen, "A combined forecasting method for intermittent demand using the automotive aftermarket data," *Data Science and Management*, vol. 5, no. 2, pp. 43–56, Jun. 2022, doi: 10.1016/j.dsm.2022.04.001.

[10] J. M. Rožanec, B. Fortuna, and D. Mladenić, "Reframing Demand Forecasting: A Two-Fold Approach for Lumpy and Intermittent Demand," *Sustainability (Switzerland)*, vol. 14, no. 15, Aug. 2022, doi: 10.3390/su14159295.

[11] R. Szczepanek, "Daily Streamflow Forecasting in Mountainous Catchment Using XGBoost, LightGBM and CatBoost," *Hydrology*, vol. 9, no. 12, Dec. 2022, doi: 10.3390/hydrology9120226.

[12] İ. Güven, Ö. Uygun, and F. Şimşir, "Machine learning algorithms with intermittent demand forecasting: An application in retail apparel with plenty of predictors," *Tekstil ve Konfeksiyon*, vol. 31, no. 2, pp. 99–110, Jun. 2021, doi: 10.32710/tekstilvekonfeksiyon.809867.

[13] L. Breiman, "Random Forests," 2001.

[14] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, Accessed: Jul. 16, 2024. [Online]. Available: http://scikit-learn.sourceforge.net.

[15] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining Practical Machine Learning Tools and Techniques*, Fourth Edition. 2017. [Online]. Available: https://www.elsevier.com

[16] M. Aghaabbasi, Z. A. Shekari, M. Z. Shah, O. Olakunle, D. J. Armaghani, and M. Moeinaddini, "Predicting the use frequency of ride-sourcing by off-campus university students through random forest and Bayesian network techniques," *Transp Res Part A Policy Pract*, vol. 136, pp. 262–281, Jun. 2020, doi: 10.1016/j.tra.2020.04.013.

[17] F. Zhang, H. Fleyeh, and C. Bales, "A hybrid model based on bidirectional long short-term memory neural network and Catboost for short-term electricity spot price forecasting," *Journal of the Operational Research Society*, vol. 73, no. 2, pp. 301–325, 2022, doi: 10.1080/01605682.2020.1843976.

[18] J. A. Irvin *et al.*, "Incorporating machine learning and social determinants of health indicators into prospective risk adjustment for health plan payments," *BMC Public Health*, vol. 20, no. 1, pp. 1–10, 2020, doi: 10.1186/s12889-020-08735-0.

[19] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *J Big Data*, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00369-8.

[20] A. V. Dorogush, V. Ershov, and A. G. Yandex, "CatBoost: gradient boosting with categorical features support," in *Workshop on ML Systems at NIPS 2017*, 2017. [Online]. Available: https://github.com/Microsoft/LightGBM

[21] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada*, 2018. [Online]. Available: https://github.com/catboost/catboost

[22] S. A. Shahriar *et al.*, "Potential of arima-ann, arima-svm, dt and catboost for atmospheric pm2.5 forecasting in bangladesh," *Atmosphere (Basel)*, vol. 12, no. 1, pp. 1–21, Jan. 2021, doi: 10.3390/atmos12010100.

[23] Sepp Hochreiter and Jurgen Schmidhuber, "Long Short-Term Memory," *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," Nov. 01, 2021, *Springer*. doi: 10.1007/s42979-021-00815-1.

[25] M. Chovatiya, A. Dhameliya, J. Deokar, J. Gonsalves, and A. Mathur, "Prediction of dengue using recurrent neural network," *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, vol. 2019-April, no. ICOEI 2019, pp. 926–929, 2019, doi: 10.1109/icoei.2019.8862581.

[26] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, p. 3285.

[27] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Accessed: Jul. 17, 2024. [Online]. Available: tensorflow.org

[28] F. Chollet and et al., "Keras," 2015. Accessed: Jul. 17, 2024. [Online]. Available: https://keras.io

[29] X. Zhou, P. Lu, Z. Zheng, D. Tolliver, and A. Keramati, "Accident Prediction Accuracy Assessment for Highway-Rail Grade Crossings Using Random Forest Algorithm Compared with Decision Tree," *Reliab Eng Syst Saf*, vol. 200, Aug. 2020, doi: 10.1016/j.ress.2020.106931.

[30] R. J. Hyndman, "Another Look at Forecast Accuracy Metrics for Intermittent Demand," *FORESIGHT: The International Journal of Applied Forecasting*, vol. June, no. 4, pp. 43–46, 2006.