Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Digital Transformation of Electricity Bill Collection: Predicting Delays using Machine Learning

Dyah Puspita Sari Nilam Utami[1)*], Mochamad Ikbal Arifyanto[2)]
[1)2)] Magister Sains Komputasi, Institut Teknologi Bandung, Indonesia
[1)] 20922327@mahasiswa.itb.ac.id, [2)] ikbal@as.itb.ac.id

Submitted :Dec 14, 2024 | **Accepted** : Dec 25, 2024 | **Published** : Jan 17, 2025

Abstract: Delays in electricity bill payments pose a significant challenge for PLN in maintaining financial stability and delivering equitable service quality to the public. This study aims to develop a payment delay prediction system to assist PLN UP3 Makassar Utara in prioritizing invoice distribution to customers with a high likelihood of late payments. The Random Forest algorithm was chosen for its ability to handle complex data and produce reliable predictions. This research analyses historical electricity customer data from 2018 to 2023, encompassing 227,163 entries. The data is processed using validation techniques such as K-Fold Validation and Rolling Window Validation to ensure the accuracy and generalizability of the model. The study's findings demonstrate that an accurate payment delay prediction model can be developed using the Random Forest method, incorporating historical features such as lag features, moving averages, and seasonal variables. Additionally, the system prioritizes invoice delivery to high-risk customers based on risk scores derived from historical delay patterns. This system reduces payment arrears at PLN UP3 Makassar Utara through proactive measures such as early notifications, personalized reminders, or payment incentives to encourage timely payments. As a result, the study indicates that the system effectively enhances the efficiency of payment management and supports the company's financial stability. However, the research is limited by the use of data from a single region, the absence of external factors in the model, and the high computational requirements. For broader implementation, further research should include data from other regions, consider external factors, and optimize computational resource usage.

Keywords: electricity bill payment; payment delay; payment management; payment prediction; random forest.

INTRODUCTION

Electricity is fundamental to modern society, from lighting to operating essential electronic devices daily. Moreover, electricity is a cornerstone for economic and industrial development across various sectors (Ibekwe et al., 2024; Usman et al., 2024). Ensuring reliable and affordable electricity services is a crucial responsibility for electricity providers, such as Perusahaan Listrik Negara (PLN), which guarantees equal access to electricity for all communities (Daryanti & Sudarwanto, 2024; Maulidia et al., 2019). However, operations and service quality sustainability depend on an efficient and timely electricity bill payment system. Timely bill payments are vital to maintaining the company's financial stability and ensuring smooth public service operations, which are central to PLN's strategic state-owned enterprise (SOE) role (Negara & Ramayandi, 2020). PLN UP3 Makassar Utara faces significant challenges in managing increasing electricity payment arrears. A notable issue is the high number of customers needing to pay their electricity bills on time, surpassing the due date of the 20th of each month. As of April 2024, total arrears amounted to IDR 11 billion, creating substantial financial pressure that threatens operational continuity and risks degrading customer service quality. The leading causes of this problem include customers' need for payment discipline and inefficiencies in the invoice delivery system. The current system sends invoices en masse without considering individual customer payment patterns, often resulting in invoices not reaching customers at the appropriate time.

Historical payment data for PLN UP3 Makassar Utara customers from 2018 to 2023 includes customer IDs, bill amounts, payment dates, arrears, and demographic information. Initial analysis revealed recurring late payment patterns, particularly among customers whose payday does not align closely with the payment due date.

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Additionally, arrears trends indicate a concentration of customers consistently paying late every month. This data is valuable for developing a data-driven payment delay prediction system to identify high-risk customers and implement more targeted preventive measures. This study proposes developing a payment delay prediction system using the Random Forest algorithm, a highly effective machine learning method for classification data analysis to address this issue. Random Forest employs ensemble learning, combining multiple decision trees to produce more accurate and reliable predictions (Ampomah et al., 2020; Ghiasi & Zendehboudi, 2021). By training the model on historical customer data, the system can identify late payment patterns and prioritize invoice delivery for customers at high risk of late payment. This approach is expected to reduce arrears while significantly enhancing PLN's operational efficiency.

Random Forest is an ensemble learning-based algorithm that builds and combines multiple decision trees to improve prediction accuracy (Mienye & Sun, 2022; Y. Zhang et al., 2022). It excels at mitigating overfitting, particularly with complex and diverse datasets (Kalusivalingam et al., 2021; Mienye & Sun, 2022). Each tree in the Random Forest is constructed using a random subset of training data, resulting in a more robust model against data variation (X. Zhou et al., 2020). In the context of payment delay prediction, Random Forest can capture non-linear relationships between variables such as bill amounts, payment dates, and customer demographics. Previous studies have demonstrated the effectiveness of this algorithm in identifying risk factors and providing reliable predictions for decision-making (Wang et al., 2022; Y. Zhou et al., 2019). This study aims to develop a payment delay prediction system based on Random Forest to assist PLN UP3 Makassar Utara in identifying customers at high risk of late payment. The objectives are to (1) Develop an accurate payment delay prediction model using historical customer data and the Random Forest method; (2) Prioritize invoice delivery to customers with a high risk of late payment; and (3) Demonstrate how the prediction system can help reduce payment arrears at PLN UP3 Makassar Utara.

This research develops an innovative solution to address electricity bill arrears at PLN UP3 Makassar Utara by building a payment delay prediction system based on the Random Forest algorithm. The system uses customers' historical data to generate an accurate predictive model. Consequently, PLN can more effectively identify customers at high risk of payment delays. Through this approach, PLN can strategically prioritize invoice delivery to high-risk customers, enabling the implementation of more targeted preventive measures. In addition to improving operational efficiency in billing management, this system can significantly reduce arrears, support the company’s financial stability, and enhance service quality to customers. It also provides added value through resource optimization and improved relationships between the company and its customers.

LITERATURE REVIEW

Random Forest is a robust ensemble learning algorithm renowned for its accuracy, scalability, and ability to handle complex datasets with minimal parameter adjustments (Asselman et al., 2023; Boutahir et al., 2024; Mohammed & Kora, 2023). The algorithm operates by building multiple decision trees during training and combining their outputs for classification or regression tasks, resulting in stable predictions while reducing the risk of overfitting (Liu et al., 2020; Probst et al., 2019). Using bootstrap aggregating (bagging) and random feature selection, Random Forest effectively addresses overfitting issues commonly encountered with single decision tree models (Salman et al., 2024). It is also flexible in handling non-linear and high-dimensional data, making it highly suitable for datasets with numerous independent variables. Its advantages, including the ability to generate consistent predictions even in the presence of data noise and the capacity to provide variable importance estimations, have made Random Forest widely applicable in fields such as finance, healthcare, and utilities (Kalusivalingam et al., 2020; Ong et al., 2023). However, the algorithm has limitations, including the interpretability of results due to its "black box" nature and the need for parameter tuning, which can be time-consuming and resource-intensive, particularly with large datasets or models involving many trees (Javed et al., 2024; Preuveneers et al., 2020).

Previous studies have demonstrated the extensive application of Random Forest. For example, Schonlau & Zou, (2020) used this algorithm to predict credit card defaults and the popularity of online articles. At the same time, Rahmi et al. (2023) utilized it to forecast delays in JKN-KIS premium payments, aiming to improve payment management efficiency through historical data analysis. Teles et al.'s (2021) research applied Random Forest to credit risk analysis and bankruptcy prediction, highlighting the algorithm’s relevance in processing large datasets for predictive decision-making. Although similar in algorithm use, this study differs significantly from previous research. This research focuses on predicting electricity bill payment delays among PLN UP3 Makassar Utara customers, contrasting with Schonlau & Zou's (2020) study on credit card defaults or Rahmi et al.'s (2023) work in the healthcare sector for JKN-KIS programs. While related to financial risk, Teles et al.'s (2021) study is more relevant to predicting bankruptcy in large corporations rather than individual customers in utility services.

In the current study, the Random Forest algorithm is tailored to the needs of PLN UP3 Makassar Utara to address the growing issue of electricity bill arrears by utilizing customers' historical data to predict payment delays. This differs from previous studies focusing on the financial and healthcare sectors. This research expands the

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

application context of Random Forest to the utilities sector, particularly in electricity services, with the aim of reducing the risk of payment arrears, which directly impacts the company's operational sustainability. This approach highlights the relevance of the algorithm in supporting more strategic and efficient data-driven decision-making in the public service sector.

METHOD

This study was quantitative research using an exploratory-predictive method to analyze historical electricity payment data and develop a prediction model based on machine learning algorithms. The research sample consisted of historical payment data from PLN UP3 Makassar Utara electricity customers between 2018 and 2023. The dataset included 227,163 entries covering customer IDs, bill amounts, payment dates, arrears, and demographic details. The sampling technique was purposive, based on customers with complete and relevant payment histories. The research instrument was a dataset of customer payment records in .csv format obtained from the relevant division at PT. PLN. This data was collected through direct documentation from PLN's customer information system with official authorization.

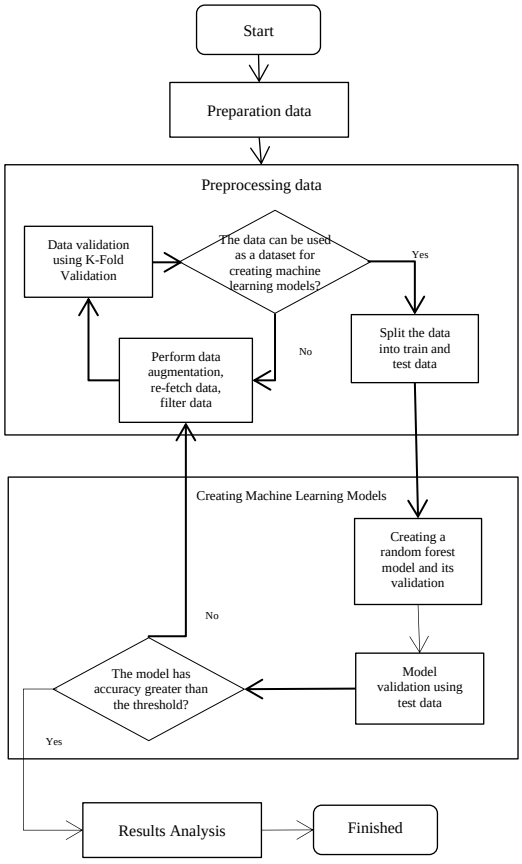

Fig 1. Research Flow

Research Procedure

The analysis of electricity payment delays among PLN customers began with processing a dataset containing 227,163 entries. This data included key information such as customer IDs, bill amounts, payment dates, arrears, and demographic details. During the preprocessing stage, the data was read using Python, where the payment date column was converted into DateTime format and split into attributes such as year, month, date, hour, and minute for more detailed analysis. The data was then normalized, for example, transforming payment dates into a value of 0 for on-time payments. At the same time, late payments were marked with negative values corresponding to the days delayed. To ensure data quality, validation was performed using various techniques. K-fold validation was used to prevent overfitting and ensure model generalization, while Rolling Window Validation was applied to time-series data to maintain temporal order. Cosine Similarity Validation was employed to evaluate the similarity between data points in vector space, ensuring consistent sample patterns. In the model-building stage, the Random Forest algorithm was utilized. The data was split into training and testing sets, and the model was

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

trained using various Random Forest parameters to improve accuracy and reduce the risk of overfitting. The trained model was evaluated using error metrics to measure its performance in predicting payment delays. The evaluation results were then analyzed to identify patterns and trends in payment delays. The best model was selected based on these evaluations, focusing on its ability to accurately predict payment delays. Finally, the model's predictions were compared to determine the most effective method for predicting electricity payment delays.

Preprocessing and Data Exploration

The preprocessing steps in this research included a series of critical stages to ensure the data used was accurate and consistent. The process began with reading the historical electricity payment data from a .csv file using Python, which included customer IDs, bill amounts, payment dates, arrears, and demographic details. Next, the payment date column was converted into Python's DateTime format, which was then split into year, month, date, hour, and minute attributes for more straightforward analysis. Data normalization followed, aligning the scale of data, particularly in the payment date column. In this column, values were converted to 0 for on-time payments (non-delinquent), and negative values (e.g., -5) were used to indicate the number of days the payment was delayed, with larger negative values reflecting longer delinquency durations. Once preprocessing was completed, data exploration was conducted to understand patterns and trends. Initial analysis included identifying arrears trends through bar charts to highlight periods with the highest and lowest arrears, analyzing electricity tariffs frequently associated with arrears using bar or pie charts to identify high-risk customer segments and examining the distribution of payment dates using histograms to observe payment patterns on specific dates. This process provided a comprehensive understanding of the data and guided further analysis.

Data Validation

Data validation was crucial to ensure the prediction model had good generalization capability and avoided overfitting the training data. This research employed three main validation methods: K-Fold Validation, Rolling Window Validation, and Cosine Similarity. K-Fold Validation evaluated the model's overall performance without considering the temporal order, providing an average accuracy score from various data splits. Meanwhile, Rolling Window Validation focused on temporal patterns, making it relevant for time-series data by considering the time dynamics in model validation. Additionally, Cosine Similarity was used for manual validation to analyze pattern similarity between data points in interpreting prediction results and analyzing distribution. These methods complemented each other to ensure the accuracy and reliability of the developed prediction model.

Random Forest Algorithm

The model-building stage utilized the Random Forest algorithm, which offered advantages in handling complex data and preventing overfitting. The process began with data splitting, dividing the data into training and testing sets in multiple variations to evaluate model performance comprehensively. Next, the model was trained using multiple decision trees to enhance accuracy and robustness against outliers. Once the model was trained, the next step was prediction, where the model was used to predict payment delays based on the test data. Finally, model evaluation was conducted using accuracy and error visualization metrics to assess the model's performance in accurately predicting customer payment delays.

Table 1. Parameter Random Forest

Parameter	Description	Value
Number of Trees (n_estimators)	The number of decision trees to be built in the model.	100 to 500
Maximum Depth (max_depth)	The maximum depth of the decision trees to prevent overfitting.	10 to 50, or None
Maximum Features (max_features)	The maximum number of features considered for splitting at each node.	'sqrt'
Splitting Criterion (criterion)	The criterion used to measure the quality of a split..	'Gini'
Minimum Samples for Split (min_samples_split)	The minimum number of samples required to split a node.	2
Minimum Samples for Leaf (min_samples_leaf)	The minimum number of samples required to form a leaf node.	1

Based on Table 1 regarding the Random Forest parameters, the main parameters used in this model encompass several key aspects to ensure accurate and stable predictions. The number of trees (*n_estimators*) is set within the range of 100 to 500, aiming to improve accuracy by leveraging ensemble learning through multiple decision trees. The maximum depth (*max_depth*) is configured between 10 and 50 or left unlimited (*None*), providing flexibility

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

to prevent overfitting in complex data. Additionally, the maximum features (*max_features*) considered at each node split is set to the square root of the total number of features ('*sqrt*'), balancing accuracy and computational efficiency. The split criterion is determined as '*Gini*', which evaluates the quality of splits based on the Gini index. Furthermore, *min_samples_split* and *min_samples_leaf* are set to a minimum of 2 and 1, respectively, ensuring that each node has sufficient data for further splitting and leaves contain an adequate minimum number of data points. This combination of parameters is designed to produce an accurate model capable of capturing non-linear patterns in the historical electricity payment dataset while minimizing the risk of overfitting.

RESULT

Data Preparation

In the data preparation stage, the data used in this study includes information on customer electricity bill payments from 2018 to 2023. This data was imported and processed using various Python libraries, such as Pandas, Numpy, and the Random module. Pandas were employed to read CSV-formatted data, organize it into DataFrames, and perform manipulations like column selection, grouping, and filling in missing values. Numpy was used for advanced mathematical operations, including calculating averages and other statistical metrics. The Random module was utilized to randomly split the dataset into training and testing sets, ensuring fair and generalizable analysis results. The data normalization process began by determining the due date and payment date. The due date was set to the 30th for months other than February; for February, it was set to the 28th. Customer payment data was converted to the datetime format to enable further calculations. The next step involved calculating payment delays by subtracting the customer's payment date from the due date. This difference was expressed in days of delay and normalized into binary values: 0 for on-time or early payments and 1 for late payments. This process produced data that revealed customer payment patterns, enabling the identification of on-time and late payments for each recorded period in the dataset. The normalization results covered all customer data, ensuring the dataset was processed comprehensively and accurately.

Data Validation

Group K-Fold Cross Validation was employed to ensure that validation was conducted without mixing data from a single customer between the training and test sets. This process began by normalizing the dataset into a numeric array format using numpy. Array, resulting in a binary vector normalize_bill, where a value of 1 represented timely or marginally late payments, and a value of 0 represented significantly late or unpaid bills. The dataset was then divided into seven folds using GroupKFold from the scikit-learn library, with customer indices as parameters to ensure that data from a single customer did not appear simultaneously in both the training set and test set. In each iteration, the data was split into a training set for model training and a test set for performance testing. Predictions were made on the test set using the average values from the training set, and model performance was evaluated using the Mean Squared Error (MSE) for each fold. The MSE results for each fold demonstrated consistency, with an average MSE of 0.4934, reflecting the model's ability to generalize across different customer groups.

Rolling Window Validation was used to validate the model based on the temporal nature of the dataset while preserving the chronological order of the data. This process began by setting a window size of 24 months, where data within the initial window served as the training set, while subsequent data served as the test set. In each iteration, the window was advanced by one month, so the training data always encompassed data within the moving window. In contrast, the test data consisted of data outside the window immediately following the training period. The Logistic Regression model was trained on the training set and tested on the test set to predict timely payments. Performance evaluation was conducted using accuracy and F1-Score, both of which were averaged to provide an overall view of model performance. The evaluation results showed an average accuracy of 0.5592 and an average F1-Score of 0.5598, reflecting the model's performance in accounting for the chronological nature of the data. Based on the analysis of these two validation techniques, Group K-Fold Validation is suitable for evaluating the model's generalization to customer groups, as it ensures no mixing of customer data between the training and test sets. However, this technique does not account for the temporal order of the data. Conversely, Rolling Window Validation is more appropriate for time-based applications, such as predicting electricity bill payment delays, as it preserves the chronological order of the data, although it requires more iterations. Therefore, the choice of validation technique should align with the data characteristics and application objectives.

Table 2. Analysis of Both Validation Techniques

Aspect	Group K-Fold Validation	Rolling Window Validation
Objective	Evaluate model generalization on customer groups	Evaluate prediction performance based on time
Data Splitting	Based on customer groups	Based on chronological order
Advantages	No mixing of customer data	Maintains data chronology

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Disadvantages	Does not consider chronological order	Requires more iterations
Evaluation Results	Average MSE: 0.4934	Average Accuracy: 0.5592; Average F1-Score: 0.5598

Random Forest

Modelling Data with a 3-Month Moving Average

The data modeling process using a 3-month Moving Average is carried out through several systematic steps. The first step is preparing the data for modeling, where the initial dataset includes several key features designed to aid in predicting payment delays. Each customer is assigned a unique identifier, CustomerID, while the pandas generate the payment dates.date_range() function with a monthly frequency. The binary target variable represents payment delays: 0 for on-time payments and 1 for late payments. Additionally, supplementary features are prepared, including lag features created using the shift() method based on CustomerID to record payment delays in previous months (e.g., lag_1, lag_2, and so on). The 3-month Moving Average is calculated using the rolling(window=3).mean() method to obtain the average payment delay over the last three months. The result is then shifted forward by one month to make it usable for predictions the following month. A month feature is also extracted from the Date column to capture seasonal patterns that might influence payment delays. Once the features are prepared, the data is split into training and testing sets to ensure the validity of future predictions. The split is performed temporally without shuffling, using the parameter shuffle=False in the train_test_split function to maintain the chronological order of the data:

```
# Bagi data menjadi train dan test set (berdasarkan waktu, tidak di-shuffle)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, shuffle=False)
```

The prepared dataset was then divided into two parts to ensure the validity of predictions and prevent data leakage. The training set, comprising 80% of the data, was used to train the model, a process where the model learns to recognize patterns in the data to make predictions. The test set, comprising 20% of the data, was used exclusively to evaluate the model's performance. The next stage involved training the model using a Random Forest Regressor to assess the impact of model complexity on prediction performance. During this process, the parameter n_estimators was adjusted from 1 to 280 trees to determine the optimal number of trees for the model. In contrast, the parameter random_state=42 was applied to ensure reproducibility of results. This process included the following steps:

```
for i, tree in enumerate(trees):
  # Melatih model Random Forest
  model = RandomForestRegressor(n_estimators=tree, random_state=42)
  model.fit(X_train, y_train)
```

The model was trained using training data (X_train and y_train) with variations in the number of trees, where each model was built based on a subset of the data and collectively generated predictions. After the training, predictions were made on the test data (X_test) to produce predicted values for payment delays. These predictions were then evaluated by comparing them to the actual data using the Mean Absolute Error (MAE) metric to measure the model's average prediction error in the original data units.

```
# Prediksi
y_pred = model.predict(X_test)

# Evaluasi model
mae = mean_absolute_error(y_test, y_pred)
errors.append(mae)
print(i, tree, mae)
```

The evaluation results of payment delay predictions using the Mean Absolute Error (MAE) metric indicate that the model has a relatively low prediction error rate. The following are the evaluation results of the model for various numbers of trees:

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

```
<ipython-input-31-ca2a94eb8959>:8: FutureWarning: 'M' is deprecated and will be removed in a future version, please use 'ME' instead.
  'Date': pandas.date_range(start='2018-01-01', periods=count_month, freq='M').tolist() * total_customer,
0 1 0.004419117647058823
1 2 0.004419117647058823
2 3 0.004419117647058823
3 4 0.004419117647058823
4 5 0.004419117647058823
5 6 0.004419117647058823
6 7 0.004419117647058823
7 8 0.004419117647058823
8 9 0.004419117647058823
9 10 0.004419117647058823
10 20 0.004419117647058823
11 30 0.004435171568627451
12 40 0.004431158088235295
13 50 0.0044383823529411765
14 60 0.004435171568627451
15 70 0.004443329831932772
16 80 0.004440303308823529
17 90 0.004437949346405229
18 100 0.004436066176470588
19 120 0.004437254901960783
20 140 0.004459164915966385
21 160 0.004460179227941176
22 180 0.004462418300653596
23 200 0.004466562500000001
24 220 0.004462249331550801
25 240 0.004460661764705883
26 260 0.004463178733031674
27 280 0.004494428396358542
```

MAE measures the average discrepancy between the model's predictions and the actual values, where a smaller MAE value indicates better model performance. Across various numbers of trees tested in the model, the MAE value remained relatively stable at approximately 0.0044. This stability suggests that increasing the number of trees does not significantly impact the prediction error rate. With such a low average error, the model can accurately and consistently predict payment delays.

The next step is Error Visualization Against the Number of Trees, based on the evaluation results, to understand the relationship between the number of trees and model performance. A line graph is created to display the error patterns. The steps for creating the graph are as follows:

```python
# Membuat plot
plt.plot(trees, errors, marker='o', linestyle='-', color='b')

# Menambahkan label dan judul
plt.xlabel('Number of Trees')
plt.ylabel('Error')
plt.title(f'Plot Error for {total_customer} customers and back month {back_months}')

# Menampilkan grid
plt.grid(True)

# Menampilkan plot
plt.show()
```

Based on the interpretation of the graph: The graph shows that the error fluctuates at a low number of trees (less than 50) but begins to stabilize after the number of trees reaches around 100–200. Adding more trees does not provide a significant performance improvement beyond a certain point (around 200–300 trees).

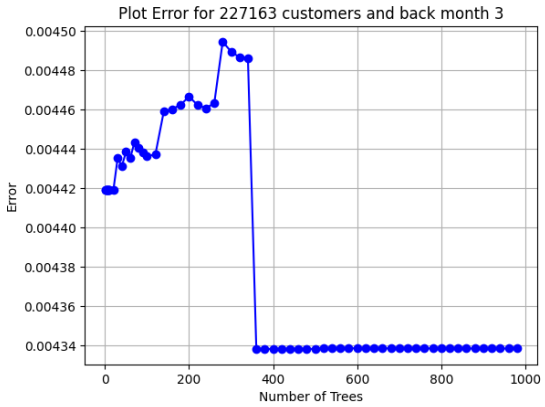

Fig 2. Error Analysis for 227,163 Customers with a Back Month of 3

This pattern is consistent across datasets with varying numbers of customers. In smaller datasets, the error shows more significant fluctuations with fewer trees but stabilizes after reaching a certain number of trees. In larger datasets, such as those with 227,163 customers, the error decreases significantly up to around 400 trees,

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

after which it remains constant. This indicates that the model has achieved optimal performance. Overall, the visualization demonstrates that while increasing the number of trees can enhance the model's stability, there is a threshold where adding more trees no longer provides significant benefits. Therefore, the Random Forest Regressor model exhibits stable and effective performance in predicting electricity payment delays, with optimal parameters ranging between 200–300 trees for various customer scenarios.

Based on the results and final interpretation from the five experiments conducted with different numbers of customers, the following general patterns were observed: High and fluctuating error due to the model not being complex enough to capture patterns in the data (Low Number of Trees (<50)). Significant error reduction as the model becomes more stable and better at capturing data patterns (Moderate Number of Trees (100–200)).No significant error reduction, indicating that the model has reached optimal performance (High Number of Trees (>200)). In conclusion, models with 200–300 trees demonstrate stable and efficient performance. Adding more trees beyond this range does not significantly improve accuracy but increases computational time.

Modelling Data with a 6-Month Moving Average

Modeling using a 6-Month Moving Average feature is designed to analyze and leverage historical payment delay patterns over extended periods. This approach gives customers a medium-term (six months) behavioral context compared to shorter monthly patterns. The initial step in this modeling involves preparing a dataset consisting of key columns such as CustomerID for unique customer identification, Date indicating payment dates with monthly frequency, and Delay, a binary target column reflecting timely payments (0) or delayed payments (1). Additionally, supplementary features like lag features, representing delays in previous months using the shift() method, and the six-month Moving Average calculated using rolling(window=6).mean(), are created to capture average delay patterns. The Month feature has also been added to account for seasonal payment patterns.

The processed dataset is then split into a training set and a test set using a temporal approach without randomization to maintain the time sequence, allocating 80% of the data for training and 20% for testing. This split ensures a simulation of future predictions based on historical data. The training set is used to train the model, while the test set evaluates performance. The model used is the Random Forest Regressor from scikit-learn, trained with varying numbers of trees (estimators) ranging from 1 to 280 to observe the impact of tree quantity on model performance. The parameter random_state=42 is employed to ensure result reproducibility. Evaluation uses the Mean Absolute Error (MAE) metric to measure the average error between model predictions and actual values.

```
# Prediksi
y_pred = model.predict(X_test)

# Evaluasi model
mae = mean_absolute_error(y_test, y_pred)
errors.append(mae)
print(i, tree, mae)
```

The prediction of payment delays was evaluated using the Mean Absolute Error (MAE) metric to measure the average error between the model's predicted results and the actual values. The following are the evaluation results of the model for different numbers of trees:

```
39 520 0.004230769230769231
40 540 0.004230769230769231
41 560 0.004230769230769231
42 580 0.004230769230769231
43 600 0.004230769230769231
44 620 0.004230769230769231
45 640 0.004230769230769231
46 660 0.004230769230769231
47 680 0.004230769230769231
48 700 0.004230769230769231
49 720 0.004230769230769231
50 740 0.004230769230769231
51 760 0.004230769230769231
52 780 0.004230769230769231
53 800 0.004230769230769231
54 820 0.004230769230769231
55 840 0.004230769230769231
56 860 0.004230769230769231
57 880 0.004230769230769231
58 900 0.004230769230769231
59 920 0.004230769230769231
60 940 0.004230769230769231
61 960 0.004230769230769231
62 980 0.004230769230769231
```

The evaluation results indicate that the model produced consistent predictions with a stable MAE value of approximately 0.0044. This low MAE value demonstrates that the average prediction error for payment delays is relatively small, indicating that the model performs well in identifying customer payment patterns. However,

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

further optimization is required to improve prediction accuracy, especially for delay categories representing a minority in the dataset.

Subsequently, the final results compare the model's predictions with the actual payment delay data to assess how closely the predictions align with the exact values:

```
result = pandas.DataFrame({'Actual': y_test.values, 'Predicted': y_pred})

print(result.head())
```

Here is an example of the prediction results for the first few customers:

	Actual	Predicted
0	0	0.0
1	0	0.0
2	0	0.0
3	0	0.0
4	0	0.0

The relationship between Mean Absolute Error (MAE) and the number of trees varies depending on the size of the customer dataset. The error fluctuates significantly for a small dataset with 10 customers, especially with fewer than 50 trees. However, it begins to decrease and stabilize when the number of trees reaches 100, though minor fluctuations persist with more significant numbers of trees. For a dataset with 100 customers, the graph shows error fluctuations with fewer than 50 trees but demonstrates a consistent decline and stabilization after 100 trees, indicating improved model accuracy with more data.

For larger datasets, such as 1,000 customers, the error decreases significantly as the number of trees increases up to 200 and then stabilizes. However, there is a slight spike around 400 trees, likely due to noise or model complexity. Meanwhile, the error remains relatively stable for a dataset with 6,000 customers, even with fewer than 50 trees. Beyond 200 trees, there is no significant decrease in error, even when the number of trees is increased to 1,000. This indicates that larger datasets provide the model with better generalization capability.

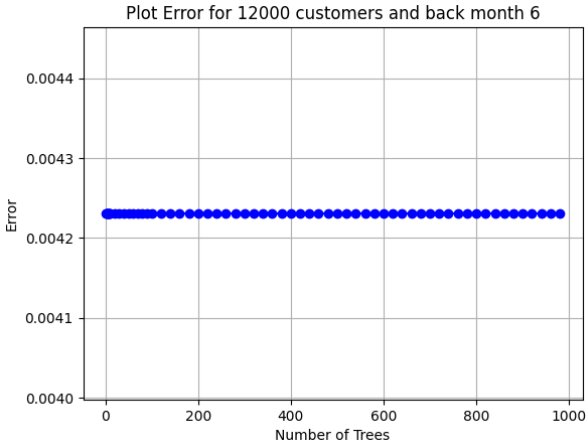

Fig 3. Error Analysis for 12,000 Customers with Back Month 6

The error remained stable from the beginning in a large dataset with 12,000 customers, showing no significant fluctuations even with a small number of trees. After reaching around 100–200 trees, the error showed no significant changes, even as the number of trees increased to 1,000, indicating that the large dataset provided sufficient information for the model to produce consistent and stable predictions. This pattern highlights that the dataset size influences the model's sensitivity to the number of trees, with larger datasets providing better stability and accuracy.

Based on the visualization results, several patterns can be interpreted regarding the relationship between the number of trees in the Random Forest model and the error value. With a low number of trees, especially fewer than 50, the error tends to be fluctuating and unstable. This is because the model's complexity is still too low, preventing it from effectively capturing the data patterns. As the number of trees increases to around 100–200, the error decreases significantly, indicating that the model has achieved sufficient complexity to understand the data patterns and produce more accurate predictions. After the number of trees reaches 200–300, the error stabilizes without showing significant improvement, even as the number of trees continues to increase. This suggests that adding more trees does not necessarily improve model accuracy because the optimal capacity has already been reached. The dataset size also affects error fluctuation; the error is more volatile in smaller datasets (10–100 customers) because the limited amount of data results in less consistent predictions. Conversely, in larger datasets

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

(1,000–12,000 customers), the error is more stable as the model has more data to train on and can capture patterns more effectively. The optimal number of trees ranges between 100 and 200, where the model balances complexity and performance. Adding more trees beyond this range does not significantly enhance model accuracy.

Modelling Data with a 12-Month Moving Average

The initial stage in the modeling process involves preparing a dataset consisting of essential features to predict customer payment delays. This dataset includes three main columns to ensure the data is systematically structured and relevant. The first column, CustomerID, is a unique identifier used to distinguish each customer individually. The second column, Date, contains date data with a monthly frequency generated using the pandas.date_range() function. This column ensures the data is organized temporally to support trend or pattern analysis. The third column, Delay, is the target column containing binary values, where 1 indicates a payment delay and 0 indicates an on-time payment. This dataset structure is designed to facilitate predictive analysis by leveraging customer information and their payment patterns over time.

```
# lag_feature
for back_month in range(back_months):
  df[f'lag_{back_month+1}'] = df.groupby('CustomerID')['Delay'].shift(back_month+1)

# Fitur moving average 12 bulan
df['MA_12'] = df.groupby('CustomerID')['Delay'].rolling(window=12).mean().shift(1).reset_index(0, drop=True)

# Fitur month
df['month'] = df['Date'].dt.month

# Hapus baris dengan NaN akibat lag
df.dropna(inplace=True)

# Data dan target
features = [f'lag_{i+1}' for i in range(back_months)] + ['month', 'CustomerID']
X = df[features]
y = df['Delay']
```

The additional features include several steps to enhance the predictive model's performance. First, Lag Features were added to represent payment delays in the previous months (lag_1, lag_2, ..., lag_12). These features were calculated based on the Delay column using the shift() method, grouped by CustomerID, providing historical context to customers' payment patterns. Next, a 6-month Moving Average was computed to capture the average delay behavior of customers over the past six months. This process was performed using the rolling(window=6).mean() method for each customer, enabling more accurate identification of short-term delay patterns. Additionally, a Month Feature was included to account for seasonal patterns, as electricity payments often exhibit variations based on specific months, such as during holidays or year-end periods. After preparing these features, the dataset was split into a Train Set (80%) and a Test Set (20%) while maintaining the temporal order of the data. This approach ensured that historical data was used to predict future data, making the prediction simulation more realistic. A Random Forest Regressor from scikit-learn was used during the model training phase. The model was trained with various values of the n_estimators parameter (number of trees), ranging from 1 to 1000, to evaluate the impact of the number of trees on prediction accuracy. The random_state parameter was set to 42 to ensure reproducibility of experimental results. Once the training was complete, the model made predictions on X_test, which consisted of data that the model had not seen before.

```
# Prediksi
y_pred = model.predict(X_test)

# Evaluasi model
mae = mean_absolute_error(y_test, y_pred)
errors.append(mae)
print(i, tree, mae)
```

The model was tested using a test set, and its performance was evaluated using the Mean Absolute Error (MAE) metric, which represents the average error between actual and predicted values. The following is the model evaluation results for various numbers of trees:

```
54 820 0.0038727780074410906
55 840 0.0038728140973903692
56 860 0.0038728485087373534
57 880 0.0038728091294298918
58 900 0.003872669491525423
59 920 0.003872697126013264
60 940 0.0038728287654766196
61 960 0.003872888712335216
62 980 0.0038727660555747732
```

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

Based on the evaluation results, the MAE value consistently remained at 0.00387 across various numbers of trees in the model, indicating that the average prediction error for payment delays is very low. This consistency in MAE demonstrates that the model accurately predicts payment delays and effectively recognizes customer payment patterns. Its strong performance underscores the model's reliability in handling the task of predicting payment delays. The model's predictions were compared with actual payment delay data to assess how closely the predictions aligned with the actual values. For illustration, several prediction examples for specific customers showed good alignment with actual data. Furthermore, the relationship between Mean Absolute Error (MAE) and the number of trees in the model was mapped using a graph for various customer dataset sizes.

For a dataset with 10 customers, the error varied significantly when the number of trees was below 50, indicating instability in the model during the early stages. Stability began to emerge after the number of trees reached 100. However, small fluctuations persisted up to 1,000 trees, as the small dataset size made the model more sensitive to changes in tree structure. A similar pattern was observed for a dataset with 100 customers, where the error initially fluctuated until the number of trees reached 50–100. After that, the model stabilized, with minor fluctuations persisting up to 1,000 trees. This dataset provided more information to the model, resulting in more accurate predictions. For a dataset with 1,000 customers, the graph showed a drastic reduction in error up to around 200 trees, after which the error stabilized with minimal fluctuations. A larger dataset helps the model capture payment delay patterns more effectively. For a dataset with 6,000 customers, a significant reduction in error occurred up to around 200 trees. Although slight fluctuations were observed with a more significant number of trees, these results demonstrate that a more extensive dataset provides better generalization capacity for the model.

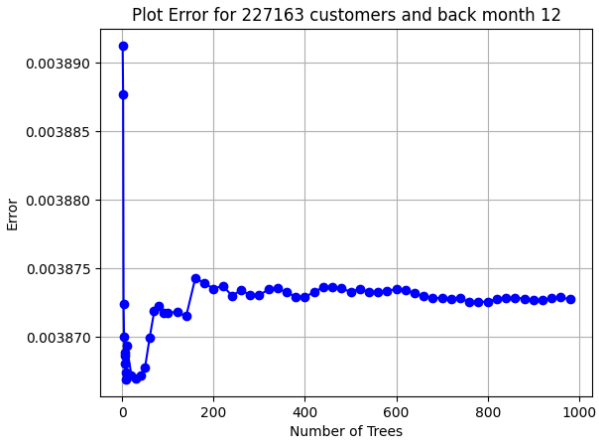

Fig 4. Error untuk 227.163 Pelanggan dengan Back Month 12

For 227,163 customers, the graph shows that the error becomes highly stable and consistent after reaching approximately 100–200 trees. Adding more trees beyond this point does not yield significant improvements in model performance, as the large dataset size is already sufficient to capture payment delay patterns effectively, resulting in an accurate and stable model.

The graph visualization reveals several general patterns regarding the relationship between the number of trees in the model and the error values (Mean Absolute Error/MAE). With a low number of trees (less than 50), the error tends to fluctuate because the model lacks sufficient complexity to understand the patterns in the data. A significant reduction in error occurs when the number of trees is 100–200, indicating that the model begins to reach adequate complexity to capture patterns more effectively. After the number of trees reaches 200–300, the error stabilizes and does not show a meaningful decline despite adding more trees, suggesting that additional complexity no longer provides significant performance improvements. The dataset size also influences the error pattern. With smaller datasets (10–100 customers), the error is more volatile due to limited data, making the model more susceptible to minor variations in tree structure. In contrast, the error tends to be more stable with larger datasets (1,000 customers and above) because the model has sufficient data to identify patterns more accurately. Overall, the optimal number of trees is within the range of 100–200 trees, where the model achieves a good balance between complexity and performance, delivering accurate predictions without overfitting.

Comparison of Model Evaluation on Random Forest

Modelling with Moving Average features over 3, 6, and 12-month periods highlights different customer payment delay patterns over varying timeframes. Each period demonstrates different accuracy levels based on the model's ability to recognize payment patterns from historical data. The 3-month Moving Average has an average MAE of 0.0044, with significant error fluctuations at low tree counts, stabilizing at 200–300 trees. However, this period results in higher errors than other periods and is less effective in capturing long-term patterns, particularly

*name of corresponding author

Sinkron : Jurnal dan Penelitian Teknik Informatika
Volume 9, Number 1, January 2025
DOI : https://doi.org/10.33395/sinkron.v9i1.14340
e-ISSN : 2541-2019
p-ISSN : 2541-044X

on large datasets. The 6-month Moving Average improves performance with an average MAE of 0.0042. Errors decrease more rapidly than in the 3 months, achieving stability at 200 trees, although small fluctuations persist on datasets with fewer customers. The 12-month Moving Average yields the best results with an average MAE of 0.00387. The error graph indicates stable performance even with fewer trees, achieving stability starting from 100 trees. Longer historical information enables the model to capture long-term patterns more effectively, resulting in more accurate and consistent predictions. Therefore, the 12-month Moving Average is recommended for modelling payment delays, especially on large-scale datasets.

Data Validation with Test Data

During the validation phase, the Random Forest Regressor model was tested using test data to evaluate its predictive performance. The dataset was processed into a 12-month time window to predict delays in the 13th month, with seasonal variables added to capture periodic patterns. The model was trained with key parameters such as n_estimators set to 220, test_size of 10%, and random_state of 42. The evaluation was conducted using Mean Squared Error (MSE), which recorded an average of 3.9089 on the test data. MSE varied among customers, with lower values without significant delay patterns but higher values for customers with complex delay variations. The model tends to produce predictions close to zero, reflecting its accuracy for customers with low or stable delay patterns, though it struggles to capture significant variations.

Delay Predictions for 2024

The model was used to predict electricity payment delays for 2024 iteratively, based on the past 12 months of data. The process began with delay data from the previous year, including seasonal variables to maintain periodic patterns. Predictions were made for each month, with the first month's prediction replacing the oldest data in the time window, repeated until all 12 months were predicted. The prediction results show that the model tends to output zero for most customers, reflecting the dominance of customers with low or stable delay patterns. However, the model must be revised to predict delays for customers with unstable patterns or significant delay amounts. Nonetheless, the model successfully captures patterns in customers with stable delays, producing predictions close to actual values, though performance declines for customers with complex patterns.

DISCUSSIONS

This study develops a payment delay prediction model utilizing the Random Forest Regressor algorithm, incorporating additional features such as lag features and moving averages for 3, 6, and 12 months. These features aim to capture seasonal patterns and customer payment behaviour more accurately. Previous literature has employed similar approaches to predict financial risks, demonstrating that ensemble-based algorithms like Random Forest handle temporal data and feature complexity (Mashrur et al., 2020; Pavlicko et al., 2021). This study reinforces those findings by showing that using a 12-month moving average yields the best performance, with an average Mean Absolute Error (MAE) of 0.00387, indicating the model's ability to recognize long-term patterns. Additionally, optimizing the number of trees to between 100 and 200 improves prediction stability, consistent with studies highlighting the importance of parameter tuning in ensemble models (Lopez & Jeronimo, 2015; Mienye & Sun, 2022). This approach provides practical benefits for PLN UP3 Makassar Utara by identifying customers likely to delay payments and supporting more strategic and efficient bill management.

The payment delay prediction system developed for PLN UP3 Makassar Utara uses the Random Forest Regressor approach trained on historical customer payment data. It enables the identification of customers at high risk of payment delays based on consistent or unstable payment patterns. This system aligns with previous research demonstrating that machine learning algorithms, such as Random Forest, effectively predict customer behaviour, including payment delays, by analysing historical patterns and seasonal variables (Chapman & Desai, 2023). Leveraging these predictions, PLN can prioritize invoice delivery to high-risk customers, particularly during months with increasing delay trends, and integrate more proactive communication strategies aligned with seasonal patterns. These findings support prior research emphasizing the importance of risk-based customer segmentation to enhance receivables management efficiency and mitigate payment delay risks (Aysan et al., 2024). Thus, this approach is operationally relevant and contributes to the literature on applying machine learning to financial management.

The payment delay prediction system developed using the Random Forest Regressor algorithm, with historical feature analysis such as a 12-month moving average, has demonstrated significant capability in accurately identifying customer payment delay patterns. This aligns with previous findings highlighting the effectiveness of Random Forest in handling complex structured data for predictive analysis (Aria et al., 2021; D. Zhang et al., 2020). Prior studies also show that historical features, such as moving averages, aid in capturing long-term trends in customer payment patterns, particularly in segments with stable payment behaviour (Herhausen et al., 2019; Seyedan & Mafakheri, 2020). In the context of PLN UP3 Makassar Utara, this model enables resource optimization by prioritizing invoice delivery and reminders to customers with high payment delay risks. This is similar to

*name of corresponding author

e-ISSN : 2541-2019
p-ISSN : 2541-044X

payment risk management strategies proven effective in various prior studies (Saeidi et al., 2019). By leveraging these predictions, PLN can improve payment management efficiency, accelerate cash flow, and reduce long-term delinquency risks, as suggested in the literature discussing predictive technology implementation for enhancing corporate financial management (Chakraborty, 2020; Oyedokun et al., 2024).

The results of this study surpass previous research by integrating Random Forest Regressor to predict electricity payment delays using more detailed historical feature analysis, such as 3-, 6-, and 12-month moving averages. This approach captures short-term patterns and accurately identifies long-term customer trends. Compared to Schonlau & Zou's (2020) Study, which only utilized Random Forest for classification and regression tasks without focusing on temporal patterns, this research offers a more comprehensive approach by preserving time order to enhance prediction realism. Unlike Teles et al.'s (2021) research, which applied Random Forest to credit risk analysis without incorporating temporal patterns, this study demonstrates that processing temporal data can improve prediction stability and accuracy in the payment domain. Furthermore, while Rahmi et al.'s (2023) study also used Random Forest to predict payment delays, this study goes further by evaluating the impact of moving averages over various timeframes, showing that the 12-month moving average provides the best performance with an MAE of 0.00387. This supports previous findings on the effectiveness of Random Forest in capturing complex patterns while contributing new insights through temporal approaches and model parameter optimization strategies. The findings of this study support previous research on the effectiveness of Random Forest for historical data-based prediction while extending its application with a focus on time-based validation to improve payment delay prediction efficiency.

This study has several limitations that need to be acknowledged. First, the dataset originates from a single operational area, PLN UP3 Makassar Utara. Therefore, the prediction results must be more generalizable to other regions with different customer characteristics and payment patterns. Second, while the 12-month moving average yielded the best results, the model tends to be less effective in predicting customers with highly unstable or sporadic delay patterns, which may affect prediction accuracy in specific categories. Third, the Random Forest model requires considerable computational resources, particularly with large-scale datasets, which could pose challenges for operational systems with limited infrastructure. Additionally, this study does not consider external factors such as electricity tariff changes or macroeconomic conditions that may influence customers' ability to pay on time. These findings imply that a Random Forest-based payment delay prediction system can significantly enhance payment management efficiency and reduce delinquency risks when integrated with proactive invoice prioritization and receivables management strategies. This model allows PLN to allocate resources more optimally and improve the company's financial stability. However, to broaden its benefits, further research should involve data from different regions and consider external factors to enhance model accuracy and generalizability.

.

CONCLUSION

This study successfully developed a bill payment delay prediction system using the Random Forest Regressor algorithm, supported by the historical data analysis of PLN UP3 Makassar Utara customers. The system utilizes moving average features for 3, 6, and 12-month periods, with the best results achieved using a 12-month moving average, yielding an average Mean Absolute Error (MAE) value of 0.00387. This system enables PLN to identify high-risk customers, prioritize invoice delivery, and optimize cash flow management. The analysis graph shows prediction stability after the number of trees reaches 100–200, improving accuracy on larger datasets. However, this study is limited to data from a specific region, does not account for external factors such as electricity tariffs or economic conditions, and requires significant computational resources. Future research is recommended to expand the geographical scope, consider external factors, and reduce computational demands for broader and more efficient implementation.

REFERENCES

Ampomah, E. K., Qin, Z., & Nyame, G. (2020). Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. *Information (Switzerland)*, *11*(6), 1–21. https://doi.org/10.3390/info11060332

Aria, M., Cuccurullo, C., & Gnasso, A. (2021). A comparison among interpretative proposals for Random Forests. *Machine Learning with Applications*, *6*(1), 1–8. https://doi.org/10.1016/j.mlwa.2021.100094

Asselman, A., Khaldi, M., & Aammou, S. (2023). Enhancing the prediction of student performance based on the machine learning XGBoost algorithm. *Interactive Learning Environments*, *31*(6), 3360–3379. https://doi.org/10.1080/10494820.2021.1928235

Aysan, A. F., Ciftler, B. S., & Unal, I. M. (2024). Predictive Power of Random Forests in Analyzing Risk Management in Islamic Banking. *Journal of Risk and Financial Management*, *17*(3), 1–19. https://doi.org/10.3390/jrfm17030104

Boutahir, M. K., Hessane, A., Farhaoui, Y., Azrour, M., Benyeogor, M. S., & Innab, N. (2024). Meta-Learning Guided Weight Optimization for Enhanced Solar Radiation Forecasting and Sustainable Energy

*name of corresponding author

Management with VotingRegressor. *Sustainability (Switzerland)*, *16*(13), 1–10. https://doi.org/10.3390/su16135505

Chakraborty, G. (2020). Evolving profiles of financial risk management in the era of digitization: The tomorrow that began in the past. *Journal of Public Affairs*, *20*(2), 1–9. https://doi.org/10.1002/pa.2034

Chapman, J. T. E., & Desai, A. (2023). Macroeconomic Predictions Using Payments Data and Machine Learning. *Forecasting*, *5*(4), 652–683. https://doi.org/10.3390/forecast5040036

Daryanti, D., & Sudarwanto, A. S. (2024). Good Governance in the Policy on Using Solar Cells as Efforts to Reduce Emissions in Indonesia. *Good Governance in the Policy on Using Solar Cells*, 441–447. https://doi.org/10.2991/978-2-38476-218-7_74

Ghiasi, M. M., & Zendehboudi, S. (2021). Application of decision tree-based ensemble learning in the classification of breast cancer. *Computers in Biology and Medicine*, *128*(1), 1–35. https://doi.org/10.1016/j.compbiomed.2020.104089

Herhausen, D., Kleinlercher, K., Verhoef, P. C., Emrich, O., & Rudolph, T. (2019). Loyalty Formation for Different Customer Journey Segments. *Journal of Retailing*, *95*(3), 9–29. https://doi.org/10.1016/j.jretai.2019.05.001

Ibekwe, K. I., Umoh, A. A., Nwokediegwu, Z. Q. S., Etukudoh, E. A., & Etukudoh, E. A. (2024). Energy Efficiency in Industrial Sectors: a Review of Technologies and Policy Measures. *Engineering Science & Technology Journal*, *5*(1), 169–184. https://doi.org/10.51594/estj/v5i1.742

Javed, M. F., Siddiq, B., Onyelowe, K., Khan, W. A., & Khan, M. (2024). Metaheuristic optimization algorithms-based prediction modeling for titanium dioxide-Assisted photocatalytic degradation of air contaminants. *Results in Engineering*, *23*(1), 1–20. https://doi.org/10.1016/j.rineng.2024.102637

Kalusivalingam, A. K., Sharma, A., Patel, N., & Singh, V. (2020). Enhancing Predictive Business Analytics with Deep Learning and Ensemble Methods : A Comparative Study of LSTM Networks and Random Forest Algorithms. *International Journal of AI and ML*, *1*(2), 1–23. https://www.cognitivecomputingjournal.com/index.php/IJAIML-V1/article/view/58

Kalusivalingam, A. K., Sharma, A., Patel, N., & Singh, V. (2021). Leveraging Deep Learning and Random Forest Algorithms for AI-Driven Genomics in Personalized Medicine Authors : *International Journal of AI and ML*, *2*(3), 1–23. https://www.cognitivecomputingjournal.com/index.php/IJAIML-V1/article/view/79

Liu, W., Qi, X., Lu, J., Jia, X., & Li, P. (2020). Finite-time fault-tolerant control for nonlinear systems with input quantization and its application. *IEEE Transactions on Circuits and Systems II: Express Briefs*, *67*(7), 1249–1253. https://doi.org/10.1109/TCSII.2019.2928460

Lopez, R. F., & Jeronimo, J. M. R. (2015). Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal. *Expert Systems with Applications*, *42*(13), 5737–5753. https://doi.org/10.1016/j.eswa.2015.02.042

Mashrur, A., Luo, W., Zaidi, N. A., & Robles-Kelly, A. (2020). Machine learning for financial risk management: A survey. *IEEE Access*, *8*(1), 203203–203223. https://doi.org/10.1109/ACCESS.2020.3036322

Maulidia, M., Dargusch, P., Ashworth, P., & Ardiansyah, F. (2019). Rethinking renewable energy targets and electricity sector reform in Indonesia: A private sector perspective. *Renewable and Sustainable Energy Reviews*, *101*(1), 231–247. https://doi.org/10.1016/j.rser.2018.11.005

Mienye, I. D., & Sun, Y. (2022). A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. *IEEE Access*, *10*(1), 99129–99149. https://doi.org/10.1109/ACCESS.2022.3207287

Mohammed, A., & Kora, R. (2023). A comprehensive review on ensemble deep learning: Opportunities and challenges. *Journal of King Saud University - Computer and Information Sciences*, *35*(2), 757–774. https://doi.org/10.1016/j.jksuci.2023.01.014

Negara, S. D., & Ramayandi, A. (2020). Laying the Foundations for Future Growth Acceleration? *Bulletin of Indonesian Economic Studies*, *56*(1), 1–21. https://doi.org/10.1080/00074918.2020.1743014

Ong, A. K. S., Cordova, L. N. Z., Longanilla, F. A. B., Caprecho, N. L., Javier, R. A. V., Borres, R. D., & German, J. D. (2023). Purchasing Intentions Analysis of Hybrid Cars Using Random Forest Classifier and Deep Learning. *World Electric Vehicle Journal*, *14*(8), 1–26. https://doi.org/10.3390/wevj14080227

Oyedokun, O., Ewim, S. E., Oyeyemi, O. P., Researcher, I., & Texas, D. (2024). Leveraging advanced financial analytics for predictive risk management and strategic decision-making in global markets. *Multidisciplinary Studies*, *1*(1), 1–26. https://gsjournals.com/gjrms/sites/default/files/GJRMS-2024-0051.pdf

Pavlicko, M., Durica, M., & Mazanec, J. (2021). Ensemble model of the financial distress prediction in visegrad group countries. *Mathematics*, *9*(16), 1–26. https://doi.org/10.3390/math9161886

Preuveneers, D., Tsingenopoulos, I., & Joosen, W. (2020). Resource usage and performance trade-offs for machine learning models in smart environments. *Sensors (Switzerland)*, *20*(4), 1–27. https://doi.org/10.3390/s20041176

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(3), 1–15.

*name of corresponding author

https://doi.org/10.1002/widm.1301

Rahmi, I. A., Afendi, F. M., & Kurnia, A. (2023). Metode AdaBoost dan Random Forest untuk Prediksi Peserta JKN-KIS yang Menunggak. *Jambura Journal of Mathematics*, *5*(1), 83–94. https://doi.org/10.34312/jjom.v5i1.15869

Saeidi, P., Saeidi, S. P., Sofian, S., Saeidi, S. P., Nilashi, M., & Mardani, A. (2019). The impact of enterprise risk management on competitive advantage by moderating role of information technology. *Computer Standards and Interfaces*, *63*(1), 67–82. https://doi.org/10.1016/j.csi.2018.11.009

Salman, H. A., Kalakech, A., & Steiti, A. (2024). Random Forest Algorithm Overview. *Babylonian Journal of Machine Learning*, *2024*(1), 69–79. https://doi.org/10.58496/bjml/2024/007

Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. *Stata Journal*, *20*(1), 3–29. https://doi.org/10.1177/1536867X20909688

Seyedan, M., & Mafakheri, F. (2020). Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. *Journal of Big Data*, *7*(1), 1–22. https://doi.org/10.1186/s40537-020-00329-2

Teles, G., Rodrigues, J. J. P. C., Rabêlo, R. A. L., & Kozlov, S. A. (2021). Comparative study of support vector machines and random forests machine learning algorithms on credit operation. *Software - Practice and Experience*, *51*(12), 2492–2500. https://doi.org/10.1002/spe.2842

Usman, F. O., Ani, E. C., & Ebirim, W. (2024). Integrating Renewable Energy Solutions in the Manufacturing Industry: Challenges and Opportunities: a Review. *Engineering Science & Technology Journal*, *5*(3), 674–703. https://doi.org/10.51594/estj.v5i3.865

Wang, F., Wang, Y., Ji, X., & Wang, Z. (2022). Effective Macrosomia Prediction Using Random Forest Algorithm. *International Journal of Environmental Research and Public Health*, *19*(6), 1–10. https://doi.org/10.3390/ijerph19063245

Zhang, D., Yin, C., Zeng, J., Yuan, X., & Zhang, P. (2020). Combining structured and unstructured data for predictive models: a deep learning approach. *BMC Medical Informatics and Decision Making*, *20*(1), 1–11. https://doi.org/10.1186/s12911-020-01297-6

Zhang, Y., Liu, J., & Shen, W. (2022). A Review of Ensemble Learning Algorithms Used in Remote Sensing Applications. *Applied Sciences (Switzerland)*, *12*(17), 1–20. https://doi.org/10.3390/app12178654

Zhou, X., Lu, P., Zheng, Z., Tolliver, D., & Keramati, A. (2020). Accident Prediction Accuracy Assessment for Highway-Rail Grade Crossings Using Random Forest Algorithm Compared with Decision Tree. *Reliability Engineering and System Safety*, *200*(1), 1–20. https://doi.org/10.1016/j.ress.2020.106931

Zhou, Y., Li, S., Zhou, C., & Luo, H. (2019). Intelligent Approach Based on Random Forest for Safety Risk Prediction of Deep Foundation Pit in Subway Stations. *Journal of Computing in Civil Engineering*, *33*(1), 1–14. https://doi.org/10.1061/(asce)cp.1943-5487.0000796

*name of corresponding author