A Comprehensive Evaluation of CatBoost and LightGBM Algorithms for Honorarium Prediction on Categorical Datasets with Class Imbalance

Slamet Widodo[1*], Fandy Setyo Utomo[2], Berlilana[3]
[1,2,3]*Magister of Computer Science, Postgraduate Program, Universitas Amikom Purwokerto, Indonesia*
*corr-author: swidodo009@gmail.com

Abstract - Determining income, including honoraria in the academic environment, is often done manually and subjectively, necessitating a predictive model to objectively determine the honorarium amount. However, the development of the prediction model faces challenges due to the dataset's characteristics, which include categorical data and an imbalanced class distribution. This research aims to evaluate the predictive performance and computational resource efficiency of the CatBoost and LightGBM algorithms in predicting honorariums. The dataset used includes 58,332 actual honorarium data of employees from higher education institution "A" in Purwokerto for the period from January 2024 to February 2025. The methods used include data preprocessing, dataset splitting using Stratified Splitting, modeling with CatBoost, LightGBM, Random Forest, Neural Network, and Linear Regression, as well as evaluation using MSE, RMSE, MAE, R^2 metrics, and computational resources (execution time, memory, CPU time). LightGBM achieved an RMSE of 665.960 and an R^2 of 0.54, while recording the lowest memory usage at only 2.67 MB. CatBoost produced an RMSE of 667.395 and an R^2 of 0.53, excelling in processing categorical features without one-hot encoding. Meanwhile, Linear Regression showed the lowest accuracy and high memory usage. These results confirm that LightGBM is the most optimal choice for fast, efficient, and accurate honorarium predictions. However, this research is limited to testing in a laboratory environment. Further research is recommended to implement direct integration with an active database and the integration of information retrieval methods to enhance the effectiveness and security of real-time honorarium predictions, as well as to integrate interpretability methods such as SHAP to improve decision-making transparency.

Keywords: CatBoost, LightGBM, honorarium, categorical data, imbalanced classes, computational efficiency

I. INTRODUCTION

Determining income, including honoraria in the academic environment, often faces challenges because it tends to be done manually and subjectively, which can lead to injustice and dissatisfaction among the recipients [1-2]. Several studies have developed income or salary prediction models. One study revealed that the Neural Network algorithm can achieve good accuracy in salary prediction (83.2%), but requires a relatively long training time (82.79 seconds) [1]. Another study introduced a new salary prediction system using machine learning techniques, but still showed weaknesses in terms of categorical data complexity and class imbalance [3]. The LSTM model has been successfully optimized to improve prediction accuracy, but it still faces high computational resource requirements [4]. Ordinal regression has also been implemented and provided accurate results, but it requires a complex and intensive encoding process [5].

Other research unrelated to income prediction also faces similar challenges. The development of a generic model to predict student outcomes also faces challenges in handling categorical features and imbalanced classes [6]. Random Kernel Forests offer improved accuracy but still face challenges in managing data with many categorical features [7]. Another method proposing the acceleration of recurrent neural network training still has weaknesses in computational resource usage [8]. Another approach uses recurrent neural networks with adaptive training strategies, but still encounters obstacles in high memory usage [9].

Various studies have proposed methods to effectively handle imbalanced classes and categorical features. One approach proposes an optimal ensemble model for bankruptcy prediction with imbalanced data, but the method still requires high computational resources [10]. Another study examined the impact of data split composition on breast cancer classification performance, but did not specifically measure computational resource usage [11]. The impact of SMOTE on the performance of the Random Forest classifier has also been studied, which effectively handles the minority class but increases the computational load [12]. The Synthetic Minority Oversampling Technique for tree-boosting-

based models in intrusion detection systems has also been developed, but it still demands significant computational resources due to the increase in data dimensions [13]. Various binary classification methods on imbalanced data result in increased accuracy but also amplify computational complexity [14]. Label Encoding and One-Hot Encoding have been compared in linear regression, with the result that One-Hot Encoding significantly increases the dataset dimensions and causes high memory consumption [15].

Boosting algorithms such as CatBoost and LightGBM have received special attention due to their ability to handle categorical features directly and their effective internal mechanisms in addressing class imbalance. One study comparing the performance of LightGBM and XGBoost in handling data with imbalanced classes found that LightGBM was more effective, although without a specific evaluation of computational resource usage [16]. The CatBoost algorithm has also been utilized for post-harvest quality evaluation of grapes, demonstrating its capability in processing categorical data, although the aspect of computational resource usage was not explicitly discussed [17]. CatBoost has also been applied for diagnosing HVDC system faults using knowledge graphs, but without a detailed evaluation of computational resource usage [18]. The LightGBM technique has also been used in hydrological predictions and has shown high accuracy, although without an assessment of the computational resources used [19]. Evaluation of eight regression algorithms, including LightGBM, for forest biomass estimation places more emphasis on accuracy than computational efficiency [20]. The use of LightGBM has also been optimized in power fingerprint identification [21] and cryptocurrency price trend prediction [22], showing accurate results but without assessing computational resource efficiency. Another study surveyed ensemble learning methods, including CatBoost and LightGBM, highlighting the advantages of both but without in-depth exploration of computational needs [23].

The comparison between previous studies and the conducted research is presented in a table. Table I explains the research focus, advantages, and disadvantages or gaps between previous studies and the conducted research.

TABLE I
COMPARISON OF PREVIOUS RESEARCH

Researchers & Year	**Algorithm / Focus**	**Advantages**	**Disadvantages / Gaps**
R. Kablaoui and A. Salman (2022)	Neural Network for salary prediction	High accuracy (83,2%)	High training latency, does not discuss memory efficiency
F. Li, N. A. Majid, and S. Ding (2024)	LSTM with MLE & prior for salary prediction	High precision with a statistical approach	Complex encoding, high memory demand
G. Ramaswami, T. Susnjak, and A. Mathrani (2022)	Generic EDM model for predicting student outcomes	Adaptive to many educational domains	Still faces difficulties with categorical features and class imbalance
A. Dmitry Devyatkin and G. Oleg Grigoriev (2022)	Random Kernel Forests	High accuracy, innovative approach	Complex in managing categorical features
L. H. Li, R. Ahmad, R. Tanone, and A. K. Sharma (2023)	STB (SMOTE for boosting tree)	Effective for imbalanced data Dimensionality	increase burdens memory
C. Herdian, A. Kamila, and I. G. Agung Musa Budidarma (2024)	Text Encoding: Label vs One-Hot	Detailed Encoding Comparison	One-Hot increases dimensions and memory consumption
Q. Chen, J. Li, J. Feng, and J. Qian (2024)	CatBoost for post-harvest fruit quality	Effective for categorical data	Does not discuss computational performance
V. Kumar, N. Kedam, K. V. Sharma, D. J. Mehta, and T. Caloiero (2023)	LightGBM for river flow prediction	High accuracy, good generalization	No evaluation of resource efficiency
Y. Y. Li, T. Van Do, and H. T. Nguyen (2022)	Survey of ensemble methods (LightGBM, CatBoost, etc.)	Explaining the technical advantages of boosting	Not presenting computational measurements

The main objective of this research is to evaluate the performance of the CatBoost and LightGBM algorithms in predicting honorarium, with a particular emphasis on real-world datasets that have many categorical features and imbalanced classes. In addition to evaluating prediction performance, this research also aims to measure in detail the computational requirements such as memory usage and execution time, which are important aspects but have rarely been studied in depth by previous research, in order to determine the optimal model in terms of both accuracy and efficiency for practical application.

II. METHOD

The dataset used is a real dataset collected from the payment records of honorariums for employees of higher education institution "A" in Purwokerto during the period from January 2024 to February 2025. This dataset consists of 58,332 entries that reflect the actual conditions in the process of providing honoraria. The hardware used is an Intel Core i7 processor or equivalent, with 16GB of RAM. The software used includes Python with libraries such as Pandas, NumPy, Scikit-Learn, CatBoost, and LightGBM. For data processing and visualization, Jupyter Notebook is used for modeling experiments, along with Matplotlib and Seaborn for visualizing prediction results. This research consists of four main stages: data pre-processing, data preparation, data modeling and evaluation, and interpretation of result, as shown in Fig. 1.

A. Data Pre-processing

In the data pre-processing stage, any records with missing values are eliminated to maintain data quality Next, the variable *nominal_honor* (honorarium amount) is converted to a numeric format to ensure all data is in numeric form.

B. Data Preparation

In the data preparation stage, remove unnecessary columns or variables. The next step is to handle class imbalance by splitting the data into an 80% training set and a 20% testing set using the Stratified Splitting method, ensuring that the target distribution remains proportional in both the training and testing sets. Specifically for the LightGBM model, the parameter is_unbalance=True is used. Finally, for handling categorical features, CatBoost uses a list of categories in the cat_features parameter, LightGBM converts them to the category type, and then Logistic Regression, Random Forest, and Neural Network use one-hot encoding.

C. Data Modeling and Evaluation

Data modeling uses five main regression algorithms, namely CatBoostRegressor, LightGBMRegressor, RandomForestRegressor, Logistic Regression as Linear Regression, and Neural Network. The modeling process begins with model instantiation, where each model is assigned hyperparameters that have been adjusted based on literature references and initial experimental observations (manual tuning), without using automated approaches such as grid search or random search. LightGBM uses the parameter objective, metric, learning_rate, max_depth, is_unbalance, verbosity, and seed. Then Catboost uses the parameters iterations, learning_rate, depth, loss_function, cat_features, verbose, and random_state. Random forest uses the parameters n_estimators, max_depth, random_state, n_jobs. Neural network uses the hidden layers parameter with ReLU activation. Finally, Linear Regression does not use additional parameters.

Each model is trained using the training set determined during the data splitting process. Model performance validation is conducted using K-Fold Cross Validation with five folds (K=5), but it is only applied to the Random Forest and Linear Regression models. The selection of these two models is based on computational efficiency, considering their relatively low complexity, which allows for repeated training without significant system load. Cross-validation is used to obtain stable performance estimates and reduce the potential bias due to non-representative data splitting. The evaluation results are visualized through learning curves across folds.

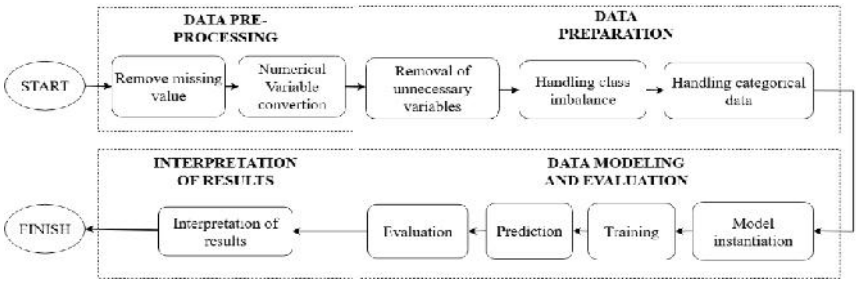


Fig. 1 Flowchart research methodology

On the other hand, the LightGBM, CatBoost, and Neural Network models do not implement K-Fold Cross Validation due to their high complexity and significant resource requirements. The application of cross-validation on these models is considered inefficient for large-scale datasets (58,332 entries) because it potentially increases training time significantly. Therefore, the evaluation of the three models was conducted using a single data split (hold-out split), which is considered more practical in the context of this experiment.

After the training process is complete, each model is used to make predictions on the test data to evaluate its performance in projecting the honorarium value. Model evaluation is conducted using Mean Squared Error (MSE) to measure the average squared error, Root Mean Squared Error (RMSE) to interpret the error in the same unit as the target, Mean Absolute Error (MAE) to calculate the average absolute error, and R-Squared (R^2) to assess how well the model explains data variability. Evaluation of the computational performance of each model is conducted based on four main indicators: execution time, memory usage, and CPU usage time by the user and the system.

D. *Interpretation of Results*

The final stage is the analysis and comparison of models, where an evaluation of the strengths and weaknesses of each model is conducted based on the calculated regression metrics. This stage will also evaluate resource usage in the computation process.

III. RESULT AND DISCUSSION

A. *Dataset*

The dataset consists of 13 input features and 1 target variable in the form of a *nominal_honor* (honorarium amount). The input features include attributes such as *pegawai_id* (employe ID), *nik* (employee identification number), *nama* (name), *status_pegawai* (employee status), *eselon* (echelon), *Pendidikan* (education), *jabatan_akademik* (academic title), *pangkat* (employee rank), *keterangan* (description), *kesulitan* (difficulty), *tugas* (task), *tgl_diterima* (date received), and *kategori_honor* (category honorarium). Table II presents a data dictionary that lists each feature in the dataset along with a brief description and the type of data for each, thereby providing an understanding of the data structure used.

Table III presents a sample excerpt from the honorarium dataset used. The sample displays several entries with the original values of each feature.

The dataset used has high complexity, with a wide variation in honorarium values, ranging from Rp26,040 to Rp7,950,000. The average is Rp677,914.87, but the median is only Rp300,000, indicating a right-skewed distribution, where most values are below the average. The high standard deviation (Rp984,360.53) also indicates the presence of outliers and significant data variability.

The data distribution used in this study is highly imbalanced, as shown in Fig. 2. As much as 67.9% of the data falls into the very small category, and 25.2% into the small category, while the medium, large, and very large categories collectively account for less than 7% of the total data. This imbalance has the potential to introduce bias toward the majority class, necessitating a predictive model capable of handling data with many categories and an uneven distribution, such as CatBoost and LightGBM.

B. *Data Pre-Processing*

In the data pre-processing stage, all rows containing missing values were removed to maintain data quality. Subsequently, the *nominal_honor* (honorarium amount) variable was converted to a numerical format so that all data was in a consistent numerical form. A total of 215 lines of data were deleted, with most originating from the *eselon* (echelon) variable, which initially had empty values.

C. *Data Preparation*

In the data preparation stage, the processes carried out include the removal of unnecessary variables in modeling, handling categorical data, and addressing class imbalance.

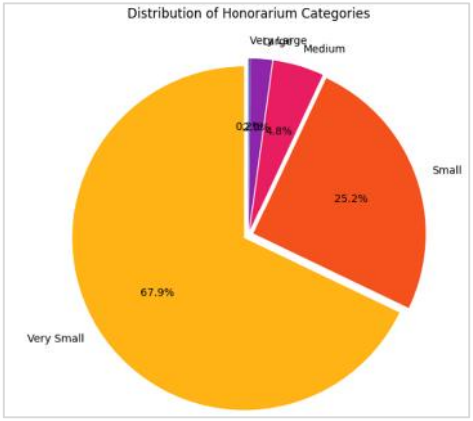


Fig. 2 Distribution of honorarium categories

TABLE II
EMPLOYEE HONORARIUM DATA DICTIONARY

No	Feature Name	Description	Data Type
1	*pegawai_id* (employe ID)	Unique internal system identity number	Numerik
2	*nik* (employee identification number)	Official employee identification number (NIK)	Numerik
3	*nama* (name)	Full name of the employee	Numerik
4	*status_pegawai* (employee status)	Employment status (Permanent, Contract, Lecturer DPK, Foundation Employee etc.)	Categorical
5	*eselon* (echelon)	Structural position level (e.g., 1a, 1b, 2a, etc)	Ordinal
6	*pendidikan* (education)	Highest level of education (Bachelor's, Master's, Doctorate)	Ordinal
7	*jabatan_akademik* (academic title)	Academic title (Assistant Professor, Lecturer, etc.)	Ordinal
8	*pangkat* (employee rank)	Employee rank (III/a, III/b, etc.)	Ordinal
9	*keterangan* (description)	Type of activity for which the honorarium is given	Categorical
10	*kesulitan* (difficulty)	Difficulty level of the activity (very easy, easy, difficult, very difficult)	Ordinal
11	*tugas* (task)	Position in the committee (Chairperson, Member, etc.)	Categorical
12	*tgl_diterima* (date received)	Date of the honorarium payment transaction	Date
13	*kategori_honor* (category honorarium)	Category of honorarium amount (Very large, large, medium, small, very small)	Ordinal
14	*nominal_honor* (honorarium amount)	Amount of honorarium in rupiah units	Numerik (Target)

1) Removal of unnecessary variables: remove the variables '*pegawai_id* (employee ID)', '*tgl_diterima* (date received)', '*keterangan* (description)', '*nik* (employee identification number)', and '*nama* (name)' from the training and testing data. After the removal, the dataset now has 8 variable columns consisting of *status_pegawai* (employee status), *eselon* (echelon), *pendidikan* (education), *jabatan_akademik* (academic title), *pangkat* (employee rank), *kesulitan* (difficulty), *tugas* (task), *kategori_honor* (category honorarium) and 1 target variable, which is *nominal_honor* (honorarium amount).

2) Handling class imbalance: the dataset was divided using Stratified Splitting with an 80:20 ratio, resulting in approximately 46,665 data for the training set and 11,667 data for the testing set from a total of 58,332 entries. This division maintains the proportion of target categories evenly in both sets, with the "very small" category dominating (67.8%), followed by "small" (25.1%), "medium" (4.7%), "large" (2.01%), and "very large" (0.2%). This technique is important to maintain the validity of model evaluation and prevent bias. In LightGBM, the parameter is_unbalance=True is used to automatically handle class imbalance. This feature allows the model to give more weight to the minority class based on the label distribution in the training data, thereby increasing sensitivity to the rarely occurring class [22].

3) Handling categorical data: In data processing for machine learning, categorical data needs to be converted into a numerical format so that it can be processed by the algorithm [15]. This study uses the One-Hot Encoding technique before being processed by the Logistic Regression, Random Forest, and Neural Network models.

TABLE III
DATASET HONORARIUM

ID	*nik* (employee identification number)	*nama* (name)	*status_pegawai* (employee status)	*keterangan* (description)	…	*kesulitan* (difficult)	*kategori_honor* (category honorarium)	*nominal_honor* (honorarium amount)
1	196503091994031002	[disguised]	Lecturer DPK	Monthly Honor July, August, September	…	easy	small	1.500.000
2	196503091994031002	[disguised]	Lecturer DPK	Honor Monitoring dan Evaluasi Internal	…	very easy	very small	250.000
…	….	…	…	…	…	…	…	…
58.332	2160687	[disguised]	Foundation Employee	Monthly Honor July, August, September	…	easy	small	1.200.000

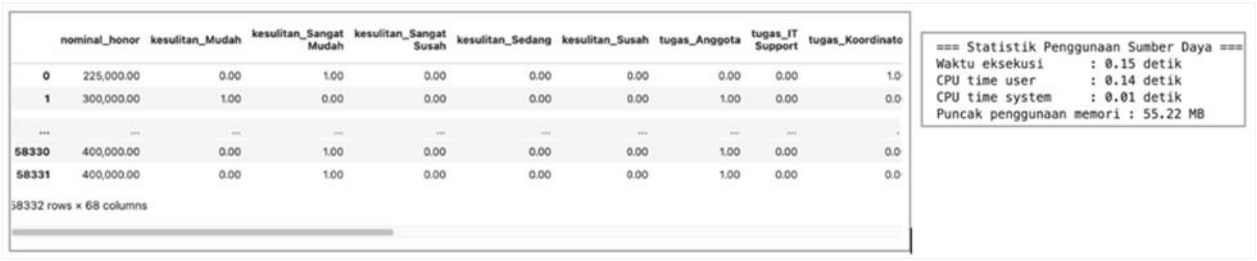
The impact of the categorical data transformation process on computational resources can be seen in Fig. 3, which displays memory usage and the results of the One-Hot Encoding process. Before encoding, there were 8 categorical variables: *status_pegawai* (employee status), *eselon* (echelon), *pendidikan* (education), *jabatan_akademik* (academic title), *pangkat* (employee rank), *kesulitan* (difficulty), *tugas* (task), *kategori_honor* (category honorarium), as well as 1 numerical variable: *honor_nominal* (honorarium amount). After undergoing the One-Hot Encoding process, the number of columns increased to 68, as each unique category was converted into its own binary column. This increase in dimensions impacts the memory and computational requirements of the model. In testing, the One-Hot Encoding process caused a spike in memory usage up to 55.22 MB, although it lasted for a short period. This is due to the large number of high-dimensional numerical arrays generated from the transformation of categorical data.

On the other hand, CatBoost and LightGBM do not require One-Hot Encoding because they can handle categorical data directly. CatBoost uses Ordered Target Encoding, which takes into account the relationship with the target during training, while LightGBM uses a histogram approach and GOSS technique to efficiently process categorical features without explicit conversion [16], as shown in Fig. 4, which illustrates the results of categorical data encoding in both models.

D. Data Modeling and Evaluation

This study uses five main regression algorithms, namely CatBoostRegressor, LightGBMRegressor, RandomForestRegressor, Linear Regression (Logistic Regression used as an approach), and Neural Network. Before the modeling process, the data is separated into features (independent variables) and target (dependent variable). The predicted target is the "*nominal_honor* (honorarium amount)" column. The "*kategori_honor* (category honorarium)" and "*nominal_honor* (honorarium amount)" columns are removed from X_train and X_test because "*kategori_honor* (category honorarium)" is only used in the stratification process, and "*nominal_honor* (honorarium amount)" as the target should not be included in the input features. The values from the "*nominal_honor* (honorarium amount)" column are then stored as y_train and y_test. The final result is the X_train/y_train pairs for training, and X_test/y_test for evaluation, arranged so that the target does not mix into the features during model training.

1) Model instantiation: each model is given hyperparameters that have been adjusted based on literature references and initial experimental observations (manual tuning), without using automatic approaches such as grid search or random search. The LightGBM model uses settings such as objective='regression', metric='rmse', learning_rate=0.1, max_depth=6, and is_unbalance=True to handle class imbalance, and seed to ensure consistent results. The CatBoost model is configured with iterations=450, learning_rate, depth=5, loss_function='RMSE', and cat_features to handle categorical features and random_state=42 to ensure reproducibility. In the Neural Network model, the architecture is built using the Sequential model with Dense layers and ReLU activation, as well as an input_shape adjusted to the data dimensions. Meanwhile, Linear Regression is directly instantiated with LinearRegression() to model the linear relationship between features and the target. As for Random Forest, it uses n_estimators=100, max_depth=7, and random_state=42 to produce stable results. All the parameters used are designed to match the characteristics of the data and enhance the effectiveness of the predictions.



	nominal_honor	kesulitan_Mudah	kesulitan_Sangat Mudah	kesulitan_Sangat Susah	kesulitan_Sedang	kesulitan_Susah	tugas_Anggota	tugas_IT Support	tugas_Koordinato
0	225,000.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.0
1	300,000.00	1.00	0.00	0.00	0.00	0.00	1.00	0.00	0.0
...	...	...	...	...	...	...	...	...	.
58330	400,000.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.0
58331	400,000.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00	0.0

58332 rows × 68 columns

```
=== Statistik Penggunaan Sumber Daya ===
Waktu eksekusi      : 0.15 detik
CPU time user       : 0.14 detik
CPU time system     : 0.01 detik
Puncak penggunaan memori : 55.22 MB
```

Fig. 3 Resource usage and results of the one-hot encoding process

```
15778          Mudah
57650    Sangat Mudah
34919          Mudah
43279          Susah
49262          Susah
          ...
45197          Mudah
31404    Sangat Mudah
39964          Susah
48118    Sangat Mudah
21676          Mudah
Name: kesulitan, Length: 46665, dtype: category
Categories (5, object): ['Mudah', 'Sangat Mudah', 'Sangat Susah', 'Sedang', 'Susah']
```

Fig. 4 Results of categorical data encoding in CatBoost and LightGBM

2) Training: the training process is carried out by each model based on the previously determined model instantiation parameters. In the LightGBM, Catboost, and Neural Network models, the number of iterations is determined, and early stopping is applied, which is used to halt the model training early if there is no significant improvement in the model's performance after several consecutive iterations.

Fig. 5 presents a comparison of the learning curves of each model based on error metrics, such as RMSE, MSE, and loss. This visualization is used to evaluate the training stability and convergence of the model, as well as to identify potential overfitting or underfitting during the training process. In the LightGBM model (Fig. 5a), the RMSE on the validation data was initially very high, but it experienced a sharp decline in the first 20–30 iterations and began to stabilize at the 100th iteration, with the best RMSE being 663.644. This indicates that the model has achieved optimal performance through the early stopping mechanism, without signs of overfitting. The CatBoost model (Fig. 5b) shows a similar trend, with a significant decrease in RMSE until it stabilizes at iteration 287 (RMSE: 665.588), indicating good convergence. For Linear Regression (Fig. 5c), the learning curve shows overfitting when the amount of training data is still small, marked by high MSE values. However, as the amount of data increases, the MSE values on the training and testing data decrease and tend to stabilize, reflecting the model's generalization capability. The shaded area around the line illustrates higher model uncertainty with smaller data sizes. The Random Forest model (Fig. 5d) was evaluated with five-fold cross-validation and produced relatively stable MSE values (around 4.5–4.7), although there was a spike in the fourth fold indicating potential overfitting or the presence of outliers in that subset of data. Meanwhile, the Neural Network (Fig. 5e) showed a very rapid decrease in loss at the beginning of the training, then stabilized around the value of 0.5. This indicates that the model has reached the convergence point and is effectively learning from the data. Overall, this graph provides a clear picture of the convergence speed, performance stability, and potential overfitting of each model during the training process, while also demonstrating the effectiveness of the model in reducing prediction errors.

3) Prediction: after the training process is complete, predictions are made by each model. Fig. 6 shows the comparison between the actual and predicted values of each model. Each graph provides an overview of how each model predicts values based on the existing actual data. The LightGBM model (Fig. 6a) shows the best performance, with prediction points closest to the ideal line, reflecting high accuracy even on complex or unbalanced data. CatBoost (Fig. 6b) is also accurate, although it tends to produce conservative predictions at low values and is slightly more dispersed compared to LightGBM, yet it still excels in handling categorical features. Linear Regression (Fig. 6c) shows the worst performance, with prediction points scattered far from the ideal line, indicating a mismatch with the non-linear data pattern. On the other hand, Random Forest (Fig. 6d) and Neural Network (Fig. 6e) provide better predictions, although with greater deviations compared to the boosting model. Random Forest is quite stable due to its ensemble nature, while Neural Network shows potential but is still suboptimal, especially at low values. Overall, LightGBM and CatBoost are the most accurate models, followed by Random Forest and Neural Network, while Linear Regression is less suitable for complex and non-linear datasets.

4) Evaluation: model evaluation is conducted using four main metrics, namely MSE, RMSE, MAE, and R^2, as shown in Table IV, to assess the prediction error rate and the model's ability to explain data variability. In addition, computational efficiency is also evaluated thru the measurement of execution time, memory usage, and CPU time (user and system). This comprehensive evaluation aims to assess not only accuracy but also the efficiency of resource usage during the model training process.

MSE measures the mean squared difference between actual values and predicted values. The smaller the MSE value, the better the model's performance in minimizing large errors. Based on the evaluation results, the LightGBM model has the lowest MSE value of 443,503,237,551.86, indicating the best performance in reducing squared errors. Conversely, Linear Regression recorded the highest MSE value of 487,196,432,706.98, indicating a significant prediction error.

MAE measures the average absolute prediction error without considering the direction of the error. The LightGBM model again shows the best performance with an MAE of 344,049.91, indicating that on average, the model's predictions deviate by approximately this amount from the actual values. On the other hand, Linear Regression has the highest MAE value at 380,498.59, indicating that this model produces the largest absolute prediction deviation compared to other models.

RMSE is the square root of MSE and gives greater weight to larger errors. The smallest RMSE value was obtained by LightGBM with 665,960.39, followed by CatBoost and Random Forest Regression. This reinforces the conclusion that LightGBM is more stable and precise in generating predictions compared to other models. The largest RMSE is held by Linear Regression and Neural Network, indicating that these two models are more vulnerable to outliers or large errors.

The R^2 value indicates the proportion of data variability that can be explained by the model. A value close to 1 signifies a very good model. In this experiment, LightGBM again emerged as the model with the highest R^2 value of 0.54, meaning this model can explain about 54% of the variability in the target data. Conversely, Linear Regression only explains 49% of the data variability, making it the model with the lowest explanatory power among all the models tested.

To evaluate the computational efficiency of each model, measurements were taken on four main aspects: execution time, memory usage, user-level CPU time, and system-level CPU time. Detailed information regarding the computational resource usage of each model is presented in Table V.

The model with the fastest execution time is Linear Regression (0.27 seconds), followed by Random Forest (1.36 seconds) and LightGBM (1.97 seconds). However, the Neural Network takes a very long time, namely 123.69 seconds, which indicates that this model is much more complex and requires a longer training process.

Linear Regression actually uses the most memory (100.12 MB), followed by Neural Network (106.52 MB) and Random Forest (77.98 MB). In contrast, LightGBM only uses 2.67 MB, making it the most memory-efficient model, much lighter than [24][25] with Random Forest and Adaboost models that require around 200-700 MB, SVM using 95-114 MB, and Deep Learning using 333-339 MB.

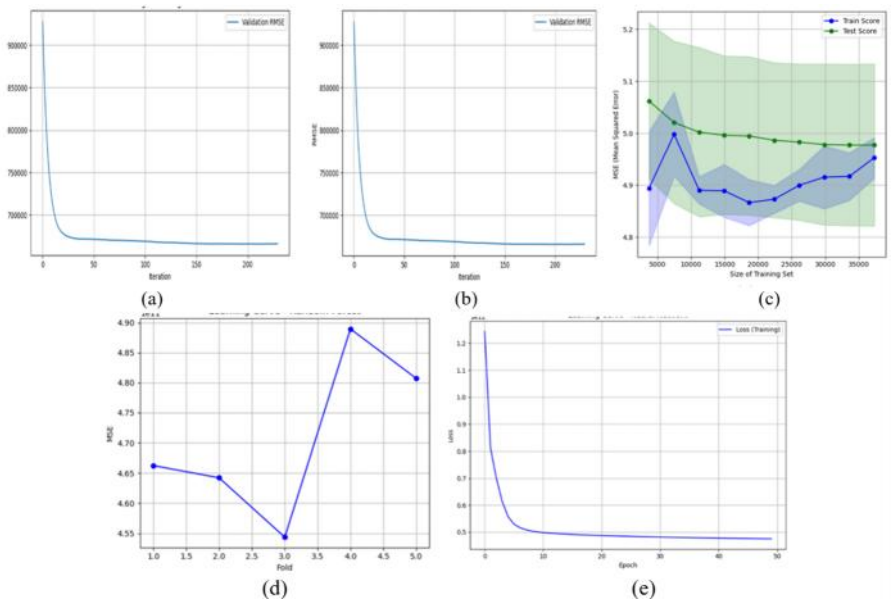


Fig. 5 Learning curve of: (a) LightGBM; (b) Catboost; (c) Linear Regression; (d) Random Forest; (e) Neural Network

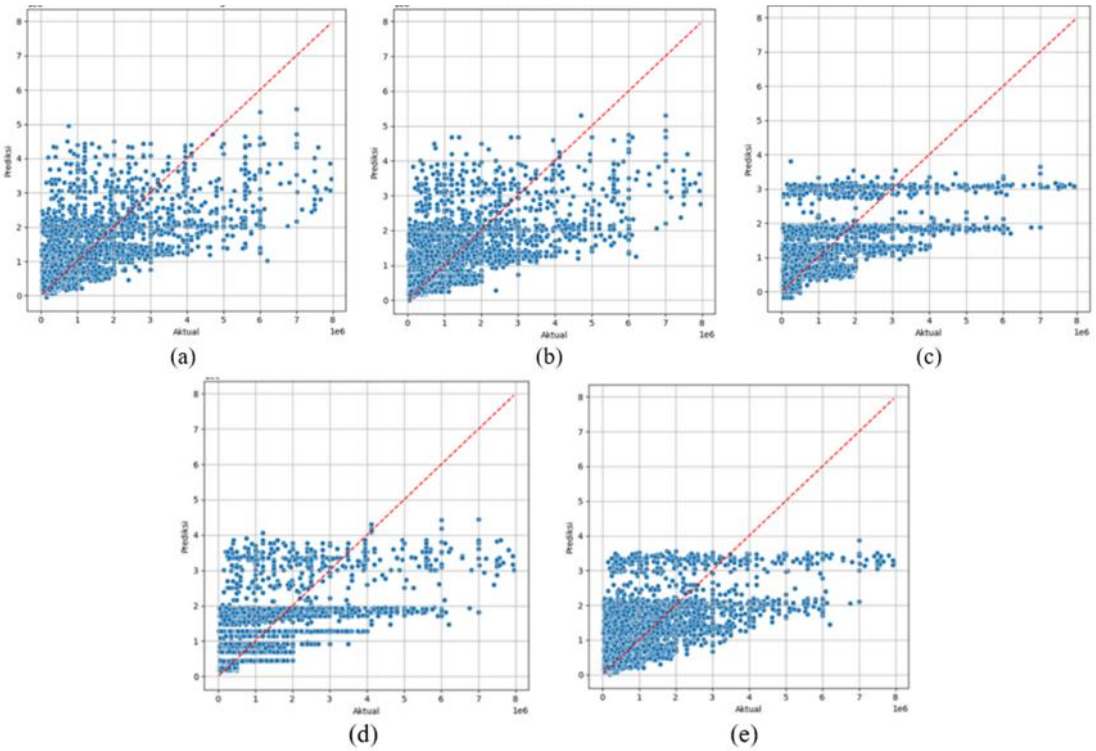


Fig. 6 Actual and predicted values of: (a) LightGBM; (b) Catboost; (c) Linear Regression; (d) Random Forest; (e) Neural Network

CPU Time User reflects the processing time performed by the CPU on behalf of the user (main process). Neural Network recorded the highest figure with 133.02 seconds, indicating intensive computational use. LightGBM is much more efficient at just 1.04 seconds, and Linear Regression is even lighter at 0.58 seconds.

This method measures the time spent by the CPU on the system process (kernel-level). The highest values are also found in Neural Network (7.61 seconds) and CatBoost (1.90 seconds), while Linear Regression and Random Forest show minimal system time.

E. INTERPRETATION OF RESULTS

The evaluation results show that LightGBM is the most optimal model overall. This model excels in all regression metrics (MSE, MAE, RMSE, R^2) and is the most efficient in terms of time and memory usage. The performance advantage of LightGBM is closely tied to one of its main features, namely Gradient-based One-Side Sampling (GOSS), which allows the model to focus more on data with high prediction errors while reducing the computational burden by ignoring some less influential data. Additionally, the use of a leaf-wise tree growth strategy differs from the traditional level-wise approach, allowing the model to build deeper and more adaptive structures to complex data patterns. Although this approach has the potential for overfitting on small datasets, in large datasets like in this study, it actually provides a significant performance improvement.

CatBoost also demonstrates competitive performance, approaching LightGBM. With the ordered target encoding technique, CatBoost handles categorical features directly without increasing data dimensions, making it more memory-efficient compared to other models like Linear Regression and Neural Network, which use one-hot encoding and experience memory bloat of over 100 MB. Random Forest and Neural Network show fairly good accuracy but are resource-intensive due to their reliance on one-hot encoding. Neural Network recorded the longest training time (123 seconds) and the highest memory usage. Linear Regression, although computationally fast, provides the lowest accuracy because it cannot capture the non-linear complexity in the data and is also affected by the increase in dimensions due to one-hot encoding.

TABLE IV
MODEL EVALUATION RESULTS

Model	MSE	MAE	RMSE	R^2
LightGBM	443,503,237,551.86	344,049.91	665,960.39	0.54
CatBoost	445,416,111,571.58	346,383.98	667,395.02	0.53
Linear Regression	487,196,432,706.98	380,498.59	697,994.58	0.49
Random Forest Regression	459,187,958,437.92	353,549.47	677,634.09	0.52
Neural Network	464,307,168,013.16	360,587.51	681,400.89	0.51

TABLE V
USE OF COMPUTING RESOURCES

Model	Execution Time (s)	Memory Usage (MB)	CPU Time (User)	CPU Time (System)
LightGBM	1.97	2.67	1.04	0.83
CatBoost	4.25	7.49	20.67	1.90
Linear Regression	0.27	100.12	0.58	0.06
Random Forest Regression	1.36	77.98	5.42	0.20
Neural Network	123.69	106.52	133.02	7.61

The high memory usage of models like Neural Networks and Linear Regression has important implications for real-world applications, especially in systems with resource constraints such as institutional local servers or web-based applications. The large memory load can reduce system performance, slow down the prediction process, and increase the risk of memory errors (out-of-memory). This also impacts cost efficiency, especially if the system is hosted on a cloud service that charges based on resource usage.

In addition, there is an important trade-off between accuracy and computational efficiency. Models like LightGBM offer an optimal balance, where high accuracy is achieved with minimal memory and time consumption. On the other hand, models with higher accuracy potential such as Neural Networks require significantly more computation, making them less suitable for scenarios that demand speed and efficiency.

Overall, the CatBoost and LightGBM models have proven to be superior compared to other models, both in terms of accuracy, resource efficiency, and the ability to handle categorical features without the need for additional encoding. This advantage makes both models very ideal for application in complex and imbalanced real data-based honorarium prediction scenarios, such as in a university environment.

However, the high predictive performance of boosting models like LightGBM and CatBoost is often offset by a lower level of interpretability compared to simpler models like linear regression. In the context of real-world applications, particularly in human resource management (HRM) systems, interpretability is an important aspect that allows end users to understand the reasons behind each prediction or recommendation of the system. When the system is used to determine the honorarium amount of employees, the decisions generated by the model must be transparently justified to policymakers.

Therefore, there is a trade-off that needs to be considered between prediction accuracy and model interpretability. To bridge the need for accuracy and transparency, approaches such as feature importance analysis or model interpretability methods like SHAP (SHapley Additive exPlanations) can be used to identify the contribution of each feature to the prediction results. Thus, complex models can still be used in predictive systems based on real data without sacrificing aspects of transparency and accountability in practical decision-making.

IV. CONCLUSION

This study comprehensively evaluates the performance of five regression algorithms, focusing on CatBoost and LightGBM, in predicting honorarium amounts based on an actual dataset that is categorical and imbalanced. The results show that LightGBM is the most optimal model, with an RMSE value of 665.960 and an R^2 of 0.54, as well as high computational efficiency (memory usage of only 2.67 MB). This model is capable of handling categorical features without one-hot encoding, which contributes to memory savings and execution time. CatBoost also demonstrates competitive predictive performance, especially in processing categorical data. Meanwhile, Random Forest and Neural Network show fairly good accuracy but with greater resource consumption, while Linear Regression cannot handle data complexity, although it excels in training

speed. This study has several limitations. The evaluation was conducted in a laboratory environment using a static dataset, thus it does not yet reflect the actual information system conditions. Additionally, the testing has not considered real-time data distribution changes, nor has it integrated with the actual human resource management system. The aspect of model interpretability has also not been discussed in depth, even though transparency is important in practical decision-making in educational institutions. Further research is recommended to implement this predictive model on an information system directly connected to an active database, in order to approach a more realistic operational scenario. In addition, it is necessary to add a mechanism for periodic data updates, as well as to explore model interpretability methods such as SHAP to enhance decision transparency. Integration with information retrieval methods can also expand the system's scope, thereby providing more contextual, safe, and adaptive honorarium recommendations in response to data dynamics.

REFERENCES

[1] R. Kablaoui and A. Salman, "Machine Learning Models for Salary Prediction Dataset using Python," in 2022 International Conference on Electrical and Computing Technologies and Applications, ICECTA 2022, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 143–147. doi: 10.1109/ICECTA57148.2022.9990316.

[2] J. Y. Kuo, H. C. Lin, and C. H. Liu, "Building graduate salary grading prediction model based on deep learning," Intell. Autom. Soft Comput., vol. 27, no. 1, pp. 53–68, 2021, doi: 10.32604/iasc.2021.014437.

[3] A. Asaduzzaman, M. R. Uddin, Y. Woldeyes, and F. N. Sibai, "A Novel Salary Prediction System Using Machine Learning Techniques," in Proceedings - 2024 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering, ECTI DAMT and NCON 2024, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 38–43. doi: 10.1109/ECTIDAMTNCON60518.2024.10480058.

[4] D. A. Gomez-Cravioto, R. E. Diaz-Ramos, N. Hernandez-Gress, J. L. Preciado, and H. G. Ceballos, "Supervised machine learning predictive analytics for alumni income," J. Big Data, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40537-022-00559-6.

[5] F. Li, N. A. Majid, and S. Ding, "Unlocking the potential of LSTM for accurate salary prediction with MLE, Jeffreys prior, and advanced risk functions," PeerJ Comput. Sci., vol. 10, 2024, doi: 10.7717/peerj-cs.1875.

[6] G. Ramaswami, T. Susnjak, and A. Mathrani, "On Developing Generic Models for Predicting Student Outcomes in Educational Data Mining," Big Data Cogn. Comput., vol. 6, no. 1, Mar. 2022, doi: 10.3390/bdcc6010006.

[7] D. A. Devyatkin and O. G. Grigoriev, "Random Kernel Forests," IEEE Access, vol. 10, pp. 77962–77979, 2022, doi: 10.1109/ACCESS.2022.3193385.

[8] J. Bilski, L. Rutkowski, J. Smoląg, and D. Tao, "A novel method for speed training acceleration of recurrent neural networks," Inf. Sci. (Ny)., vol. 553, pp. 266–279, Apr. 2021, doi: 10.1016/j.ins.2020.10.025.

[9] S. Li and Y. Yang, "A recurrent neural network framework with an adaptive training strategy for long-time predictive modeling of nonlinear dynamical systems," J. Sound Vib., vol. 506, Aug. 2021, doi: 10.1016/j.jsv.2021.116167.

[10] B. Amirshahi and S. Lahmiri, "Bankruptcy prediction using optimal ensemble models under balanced and imbalanced data," Expert Syst., vol. 41, no. 8, Aug. 2024, doi: 10.1111/exsy.13599.

[11] R. Oktafiani, A. Hermawan, and D. Avianto, "Pengaruh Komposisi Split data Terhadap Performa Klasifikasi Penyakit Kanker Payudara Menggunakan Algoritma Machine Learning," J. Sains dan Inform., pp. 19–28, Jun. 2023, doi: 10.34128/jsi.v9i1.622.

[12] E. Erlin, Y. Desnelita, N. Nasution, L. Suryati, and F. Zoromi, "Dampak SMOTE terhadap Kinerja Random Forest Classifier berdasarkan Data Tidak seimbang," MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput., vol. 21, no. 3, pp. 677–690, Jul. 2022, doi: 10.30812/matrik.v21i3.1726.

[13] L. H. Li, R. Ahmad, R. Tanone, and A. K. Sharma, "STB: synthetic minority oversampling technique for tree-boosting models for imbalanced datasets of intrusion detection systems," PeerJ Comput. Sci., vol. 9, 2023, doi: 10.7717/peerj-cs.1580.

[14] J. Y. Chiang, Y. Lio, C. Y. Hsu, C. L. Ho, and T. R. Tsai, "Binary Classification with Imbalanced Data," Entropy, vol. 26, no. 1, Jan. 2024, doi: 10.3390/e26010015.

[15] C. Herdian, A. Kamila, and I. G. Agung Musa Budidarma, "Studi Kasus Feature Engineering Untuk Data Teks: Perbandingan Label Encoding dan One-Hot Encoding Pada Metode Linear Regresi," Technol. J. Ilm., vol. 15, no. 1, p. 93, Jan. 2024, doi: 10.31602/tji.v15i1.13457.

[16] P. Septiana Rizky, R. Haiban Hirzi, U. Hidayaturrohman, U. Hamzanwadi, "Perbandingan Metode LightGBM dan XGBoost dalam Menangani Data dengan Kelas Tidak Seimbang," J Statistika, vol. 15, no. 2, pp. 228-236, 2022, doi: 10.36456/jstat.vol15.no2.a5548.

[17] Q. Chen, J. Li, J. Feng, and J. Qian, "Dynamic comprehensive quality assessment of postharvest grape in different transportation chains using SAHP–CatBoost machine learning," Food Qual. Saf., vol. 8, 2024, doi: 10.1093/fqsafe/fyae007.

[18] J. Wu, Q. Li, Q. Chen, N. Zhang, C. Mao, L. Yang, and J. Wang, "Fault diagnosis of the HVDC system based on the CatBoost algorithm using knowledge graphs," Frontiers in Energy Research, vol. 11, 2023, doi: 10.3389/fenrg.2023.1144785.

[19] V. Kumar, N. Kedam, K. V. Sharma, D. J. Mehta, and T. Caloiero, "Advanced Machine Learning Techniques to Improve Hydrological Prediction: A Comparative Analysis of Streamflow Prediction Models," Water (Switzerland), vol. 15, no. 14, Jul. 2023, doi: 10.3390/w15142572.

[20] Y. Zhang, J. Ma, S. Liang, X. Li, and M. Li, "An evaluation of eight machine learning regression algorithms for forest aboveground biomass estimation from multiple satellite data products," Remote Sens., vol. 12, no. 24, pp. 1–26, Dec. 2020, doi: 10.3390/rs12244015.

[21] L. Lin, J. Zhang, N. Zhang, J. Shi, and C. Chen, "Optimized LightGBM Power Fingerprint Identification Based on Entropy Features," Entropy, vol. 24, no. 11, Nov. 2022, doi: 10.3390/e24111558.

[22] X. Sun, M. Liu, and Z. Sima, "A novel cryptocurrency price trend forecasting model based on LightGBM," Financ. Res. Lett., vol. 32, Jan. 2020, doi: 10.1016/j.frl.2018.12.032.

[23] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," Institute of Electrical and Electronics Engineers Inc, vol. 10, pp. 99129-99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[24] Y. Y. Li, T. Van Do, and H. T. Nguyen, "A comparison of forecasting models for the resource usage of MapReduce applications," Neurocomputing, vol. 418, pp. 36–55, 2020, doi: 10.1016/j.neucom.2020.07.059.

[25] D. Preuveneers, I. Tsingenopoulos, and W. Joosen, "Resource usage and performance trade-offs for machine learning models in smart environments," Sensors (Switzerland), vol. 20, no. 4, 2020, doi: 10.3390/s20041176.