



Predicting Crime Time Intervals Using Machine Learning Models

Arief Deswandi¹, Widi Hastomo^{2,*}

^{1,2}Information of Technology Department, Institute of Technology and Business Ahmad Dahlan,
South of Tangerang, Indonesia

Email: ¹arief.deswandi@itb-ad.ac.id, ²widie.has@gmail.com

Abstract

Efforts to predict the time interval of crime are very important in supporting crime prevention and handling. This study aims to explore the ability of machine learning models to predict the time of crime using a dataset from the Central Statistics Agency of Indonesia (BPSI). The methods used include Decision Tree (DT), XGBoost, and CatBoost. These models are compared based on the mean absolute percentage error (MAPE) value to assess their accuracy. The results show that XGBoost achieved the lowest MAPE value of 8.29%, followed by Decision Tree with MAPE of 9.14% and CatBoost with MAPE of 9.68%. XGBoost achieved significant accuracy, demonstrating its potential to provide more accurate predictions. With a MAPE value of 8.29%, XGBoost has strong practical applications in crime prevention efforts, allowing authorities to be more effective in responding to potential crimes in the future.

Keywords: crime time interval, machine learning, prediction

1. INTRODUCTION

Research on the classification of crime time intervals using machine learning approaches is becoming increasingly important given the increasing crime rates in various countries [1]. This phenomenon indicates the need for innovation in crime analysis to improve the effectiveness of law enforcement and prevention strategies [2]. Traditionally, crime analysis is based on geographic patterns or offender profiles, but understanding crime time patterns can provide important additional insights for law enforcement. Classification of crime time intervals is becoming increasingly relevant because it allows the identification of crime patterns that can be used to optimize the allocation of law enforcement resources [3].

One of the challenges in crime analysis is the ever-increasing complexity of data, especially with the emergence of information technology [4]. Large and diverse data require sophisticated analytical approaches to uncover relevant patterns. Machine learning approaches are promising as a possible solution to process big data and formulate accurate predictions [5]. By utilizing machine learning



algorithms, this study aims to identify patterns and classifications of the time intervals of criminal acts, which can be the basis for law enforcement to develop more effective prevention strategies.

However, the application of machine learning technology in the classification of crime time can raise a number of social issues that need to be considered [6]. One of the main issues is the potential for algorithmic bias that can exacerbate social injustice [7]. If the data used to train a machine learning model does not fairly represent the population, the model may produce biased results against certain groups or regions [8]. This has the potential to reinforce negative stereotypes about minority groups or low-income areas that may be more often the focus of predictions [9]. In addition, the use of predictive technology can raise privacy issues, where the collection and analysis of personal data can lead to violations of individuals' privacy rights [10].

Previous studies have shown the potential of machine learning in crime analysis [11],[12], but there is still a need to develop more sophisticated and detailed approaches. Considering the development of technology and increasing accessibility of data, this study is an important step in formulating more effective solutions in law enforcement. It is hoped that the results of this study will make a significant contribution to the development of crime analysis systems that can improve public safety and the effectiveness of law enforcement as a whole.

Several recent studies related to the classification of criminal acts have been widely conducted. The study conducted by [12] using the DT, Logistic Regression (LR), Random Forest (RF), and XGBoost methods achieved an average accuracy of 0.961. Previous research by [13] using the LR, K Nearest Neighbors (KNN), Naïve Bayes (NB), DT, and XGBoost methods achieved the best accuracy of 0.996. Research by [14] using the OVR XGBoost and OVO-XGBoost methods. Previous research by [15] using the Self-Organizing Map (SOM) method. Research by [16] using the KNN, RF, Adaptive Boosting Classifier, Gradient Boosting Classifier, and Extra Trees Classifier methods with a dataset in the city of Bangalore in South India.

Research conducted by [17] using the XGboost method for predicting and preventing crime. Previous research by [18] using the LR, RF, Lightgbm, and Xgboost methods using a dataset from the San Francisco police. Research by [19] using the Ada Boosting (AB), DT, XGB, LR, RF, and Support Vector Machine (SVM) methods using a dataset from Nigeria. Research on crime prediction by [20] using the XGBoost, DT, and RF methods using a dataset of crimes in San Francisco. Research conducted by [21] used the Principle Component Analysis method on RF and DT. Research by [22] to predict crime rates in New York City used the SVM, RF, and XGBoost methods.

Based on a number of literatures obtained, research on the classification of the time interval of the occurrence of criminal acts has been widely conducted by comparing several machine learning methods, but not many are combined with the categorical boosting method. Therefore, in this study, what is developed is the classification of the time interval of the occurrence of criminal acts using the Decision Trees model, Extreme Gradient Boosting, and Categorical Boosting to detect crimes.

The annual dataset available in the form of hours, minutes, and seconds adds to the difficulty due to the high granularity of the data and irregular temporal fluctuations. Machine learning, especially XGBoost and CatBoost, is able to overcome this problem with its advantages in handling irregular and non-linear data [23], [24]. XGBoost, with its advanced boosting technique, is able to iteratively minimize prediction errors [25], while CatBoost addresses the problem of category bias and handles large-scale datasets quickly [26]. Both models significantly improve prediction performance, enabling more timely and accurate analysis [27], thus supporting more responsive and proactive crime prevention strategies.

The application of machine learning models such as XGBoost and CatBoost to crime datasets from Indonesia is novel in the context of crime prediction in the country. These models are able to process complex time-series data more accurately than traditional approaches, making them potential tools for data-driven law enforcement strategies. These models offer significant contributions in supporting law enforcement to formulate more efficient, responsive, and targeted crime prevention policies, which can ultimately improve public security in various regions of Indonesia.

2. METHODS

This study uses a crime clock dataset obtained from the Badan Pusat Statistik Indonesia (BPS) [28]. The dataset consists of 34 regional police in Indonesia from 2000 to 2022. The dataset is annual data, with units of hours, minutes, and seconds with integer type. Figure 1 is a representation of the dataset that has gone through the wrangling process.

2.1. Dataset

The data wrangling process includes several stages to clean and prepare data so that it is ready for use in analysis or modelling. Starting with gathering data, followed by loading the dataset into the Google Colab platform. The next process is assessing and manipulating by deleting NaN, renaming columns, reshaping data frames, separating hours, minutes, and seconds into new columns, converting data types, converting to seconds, and calculating time intervals in seconds. Next,

prepare the data for analysis time-based by making the time column the index of the dataframe.

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 23 entries, 2000-01-01 to 2022-01-01
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  ---
0    jateng      23 non-null     int64
1    jabar       23 non-null     int64
2    jatim       23 non-null     int64
3    sumut       23 non-null     int64
4    metro       23 non-null     int64
dtypes: int64(5)
memory usage: 1.1 KB
time: 5.5 ms (started: 2024-08-31 12:37:48 +00:00)
```

Figure 1. Crime clock dataset

The dataset falls into the category of time series data that can be used in machine learning algorithms. Decision tree regression, extreme gradient boosting, and categorical boosting are used as options to predict the likelihood of crime occurring at a certain time (crime clock).

2.2. Decision Trees (DT)

Decision trees work by breaking data into branches based on certain features, such as time in hours, minutes, and seconds, and then making decisions in the form of a branching tree [29]. Each branch in the tree represents a question related to those variables, and the end result of each branch is a prediction of when a crime will occur. Figure 2 is a pseudocode of DT.

```
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier(
    max_depth=5,          # Maksimal kedalaman tree
    criterion='gini'       # Kriteria pemilihan fitur terbaik
)

model.fit(X_train, y_train)
predictions = model.predict(X_test)

accuracy = accuracy_score(y_test, predictions)
print("Accuracy:", accuracy)
```

Figure 2. DT pseudocode

2.3. Extreme Gradient Boosting (XGBoost)

XGBoost is known for its ability to handle large and complex datasets, handle imbalanced data, and perform predictions with high accuracy [30]. In the context

of time-based crime prediction, XGBoost iteratively builds many small decision trees, where each tree learns from the errors of the previous tree. This process helps the model improve predictions gradually by minimizing prediction errors. The next process is to create an XGBoost model. Optimize the model parameters with `n_estimator = 10`, which means the number of trees built in the boosting model is 10. Gradually, the trees built will correct the errors of the previous trees. The higher the value of `n_estimator`, the more trees are built, which can improve the model's ability to learn. data patterns but also increases the risk of overfitting if not properly controlled.

`Learning_rate = 0.01`, which means controlling the learning speed of the model in each iteration or tree. A value of 0.01 indicates that the model will learn very slowly, with each added tree only contributing a small contribution to the final prediction. A low learning rate often helps the model become more stable and avoid overfitting, but it requires more iterations or trees (`n_estimators`) to achieve optimal performance. Figure 3 is the pseudocode of XGBoost.

```
import xgboost as xgb
model = xgb.XGBClassifier(
    max_depth=5,
    learning_rate=0.1,
    n_estimators=100,
    objective='binary:logistic'
)
model.fit(X_train, y_train)
predictions = model.predict(X_test)
accuracy = accuracy_score(y_test, predictions)
print("Accuracy:", accuracy)
```

Figure 3. XGBoost pseudocode

2.4. Categorical Boosting (CatBoost)

CatBoost is a machine learning algorithm that is very efficient in handling categorical data and is very suitable for use in time-based crime prediction analysis [31], [32] or crime clock. A crime clock aims to predict when a crime is likely to occur based on historical crime patterns, time, hours, minutes, and seconds. CatBoost can handle many categorical variables that often appear in crime data, such as crime type, region, or day of the week, without the need for complicated data preprocessing [26].

Iterations = 100 refers to the number of trees or boosting rounds the model will build. In this case, the model will build 100 trees. A learning rate of 0.1 controls how much the model learns on each iteration. A value of 0.1 indicates that the model will learn moderately (not too fast and not too slow) from the errors on each iteration. Depth=6: The trees built have a maximum depth of 6, which

provides a balance between capturing fairly complex patterns without overfitting. Figure 4 is the CatBoost pseudocode.

```

▶ from catboost import CatBoostClassifier
  model = CatBoostClassifier(
    iterations=100,
    learning_rate=0.1,
    depth=5,
    loss_function='Logloss'
  )

  model.fit(X_train, y_train)
  predictions = model.predict(X_test)
  accuracy = accuracy_score(y_test, predictions)
  print("Accuracy:", accuracy)

```

Figure 4. CatBoost pseudocode

2.5. Mean Absolute Percentage Error (MAPE)

It is a measure of accuracy used to measure how large the average absolute error is between the actual value and the predicted value, expressed as a percentage. MAPE is very useful in various prediction models because it provides an intuitive idea of how far the prediction is from the actual value [33], [34].

Eval_metric = 'mape' refers to the evaluation metric used to measure model performance during training and validation. Mape stands for Mean Absolute Percentage Error, which measures the average absolute percentage error between the prediction and the actual value. The MAPE value is displayed as a percentage, providing a measure of how large the average model prediction error is in percentage terms. A smaller MAPE value indicates a more accurate model, as it shows that the average model prediction error is relatively small compared to the actual value, as shown in Equation 1 [35]–[37].

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (1)$$

A_t is the actual value at t

F_t predicted value at time t

n number of observations

The flow of this research is represented in Figure 5, which begins with the data collection process, followed by the data wrangling process consisting of gathering, assessing, manipulating, transforming, and splitting data. The next process is modeling with three machine learning algorithms to find the optimal MAPE, namely DT, XGBoost, and CatBoost. Then the next evaluation is the implementation of the prediction.

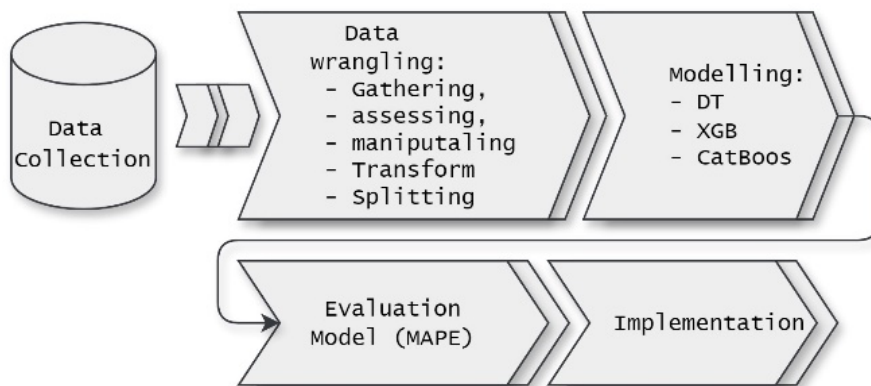


Figure 5. Research flow

3. RESULTS AND DISCUSSION

3.1 Data Wrangling

This wrangling process includes several important steps in preparing data for time series analysis or modeling [38], [39]. We load, assess, manipulate, transform into supervised form, and divide the data into training and testing sets. All of these steps are designed to produce clean, structured data that is ready to be used in time series prediction models. The data wrangling process goes through several stages, namely.

1) Load dataset

The first process in data wrangling is loading the dataset. In this case, the `all_data.csv` dataset is loaded using the `pd.read_csv()` function from Pandas and stored into the `df` variable. After that, `df.head()` is used to display the first 5 rows of the dataset to see the initial structure of the data and check the available columns. This stage is important to get an initial picture of the data. The results of loading the dataset are shown in Figure 6.

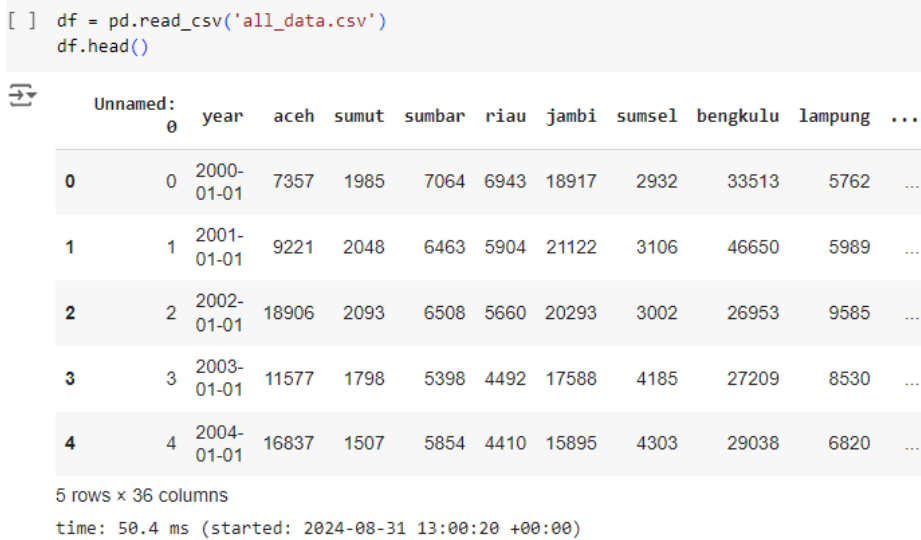


Figure 6. Initial data

2) Assessing and Manipulating

Next, the data is assessed by checking the dimensions of the dataset using `df.shape` and further information about the data type and the number of empty values with `df.info()` [33]. At this stage, the columns required for the analysis are selected, namely the year, jateng, jabar, jatim, sumut, and metro columns. Data containing only the selected columns are stored in the `lowest_df` variable to focus the analysis on the relevant data subset (Figure 7).

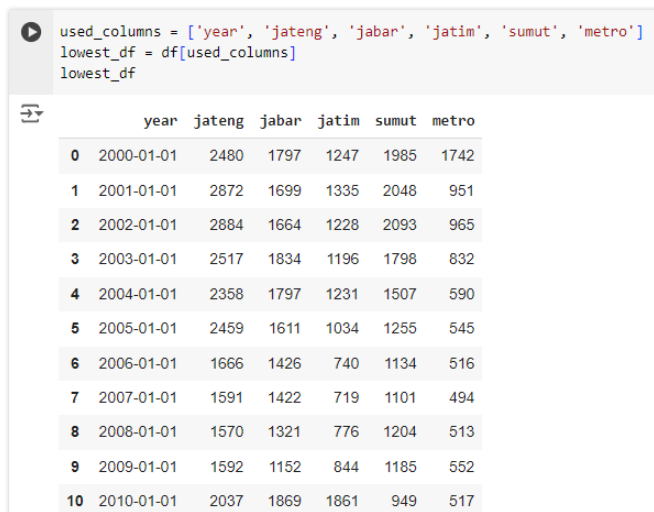
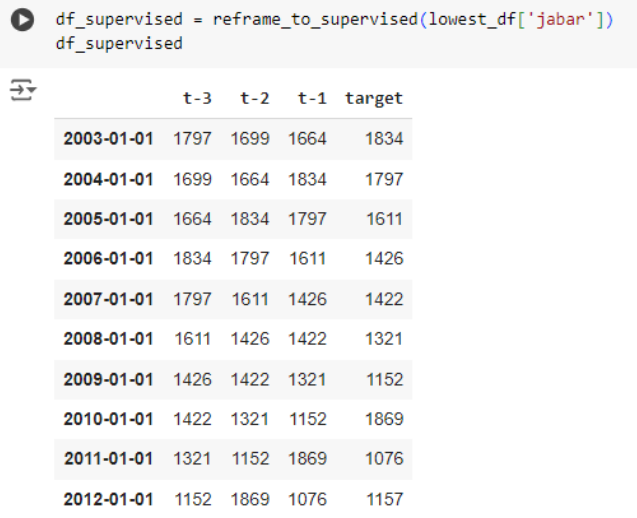


Figure 7. Five cities dataset

3) Transform Data

In the data transformation stage, the year column is converted into a date format using the `pd.to_datetime()` function, which is important to ensure that the time data is interpreted correctly. After that, year is used as an index of the DataFrame to facilitate time-based operations. Next, the data is transformed into a supervised form using the `reframe_to_supervised()` function. This function produces a new DataFrame with features in the form of lag values from previous times (e.g., t-3, t-2, t-1) and prediction targets, namely the values of the `jabar` column in the future (Figure 8). This is an important step in preparing data for time series models.



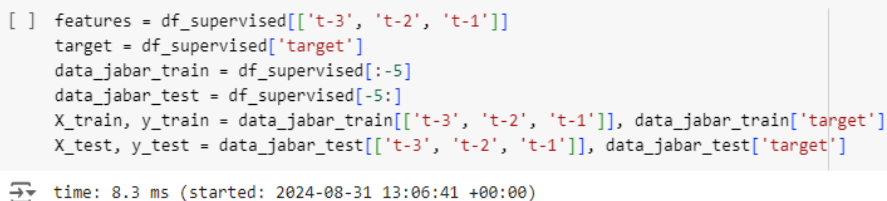
```
df_supervised = reframe_to_supervised(lowest_df['jabar'])
df_supervised
```

	t-3	t-2	t-1	target
2003-01-01	1797	1699	1664	1834
2004-01-01	1699	1664	1834	1797
2005-01-01	1664	1834	1797	1611
2006-01-01	1834	1797	1611	1426
2007-01-01	1797	1611	1426	1422
2008-01-01	1611	1426	1422	1321
2009-01-01	1426	1422	1321	1152
2010-01-01	1422	1321	1152	1869
2011-01-01	1321	1152	1869	1076
2012-01-01	1152	1869	1076	1157

Figure 8. DataFrame

4) Splitting Data

Once the data is prepared in supervised form, the next step is to split the data into training and testing sets. The data is separated based on the previous lag value to be used as features (t-3, t-2, t-1) and targets (target). The data is divided into `data_jabar_train` for training and `data_jabar_test` for testing, with the last row used for testing (Figure 9). This process allows for evaluation of model performance on unseen data.



```
[ ] features = df_supervised[['t-3', 't-2', 't-1']]
    target = df_supervised['target']
    data_jabar_train = df_supervised[:-5]
    data_jabar_test = df_supervised[-5:]
    X_train, y_train = data_jabar_train[['t-3', 't-2', 't-1']], data_jabar_train['target']
    X_test, y_test = data_jabar_test[['t-3', 't-2', 't-1']], data_jabar_test['target']
```

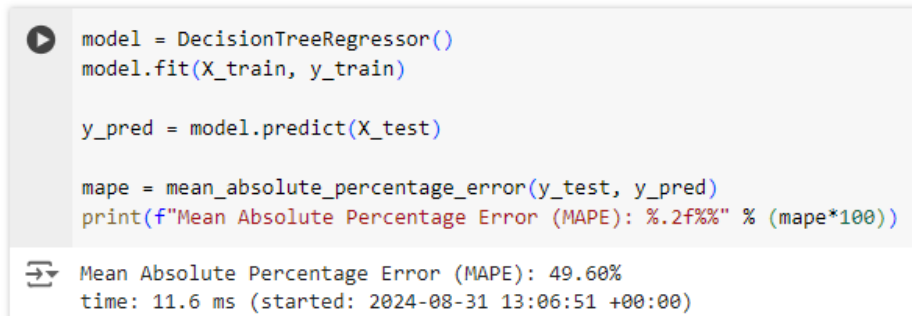
time: 8.3 ms (started: 2024-08-31 13:06:41 +00:00)

Figure 9. Code splitting data

3.2 Modelling

1) Decision Tree (DT)

The modeling process uses the DT algorithm; the model is evaluated with the Mean Absolute Percentage Error (MAPE) metric, which produces a value of 49.60% (Figure 10). This MAPE shows that the average model prediction error is around 49.60% of the actual value, which indicates that the model has a relatively low level of accuracy and tends to produce quite large errors in predicting target values. In terms of execution time, DT takes 11.6 ms to complete the training and prediction process, which is a very fast time. This indicates that although this model is not very accurate in prediction, in terms of speed and efficiency, the DT algorithm is very suitable for use in cases where fast predictions are needed, considering the fairly high error rate.

The image shows a code editor window with a light gray background. On the left, there is a vertical toolbar with a play button icon at the top and a refresh icon at the bottom. The code is written in a monospaced font with syntax highlighting: keywords are in blue, strings are in red, and comments are in green. The code defines a DecisionTreeRegressor model, fits it to training data, predicts on test data, and calculates the MAPE. The output at the bottom shows the MAPE as 49.60% and the execution time as 11.6 ms, with a timestamp indicating the start time was 2024-08-31 13:06:51 +00:00.

```
model = DecisionTreeRegressor()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mape = mean_absolute_percentage_error(y_test, y_pred)
print(f"Mean Absolute Percentage Error (MAPE): %.2f%%" % (mape*100))
```

Mean Absolute Percentage Error (MAPE): 49.60%
time: 11.6 ms (started: 2024-08-31 13:06:51 +00:00)

Figure 10. West Java DT Model

2) XGBoost

In the modeling process using XGBoost, the prediction on the test data produces a predicted value of 1848.33, while the actual value is 1060 (Figure 11). There is a significant difference between the actual and predicted values, indicating that the model is not accurate enough in capturing the pattern of the data in this case.

In terms of execution time, the prediction process using XGBoost takes 50.9 ms, which is relatively fast considering the complexity of the XGBoost algorithm, which is generally higher than other methods such as Decision Tree. However, this speed is not comparable to the level of prediction accuracy, which can be seen from the large difference between the actual and predicted values. This prediction was carried out on 2022-01-01, indicating that the XGBoost model needs to be further improved, either through hyperparameter tuning or with more features or training data in order to produce more accurate predictions according to the actual values.

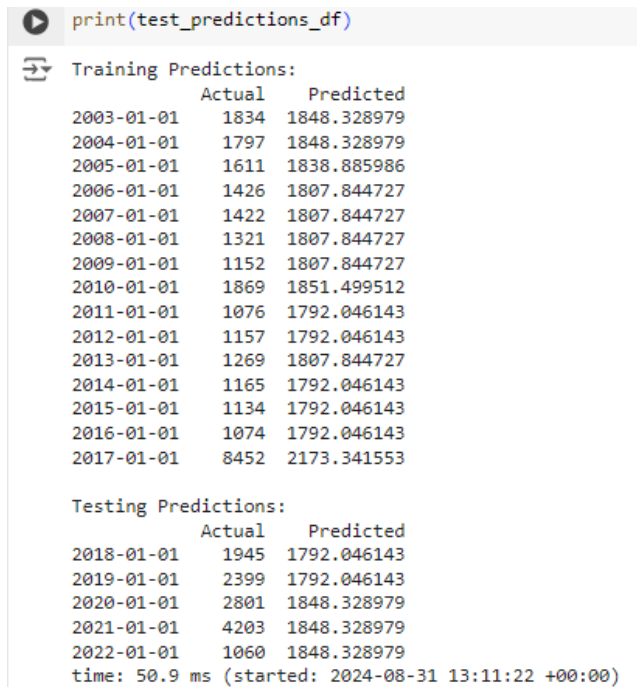


Figure 11. XGBoost model results

3) CatBoost

In the modeling process using CatBoost, the model produces a Mean Absolute Percentage Error (MAPE) value of 40.40% (Figure 12). This shows that the average model prediction error is around 40.40% of the actual value, which indicates better performance compared to the Decision Tree model used previously but is still quite high for precise prediction needs. Although the prediction error is not small, CatBoost shows superiority in capturing more complex data patterns compared to several other methods.

In terms of speed, the training and prediction process with CatBoost takes 160 ms, which is slightly slower than several other algorithms such as Decision Tree or XGBoost. However, considering the complexity and ability of CatBoost to handle categorical data and irregular data, this time is still very efficient.

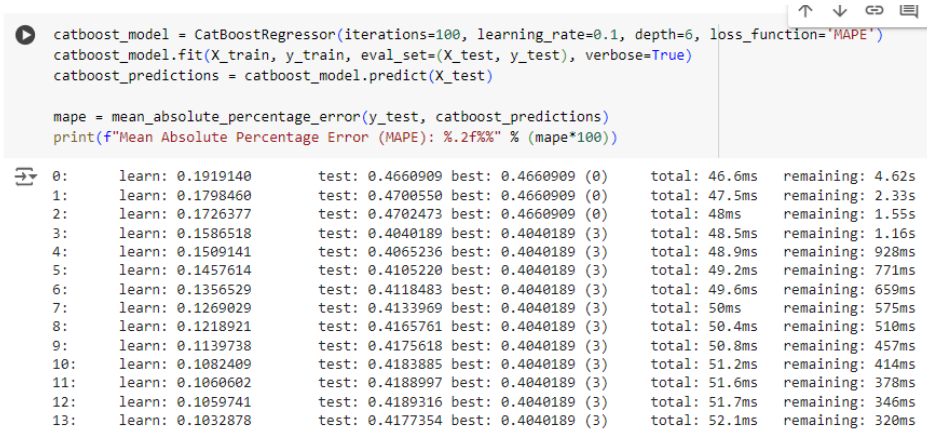


Figure 12. CatBoost model results

3.3 Mean Absolute Percentage Error (MAPE)

In the process of evaluating model performance using three different algorithms, Decision Tree, XGBoost, and CatBoost, the trial in the West Java area (Figure 13) showed significant differences in the level of accuracy and prediction performance, as measured by MAPE.

- 1) Decision Tree produces a MAPE of 49.60%, indicating that this model has a fairly high error rate. With an average error of almost 50%, the Decision Tree model tends to be less accurate in predicting target values. However, the advantage of Decision Tree is its very fast execution time, which is 17.6 ms, making it an efficient choice for fast predictions even with low accuracy.
- 2) XGBoost shows the best performance with MAPE of 39.51%, providing more accurate predictions than other models. XGBoost is able to capture more complex data patterns, thus reducing prediction errors significantly. Although the execution time is slightly slower than Decision Tree, XGBoost still offers good efficiency with a relatively short execution time.
- 3) CatBoost produces a MAPE of 40.40%, which is slightly higher than XGBoost but still much better than Decision Tree. CatBoost is known to be good at handling categorical data and has an advantage in situations where categorical features dominate. With a fairly fast execution time of 17.6 ms, CatBoost provides a balance between accuracy and speed.

```
# Decision Tree
y_pred_dt = model.predict(X_test)
mape_dt = mean_absolute_percentage_error(y_test, y_pred_dt)
print(f"Decision Tree MAPE: %.2f%%" % (mape_dt*100))

# XGBoost
y_pred_xgb = xgb_model.predict(X_test)
mape_xgb = mean_absolute_percentage_error(y_test, y_pred_xgb)
print(f"XGBoost MAPE: %.2f%%" % (mape_xgb*100))

# CatBoost
y_pred_catboost = catboost_model.predict(X_test)
mape_catboost = mean_absolute_percentage_error(y_test, y_pred_catboost)
print(f"CatBoost MAPE: %.2f%%" % (mape_catboost*100))
```

Decision Tree MAPE: 49.60%
XGBoost MAPE: 39.51%
CatBoost MAPE: 40.40%
time: 17.6 ms (started: 2024-08-31 13:12:32 +00:00)

Figure 13. Results of the evaluation model

3.4 Implementation

At this stage, predictions are made for 5 future periods using the previously trained XGBoost model. The following are the implementation steps taken:

- 1) **Creating Future Dates**
The prediction time period is determined using `pd.date_range()` from January 1, 2022, with annual frequency (YS) for the next 5 periods. These dates are then stored in a DataFrame `future_df` with columns corresponding to the lag features ($t-3$, $t-2$, $t-1$), which are used as inputs for the model prediction.
- 2) **Initialize Initial Values**
In the first iteration, the values in the first row of `future_df` are initialized with the last value of the test data (`X_test`). This is done so that the model has an initial basis for making predictions based on the last value of the existing data set.
- 3) **Prediction Calculation for Next Period**
For subsequent periods, the lag values are automatically updated from the previous predictions. At each iteration, the values in column $t-1$ are filled with the prediction results from the XGBoost model. Each prediction is made using the updated features from the previous period, so the model can continue to predict future periods sequentially.
- 4) **Prediction Results**
After the prediction process is complete, the model produces a constant prediction of 866.16174 for each period. This could indicate that the model may have overfit or not learned enough from the existing data to provide

- variation in future predictions. The prediction results are stored in an array as follows: Array ([866.16174, 866.16174, 866.16174, 866.16174], dtype=float32)
- 5) Execution Time
This prediction process is very fast, taking only 18.7 ms, which shows the efficiency of the XGBoost model in handling predictions even for multiple future periods.

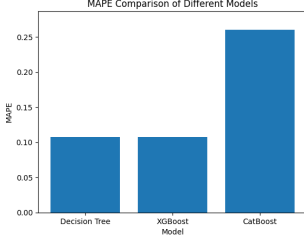
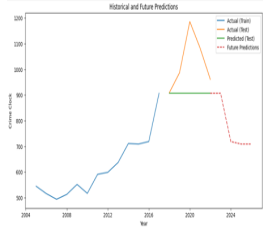
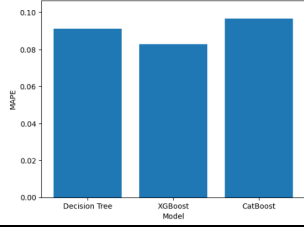
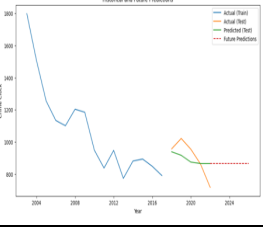
With the mean function, the five areas that are most prone to crime are obtained, shown in table 1. T0068e Metro Jaya area experiences a crime every 8' minutes, 14'' seconds.

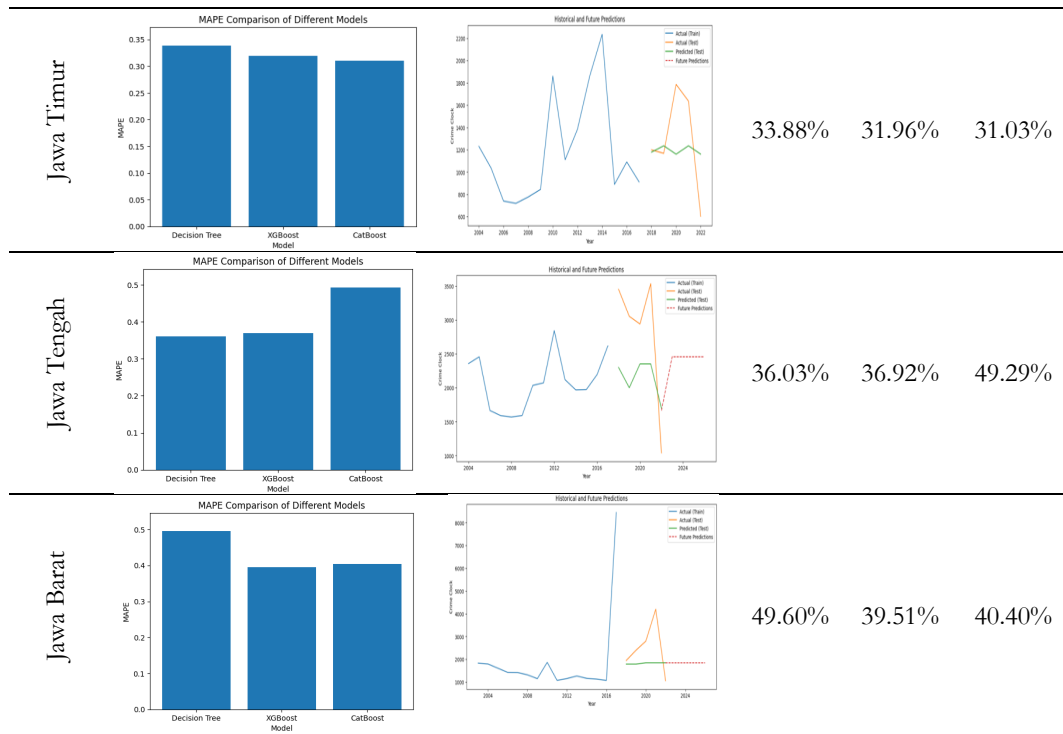
Table 1. Most crime-prone areas

Area	Crime Clock
Metro Jaya	00.08'.14''
Sumatera Utara	00.11'.58''
Jawa Timur	00.10'.02''
Jawa Tengah	00.17'.20''
Jawa Barat	00.17'.40''

The next process is modeling using three algorithms, DT, XGBoost, and CatBoost, for the five regions, the performance as shown in Table 2.

Table 2. Prediction results in five regions

Area	MAPE	Prediction	DT	XGB	CB
Metro Jaya			10.74%	10.75%	26.03%
Sumatera Utara			9.14%	8.29%	9.68%



In the process of evaluating the prediction model for the North Sumatra region, three main algorithms DT, XGBoost, and CatBoost are used to predict the target value, and the model performance is measured using the mean absolute percentage error (MAPE) with the most optimal results compared to other areas. The following are the evaluation results and interpretations of each model:

1) Decision Tree:

a) MAPE: 9,14%

The DT model produces an error rate of 9.14%, which means that on average the model predictions deviate by about 9.14% from the actual values. This result is quite good and shows that the model is able to capture data patterns quite accurately.

b) Execution Speed:

The DT algorithm requires a very short time to make predictions and overall can be relied on for fast predictions with adequate results in the North Sumatra region.

2) XGBoost

a) MAPE: 8.29%

The XGBoost model shows the best performance with the lowest MAPE among all models, which is 8.29%. This shows that XGBoost is able to predict the target value with a better accuracy rate than other models, with the smallest prediction deviation from the actual value..

- b) Execution Speed:
Although more complex, XGBoost can still provide very good results in an efficient time, making it the best choice for predictions in the North Sumatra region considering the balance between accuracy and execution time.
- 3) CatBoost
 - a) MAPE: 9.68%
CatBoost produces a MAPE of 9.68%, slightly higher than XGBoost and DT. Although still within a fairly good limit, CatBoost is less accurate than XGBoost in this case. However, this model still has the advantage of handling categorical data efficiently.
 - b) Execution Speed:
CatBoost execution time is also quite fast, taking only 22 ms, on par with other models, and providing adequate results for the North Sumatra region.

In the evaluation of three main models, DT, XGBoost, and CatBoost, in predicting crime in North Sumatra, XGBoost proved to be superior with a MAPE of 8.29% compared to other models. The superior performance of XGBoost can be explained by several factors:

- 1) XGBoost uses gradient boosting that iteratively corrects errors in previous models, allowing the model to learn more effectively from complex and feature-rich data. This gives XGBoost an advantage in understanding complex feature interactions, which may not be fully optimized by DT and CatBoost.
- 2) XGBoost has good missing values and regularization handling capabilities, which prevent overfitting, especially in complex and dynamic data such as crime, which has many causal factors.
- 3) Hyperparameter tuning in XGBoost provides better control over how the model learns from the data. Parameters such as `learning_rate`, `n_estimators`, and `max_depth` in XGBoost have been optimized to strike a balance between underfitting and overfitting. While DT, which has a simpler structure, tends to overfit the data easily, resulting in decreased performance.
- 4) Crime data is often uneven, with incidents not occurring evenly across time periods. XGBoost excels at handling imbalanced data through boosting techniques that can give more attention to hard-to-predict samples, correcting weaknesses in previous iterations of the model.

More accurate predictions from XGBoost can be practically applied to optimize law enforcement resources in high-crime areas. Some implementation steps that can be taken are:

- 1) With more accurate predictions, police can allocate personnel and resources to higher-risk areas or times, thereby increasing efficiency in crime prevention.

- 2) The crime clock predicted by the model can help adjust patrol schedules more precisely, so that surveillance is more effective at times when crime tends to increase, based on historical analysis studied by XGBoost.
- 3) Based on the prediction results, law enforcement can identify hotspots for more optimal resource placement. This allows for increased security in high-risk areas without having to spread resources evenly.

For other regions with higher MAPE, improvements can be made to model performance, such as Performing deeper optimization of XGBoost hyperparameters or using Bayesian optimization or random search can help find the best combination for regions with different crime characteristics. Using ensembling techniques, namely combining several models such as XGBoost, CatBoost, and Random Forest, can help improve accuracy. Each model may excel in certain aspects, and with ensembling, predictions can be more balanced and robust.

3.5. Discussion

This study evaluated the predictive accuracy of three machine learning algorithms—Decision Tree (DT), XGBoost, and CatBoost—in determining crime time intervals based on historical data. The results demonstrated the superior performance of XGBoost with the lowest Mean Absolute Percentage Error (MAPE) value of 8.29%, followed by DT with 9.14% and CatBoost with 9.68%. This section critically analyzes the findings, identifies their implications, and suggests pathways for practical implementation and future research.

XGBoost's superior accuracy can be attributed to its gradient boosting mechanism, which iteratively reduces prediction errors by building successive models that correct the weaknesses of prior models. This approach allows XGBoost to handle complex data patterns, particularly irregular temporal variations in crime datasets, with greater precision. Moreover, XGBoost's handling of missing values and regularization techniques minimizes overfitting, making it a robust choice for crime prediction. In contrast, DT, while computationally efficient, showed a higher MAPE due to its simpler structure and tendency to overfit, particularly when the dataset's complexity increases. Although CatBoost exhibited slightly higher errors than XGBoost, its specialized ability to handle categorical data efficiently makes it a viable option for datasets containing categorical variables like crime type or day of the week.

The results have practical implications for law enforcement agencies, particularly in high-crime regions like Metro Jaya and North Sumatra, where more accurate predictions can optimize resource allocation. With XGBoost's predictive capabilities, authorities can strategically deploy personnel and resources to areas and times with a higher likelihood of criminal activity. This proactive approach not

only enhances operational efficiency but also improves public safety. Moreover, the predicted "crime clock" can aid in adjusting patrol schedules and law enforcement strategies. For example, areas like Metro Jaya, where crimes occur approximately every 8 minutes, could benefit from increased surveillance during peak crime intervals. The integration of these predictive insights into decision-making processes could significantly enhance crime prevention efforts.

While XGBoost outperformed other models, certain regions, such as Central and East Java, reported higher MAPEs, indicating room for improvement. Possible enhancements include:

- 1) Hyperparameter Optimization: Advanced techniques like Bayesian optimization or grid search could fine-tune model parameters, potentially reducing prediction errors in regions with high MAPEs.
- 2) Incorporating Additional Features: Socio-economic, demographic, and infrastructure data could provide a more comprehensive understanding of crime patterns, thereby improving model accuracy.
- 3) Ensemble Techniques: Combining XGBoost, CatBoost, and other algorithms like Random Forest may create a more robust predictive framework by leveraging the strengths of multiple models.
- 4) Temporal Resolution: Refining the granularity of temporal data (e.g., hourly or monthly trends) could help the models better capture localized crime patterns.

Despite its potential, the use of machine learning in crime prediction raises ethical concerns, particularly regarding algorithmic bias and data privacy. Biased datasets may lead to skewed predictions that disproportionately target certain communities, exacerbating social inequalities. Additionally, the collection and analysis of personal data must comply with privacy regulations to avoid infringement on individual rights. Future studies should address these issues by incorporating fairness metrics and ensuring transparency in model development and implementation.

To build on this research, future studies could explore integrating real-time data sources, such as emergency calls and social media trends, to improve prediction timeliness. Expanding the scope to include international crime datasets could also validate the generalizability of these models. Finally, testing the models in a live implementation setting would provide valuable insights into their practical applicability and effectiveness in reducing crime rates. This study highlights the potential of machine learning algorithms, particularly XGBoost, in predicting crime time intervals with high accuracy. By addressing current limitations and ethical considerations, the findings can pave the way for data-driven law enforcement strategies that enhance public safety and resource management.

4. CONCLUSION

The use of the XGBoost algorithm in predicting crime clocks in the North Sumatra region proved to be the most optimal, with a MAPE of 8.29%, indicating high prediction accuracy. With more accurate predictions, law enforcement can allocate resources more efficiently to prevent crime, allowing for a proactive approach in dealing with increasing crime. These results provide deeper insight into crime patterns and have the potential to increase the effectiveness of regional security strategies, thereby creating a safer environment for the community. Further studies can integrate other data, such as socio-economic, weather, and infrastructure data, to improve prediction accuracy and provide deeper insights into the factors that influence crime.

ACKNOWLEDGMENT

This research is supported by the Directorate of Research, Technology, and Community Service (DRTPM) of the Ministry of Education, Culture, Research and Technology through Contract No. 761/LL3/AL.04/2024, 003/LP3M/Kontrak/VI/2024.

REFERENCES

- [1] W. Safat, S. Asghar, and S. A. Gillani, "Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques," *IEEE Access*, vol. 9, pp. 70080–70094, 2021, doi: 10.1109/ACCESS.2021.3078117.
- [2] U. M. Butt, S. Letchmunan, F. H. Hassan, M. Ali, A. Baqir, and H. H. R. Sherazi, "Spatio-Temporal Crime HotSpot Detection and Prediction: A Systematic Literature Review," *IEEE Access*, vol. 8, pp. 166553–166574, 2020, doi: 10.1109/ACCESS.2020.3022808.
- [3] T. Podzolkova, I. Shynkarenko, and P. Sergii, "Use of Information Systems in Disclosure of Criminal Offenses," in *Integrated Computer Technologies in Mechanical Engineering*, pp. 482–497, 2023.
- [4] C. M. Ruiz-Paz, "Crime Analysis in an International Context," in *The Crime Analyst's Companion*, M. Bland, B. Ariel, and N. Ridgeon, Eds. Cham: Springer International Publishing, 2022, pp. 21–39. doi: 10.1007/978-3-030-94364-6_3.
- [5] R. Yulianto *et al.*, "Innovative UNET-Based Steel Defect Detection Using 5 Pretrained Models," vol. 10, no. 04, pp. 2365–2378, 2023.
- [6] M. Saraiva, I. Matijošaitienė, S. Mishra, and A. Amante, "Crime Prediction and Monitoring in Porto, Portugal, Using Machine Learning, Spatial and Text Analytics," *ISPRS Int. J. Geo-Information*, vol. 11, no. 7, 2022, doi: 10.3390/ijgi11070400.
- [7] H. Zimmermann, A., Di Rosa, E., & Kim, "Technology Can 't Fix Algo r

- ithmic I nj us tice,” *Bost. Rev.*, pp. 1–13, 2020.
- [8] F. Ding, M. Hardt, J. Miller, and L. Schmidt, “Retiring Adult: New Datasets for Fair Machine Learning,” in *Advances in Neural Information Processing Systems*, vol. 34, pp. 6478–6490, 2021.
 - [9] F. Durante and S. T. Fiske, “How social-class stereotypes maintain inequality,” *Curr. Opin. Psychol.*, vol. 18, pp. 43–48, 2017, doi: 10.1016/j.copsyc.2017.07.033.
 - [10] R. Mühlhoff, “Predictive privacy: towards an applied ethics of data analytics,” *Ethics Inf. Technol.*, vol. 23, no. 4, pp. 675–690, 2021, doi: 10.1007/s10676-021-09606-x.
 - [11] Vengadeswaran, D. Binu, and L. Rai, “An Efficient Framework for Crime Prediction Using Feature Engineering and Machine Learning,” in *Advances in Data and Information Sciences*, 2024, pp. 49–59.
 - [12] R. de V. dos Santos, J. V. V. Coelho, N. A. A. Cacho, and D. S. A. de Araújo, “A criminal macrocause classification model: An enhancement for violent crime analysis considering an unbalanced dataset,” *Expert Syst. Appl.*, vol. 238, p. 121702, 2024, doi: 10.1016/j.eswa.2023.121702.
 - [13] N. O. Edoaka, *Crime incidents classification using supervised machine learning techniques: Chicago*, Doctoral dissertation, National College of Ireland, Dublin, 2020.
 - [14] Z. Yan, H. Chen, X. Dong, K. Zhou, and Z. Xu, “Research on prediction of multi-class theft crimes by an optimized decomposition and fusion method based on XGBoost,” *Expert Syst. Appl.*, vol. 207, p. 117943, 2022, doi: 10.1016/j.eswa.2022.117943.
 - [15] R. N. Zulfahmi, M. K. Daul, M. Al Ayyubi, I. W. Julianta, Pradnyana, and R. Dwi Bakti, “Pemetaan Kerentanan Tingkat Kriminalitas Menggunakan Metode Self Organizing Map,” *INSOLOGI J. Sains dan Teknol.*, vol. 2, no. 5, pp. 872–881, 2023, doi: 10.55123/insologi.v2i5.2566.
 - [16] D. M, H. A. S, and M. Meleet, “Crime Prediction and Forecasting using Voting Classifier,” in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2021, pp. 1–5. doi: 10.1109/ICECCT52121.2021.9616911.
 - [17] A. Alfaries, H. Mengash, A. Yasar, and E. Shakshuki, Eds., *Advances in Data Science, Cyber Security and IT Applications: First International Conference on Computing, ICC 2019, Riyadh, Saudi Arabia, December 10–12, 2019, Proceedings, Part II*, vol. 1098. Springer Nature, 2019.
 - [18] S. Moeinazade and G. Hu, “Predicting Metropolitan Crime Rates Using Machine Learning Techniques,” in *Smart Service Systems, Operations Management, and Analytics*, 2020, pp. 77–86.
 - [19] S. A. Ajagbe, J. B. Oladosu, and A. O. Adesina, “Accuracy of Machine Learning Models for Mortality Rate Prediction in a Crime Dataset Development of a web based aboriginal virtual patient system for training medical students View project Physics Electronics View project,” *Researchgate.Net*, vol. 10, no. April 2021, pp. 150–160, 2020.

- [20] M. H. Chee, "Crime rate prediction using machine learning," Universiti Tunku Abdu Rahman, 2022.
- [21] H. K. Sharma, T. Choudhury, and A. Kandwal, "Machine learning based analytical approach for geographical analysis and prediction of Boston City crime using geospatial dataset," *GeoJournal*, vol. 88, no. 1, pp. 15–27, 2023, doi: 10.1007/s10708-021-10485-4.
- [22] A. A. Almuhanha, M. M. Alrehili, S. H. Alsubhi, and L. Syed, "Prediction of Crime in Neighbourhoods of New York City using Spatial Data Analysis," in *2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA)*, 2021, pp. 23–30. doi: 10.1109/CAIDA51941.2021.9425120.
- [23] L. Zhang and D. Jánošík, "Enhanced short-term load forecasting with hybrid machine learning models: CatBoost and XGBoost approaches," *Expert Syst. Appl.*, vol. 241, p. 122686, 2024, doi: 10.1016/j.eswa.2023.122686.
- [24] S. Hussain *et al.*, "A novel feature engineered-CatBoost-based supervised machine learning framework for electricity theft detection," *Energy Reports*, vol. 7, pp. 4425–4436, 2021, doi: 10.1016/j.egyr.2021.07.008.
- [25] R. S. Mohril, B. S. Solanki, M. S. Kulkarni, and B. K. Lad, "XGBoost based residual life prediction in the presence of human error in maintenance," *Neural Comput. Appl.*, vol. 35, no. 4, pp. 3025–3039, 2023, doi: 10.1007/s00521-022-07216-2.
- [26] J. T. Hancock and T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *J. Big Data*, vol. 7, no. 1, p. 94, 2020, doi: 10.1186/s40537-020-00369-8.
- [27] M. S. K. Luu, S. Banerjee, E. N. Pavlovskiy, and B. N. Tuchinov, "Harnessing Ensemble Machine Learning Models for Timely Diagnosis of Breast Cancer Metastasis: A Case Study on CatBoost, XGBoost, and LGBM," in *2024 IEEE 25th International Conference of Young Professionals in Electron Devices and Materials (EDM)*, 2024, pp. 2320–2325. doi: 10.1109/EDM61683.2024.10615210.
- [28] bps.go.id, "Selang Waktu Terjadinya Kejahatan (Crime Clock)," *Badan pusat Statistik*, 2023.
- [29] I. Rahmatillah, E. Astuty, and I. D. Sudirman, "An Improved Decision Tree Model for Forecasting Consumer Decision in a Medium Groceries Store," in *2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS)*, 2023, pp. 245–250. doi: 10.1109/ICIIS58898.2023.10253592.
- [30] Y. Zhou, X. Song, and M. Zhou, "Supply Chain Fraud Prediction Based On XGBoost Method," in *2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, 2021, pp. 539–542. doi: 10.1109/ICBAIE52039.2021.9389949.
- [31] G. S. Kumar and R. Dhanalakshmi, "Performance Analysis of CatBoost Algorithm and XGBoost Algorithm for Prediction of CO2Emission Rating," in *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, 2023, vol. 6, pp. 1497–1501. doi:

- 10.1109/IC3I59117.2023.10398160.
- [32] S. Wang, "Comparison and Analysis of the Effect of XGBoost Classification, BP Neural Network Classification and CatBoost Classification on Malware Attack Prediction," in *2023 International Conference on Intelligent Communication and Computer Engineering (ICICCE)*, 2023, pp. 77–80. doi: 10.1109/ICICCE61720.2023.00018.
- [33] Y. Tani, A. Kobayashi, K. Masai, T. Fukuda, M. Sugimoto, and T. Kimura, "Assessing Individual Decision-Making Skill by Manipulating Predictive and Unpredictive Cues in a Virtual Baseball Batting Environment," in *2023 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2023, pp. 775–776. doi: 10.1109/VRW58643.2023.00230.
- [34] Al-Khowarizmi, S. Efendi, M. K. M. Nasution, and M. Herman, "The Role of Detection Rate in MAPE to Improve Measurement Accuracy for Predicting FinTech Data in Various Regressions," in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, 2023, pp. 874–879. doi: 10.1109/ICCoSITE57641.2023.10127820.
- [35] S. Y. Heng *et al.*, "Artificial neural network model with different backpropagation algorithms and meteorological data for solar radiation prediction," *Sci. Rep.*, vol. 12, no. 1, p. 10457, 2022, doi: 10.1038/s41598-022-13532-3.
- [36] Ü. Ağbulut, "A novel stochastic model for very short-term wind speed forecasting in the determination of wind energy potential of a region: A case study from Turkey," *Sustain. Energy Technol. Assessments*, vol. 51, p. 101853, 2022, doi: 10.1016/j.seta.2021.101853.
- [37] L. Ding, Y. Bai, M.-D. Liu, M.-H. Fan, and J. Yang, "Predicting short wind speed with a hybrid model based on a piecewise error correction method and Elman neural network," *Energy*, vol. 244, p. 122630, 2022, doi: 10.1016/j.energy.2021.122630.
- [38] N. Shrestha, B. Chopra, A. Z. Henley, and C. Parnin, "Detangler: Helping Data Scientists Explore, Understand, and Debug Data Wrangling Pipelines," in *2023 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2023, pp. 189–198. doi: 10.1109/VL-HCC57772.2023.00031.
- [39] Z. Zhang, P. Groth, I. Calixto, and S. Schelter, "Directions Towards Efficient and Automated Data Wrangling with Large Language Models," in *2024 IEEE 40th International Conference on Data Engineering Workshops (ICDEW)*, 2024, pp. 301–304. doi: 10.1109/ICDEW61823.2024.00044.