# Synthetic Data Generation Using Time-Generative Adversarial Network (Time-GAN) to Predict Cash ATM

2023 International Conference on Computer, Control, Informatics and its Applications (IC3INA) | 979-8-3503-9487-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/IC3INA60834.2023.10285809

Feri Ranja
*Department of Data Science and Artificial Intelligence*
*University of Sumatera Utara*
Medan, Indonesia
feriranja@students.usu.ac.id

Erna Budiarti Nababan*
*Department of Data Science and Artificial Intelligence*
*University of Sumatera Utara*
Medan, Indonesia
ernabrn@usu.ac.id

Ade Candra
*Department of Information Technology*
*University of Sumatera Utara*
Medan, Indonesia
ade_candra@usu.ac.id

***Abstract*— One of the particular concerns of the Bank for ATM transaction services is the availability of cash at ATMs. Prediction of the availability of ATM money is required by the Bank to manage funds optimally. This study aims to analyze and predict cash availability at ATM machines using Time-GAN and Extreme Gradient Boosting (XGBoost). Time-Series data is highly dependent on the size and consistency of the dataset used in the training. The features available in the dataset are limited and have constraints such as missing dimensions or missing values. Therefore, synthetic data generation technique is used as an effective way to increase the amount of data and handle imbalanced data. Synthetic data generation has been shown to increase the generalizability of models with Time-Series data. The generated data will be divided into Training data, Validation data, and Testing data, resulting in a Load Model that will be analyzed using the XGBoost method. The ultimate goal of this research is to provide a summary of the evaluation and performance that results in better ATM availability for future research. Model performance is evaluated with the Mean Absolute Error (MAE) metric 2.57 value, Mean Squared Error (MSE) 1.64 value, and R-squared 5.02 value.**

***Keywords—Synthetic data generation, Time-GAN, XGBoost***

## I. INTRODUCTION

Predicting the availability of cash ATMs is needed to manage funds optimally. The Bank estimates that there are several things that affect the number of cash withdrawal transactions at ATMs. First, ATMs located in public areas tend to have more transactions on holidays compared to ATMs located at city/district government offices. Second, the number of cash withdrawals also increases on the date of salary payments. ATM transaction data is represented as time- series data. Time-Series data is highly dependent on the size and consistency of the dataset used in the training. The features available in the dataset are limited and have constraints such as missing dimensions or missing values. Therefore, synthetic data generation technique is used as an effective way to increase the amount of data.

Different approaches have been researched in forecasting ATM cash demand; among them, the most commonly used methods are statistical modeling (ARIMA, VARMAX, etc) [1] , machine learning algorithms (MLP, SVR, GRNN), and hybrid methods [2]

Riabykh et.al [3] employed CUSUM algorithm to detect periodic anomalies (e.g. mass payment days, holidays. etc.), proposed a novel automated data preprocessing pipeline for all real data peculiar properties with new fine-tuning hyperparameter scheme, and compared the forecasting LSTM, SARIMA,RF, MLP and CNN model results with the local model approach.

ANN is employed by Serengil [4] to predict ATM cash flow andto optimize ATM replenishment. The data was taken from 6,500 ATM data in Turkey for 5 years. They used 29 features with a customized scale. The research shows that ATM cash withdrawals can be predicted based on seasonal trends.

Time-Gan is one synthetic data generation technique used to overcome imbalanced data. The study conducted by Yoon and Jarret [5] created a new framework for generating realistic time-series data that incorporates the flexibility of unsupervised paradigm with control provided through supervised training using time-series generative adversarial networks (time-GAN). Co-learned embedding is optimized with both supervised and adversarial objectives, by encouraging the network to adhere to the dynamics of the training data during sampling. Empirically, researchers evaluated the ability of the method to generate realistic samples using various real and synthetic time-series datasets. Qualitatively and quantitatively creating a framework that is consistently and significantly state-of-the-art benchmarks with respect to measures of similarity and predictive ability.

Meanwhile Nababan [6] used Biased Support Vector Machine (BSVM)and weighted-SMOTE to handle class imbalance problem. Non Support Vector (NSV) sets from negative samples and Support Vector (SV) sets from positive samples will undergo a Weighted-SMOTE process. The results indicate that implementation of Biased Support Vector Machine and Weighted-SMOTE achieve better accuracy and sensitivity.

XGBoost has been tested and evaluated by Osman et.al [7] compared with ANN and SVR learning models to predict an accurate groundwater levels prediction model using machine learning algorithms in Selangor, Malaysia. The models are developed using 11 months of previously recorded data of rainfall, temperature and evaporation to predict groundwater levels. XGBoost model outperformed both the Artificial Neural Network and Support Vector Regression models for all different input combinations.

SMOTE method with XGBoost has also given better performance in air pollution prediction [8]. The dataset was collected by the Jakarta Environmental Ministry in 2021. Using the XGBoost algorithm and SMOTE as based on the Air Pollution Standard Index (ISPU) category, the study uncovered factors that affect air quality.The proposed

classification model was evaluated using the Repeated k-fold cross-validation method. The results proved that SMOTE and XGBoost have better performance than using only the XGBoost method in predicting air quality.

XGBoost also employed by Prasetyo [9] to developed a web-based diabetes prediction application. The dataset was collected from Kaagle consisted of eight medical perdictors of features and one target variable, either diabetes or non-diabetes. The performance of trained model showed 74.67 % accuracy, 57.40% precision, 65.94% recall and 78.50% specificity.

Other works have studied the utility of XGBoost to construct machine learning models to classify risk level in the insurance industry in historical data [10]. XGBoost is used to handle missing values, resulting in 94% accuracy using data without imputation, 93% using imputation with the mean, and 93% using imputation with KNN.

This paper is organized into four sections. The first section is the introduction. It describes the background of the research, motivation, and related works. The second section discusses the methodology in detail, such as corpus building, data training, and the last is data testing. The third section is the result and discussion on the research output. The last section summarizes all the conducted research.

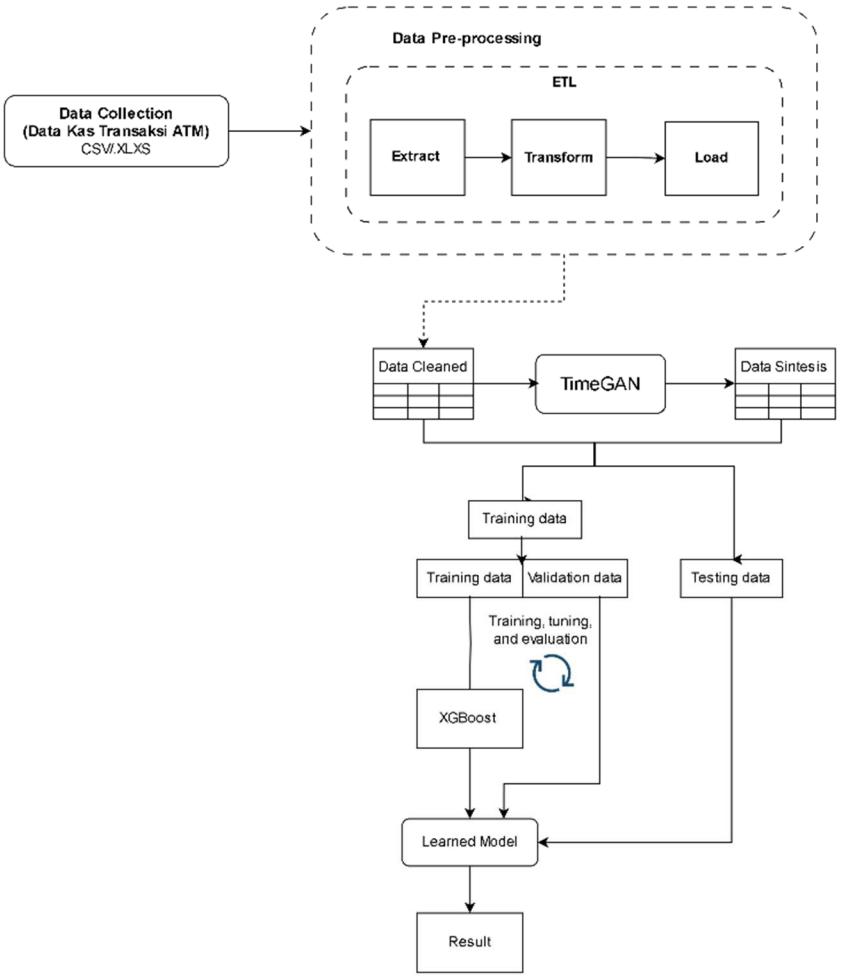


Fig. 1. General Architecture

## II. METHODOLOGY

Fig. 1 shows the general architecture of this research. It was conducted in several stages. The stages are divided into data collection, data preparation, original feature set, feature selection method, reduced feature set, data preprocessing, training, testing, model evaluation, and data preprocessing. method, reduced feature set, data preprocessing, training, testing, evaluation model, prediction model, and comparison model.

### *A. Data Collection*

Dataset used in this research is ATMs transaction data. Raw 353 ATMs withdrawal transaction data is obtained from the extraction of the transaction database at government bank which consists of 17 columns and 3932 rows.

TABLE I. RAW ATMS TRANSACTION DATA

| no | transaction date | transaction type | ccy | amount | stan | refrerence no | issuer | acquirer | destination acc no | destination bank | term id | term loc | rc | status | account no | card no |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 16/12/2021 12:14:01 | tarik tunai | idr | 2000000 | xx5272 | xx500xx | sumut | sumut | - | bank sumut | xxva01xx | atm kc | 0 | approved | xxx0204xxx | xxx4860010xxx |
| 3 | 16/12/2021 07:20:25 | tarik tunai | idr | 2000000 | xx2323 | xx500xx | sumut | sumut | - | bank sumut | xxva02xx | atm kc | 0 | approved | xxx0204xxx | xxx4860010xxx |
| 4 | 16/12/2021 07:00:53 | tarik tunai | idr | 2500000 | xx7306 | xx500xx | sumut | sumut | - | bank sumut | xxva02xx | atm kcp | 0 | approved | xxx0204xxx | xxx4860010xxx |
| 5 | 16/12/2021 07:01:38 | tarik tunai | idr | 2500000 | xx7308 | xx500xx | sumut | sumut | - | bank sumut | xxva02xx | atm kcp | 0 | approved | xxx0204xxx | xxx4860010xxx |

### *B. Data preparation with ETL*

In this research, the cleaning and transformation process is done by using ETL on both raw datasets. The raw datasets will be integrated into 1 main dataset. ETL is used to extract some data (columns) that will be used to become clean data, including removing duplicate and unused data, and checking data types. In this ETL process, it produces data that will be used after being extracted and transformed.

TABLE II. ETL AGGREGATED RESULT

| trnsaction date | amount | term id | sum of amount | Count (id) |
|---|---|---|---|---|
| 03/08/2022 17:42:28 | 500,000.00 | xxG998xx | 500,000.00 | 1 |
| 28/09/2022 13:41:49 | 2,500,000.00 | xxVA03xx | 5,000,000.00 | 2 |
| 08/08/2022 17:58:07 | 1,250,000.00 | xxAKJHxx | 3,750,000.00 | 3 |
| 10/08/2022 19:04:45 | 1,000,000.00 | Government bank | 3,000,000.00 | 3 |
| 04/09/2022 16:58:11 | 100,000.00 | xxVA01xx | 100,000.00 | 1 |
| 11/09/2022 09:52:16 | 1,000,000.00 | xxHUSUxx | 1,000,000.00 | 1 |
| 29/08/2022 14:44:08 | 500,000.00 | xx6210xx | 500,000.00 | 1 |
| 14/09/2022 18:18:37 | 1,000,000.00 | xx5500xx | 2,000,000.00 | 2 |

### *C. Generate synthetic data with TimeGAN*

Generate synthetic data is the process of generating data by generating synthetic data with the same characteristics as real data, so as to develop and test AI models develop and test AI models. TimeGAN introduces the concept of supervised loss - the model is encouraged to capture the time conditional in the data by using the original data as supervision. In this process, the generative models, trained proactively and simultaneously through integration, are learned. The purpose of data synthesis is to show the results and effects of the various studies and to identify problems with methodology and quality. At this stage, the GAN Model is initialized with the details of the training process that will be carried out. Furthermore, there are several training data processes that are carried out by the GAN Model using data that has been collected at the beginning of the training process.

TABLE III. ATMS TRANSCACTION DATA

| trx_date | atm_id | trx_ammount | nominal |
|---|---|---|---|
| 01/07/2022 | xxVA01xx | 86 | 50500000 |
| 01/07/2022 | xxVA03xx | 78 | 51350000 |
| 01/07/2022 | xxVA03xx | 89 | 103500000 |
| 01/07/2022 | xxVA03xx | 22 | 11900000 |
| 01/07/2022 | xxVA02xx | 8 | 2050000 |

The GAN model is used to generate the new synthesized data shown in Table III. The next step is to generate new synthesized data to evaluate data quality. In the evaluation stage, several aspects are examined, such as characteristics of the aggregated data characteristics, the similarity of the synthesized data type with the original data, and several other aspects that can indicate the quality of the synthesized data. There are 3932 rows of collected data. Synthetic data generation was performed twice resulting in 7862 rows of data.

TABLE IV. GENERATED DATA

| entity | trx_date | atm_id | trx_amount | nominal |
|---|---|---|---|---|
| entity_1 | 01/07/2022 | xxVA02xx | 8 | 2050000 |
| entity_1 | 01/07/2022 | xxVA00xx | 42 | 51000000 |
| entity_1 | 01/07/2022 | xxVA03xx | 22 | 13150000 |
| entity_1 | 01/07/2022 | xxVA03xx | 62 | 36900000 |
| entity_1 | 01/07/2022 | xxVA03xx | 54 | 65600000 |
| entity_1 | 01/07/2022 | xxVA03xx | 20 | 11250000 |
| entity_1 | 01/07/2022 | xxVA03xx | 51 | 32300000 |

### *D. Training Model*

After the Data Synthesis stage with Time-GAN is complete, the next step is to build an ML model using the processed data. The first thing done at this stage is to divide the dataset into raining data and testing data with a ratio of 80:20.Validation data and testing data with a ratio of 80:20. The model is made using the Extreme Gradient Boosting (XGBoost) algorithm because of its advantages in dealing with missing values. The basic concept of this algorithm is to adjust the learning parameters iteratively to decrease the loss function (the evaluation mechanism of the model). XGBoost uses a more

organized model to build the regression tree structure, so as to provide better performance and be able to reduce model complexity to avoid overfitting [11].The final prediction result of XGBoost is the sum of the prediction results from each regression tree [12].

### E. Evaluation

MAE (Mean Absolute Error), MSE (Mean Squared Error), and R2 (R-squared) are commonly used evaluation metrics in regression modeling to evaluate model performance. Mean Absolute Error (MAE) measures the average absolute error between the model prediction and the true value. MAE illustrates the extent to which model predictions can deviate from the true value on average. MAE is calculated by summing the absolute difference between the prediction and the true value, then dividing it by the total number of observations. The smaller the MAE value, the better the quality of the model.

Mean Squared Error (MSE) is an evaluation metric that measures the average squared error between the model prediction and the true value. MSE gives more weight to large errors because it uses the square of the difference between the prediction and the true value. MSE is calculated by summing the squared difference between the prediction and the true value, then dividing it by the total number of observations.The smaller the MSE value, the better the quality of the model.

R-squared is an evaluation metric that describes how well the independent variables (features) explain the variation in the dependent variable (target). R-squared measures the proportion of target variability that can be explained by the model. The value of R-squared ranges from 0 to 1, and the closer to 1, the better the model is at explaining the target variability. R-squared is calculated by dividing the variance of the model by the total variance of the data

## III. Result and Discussion

Fidelity measures how well the synthetic data statistically matches the original records. It is provided through univariate and multivariate metrics, model and assumption-free. The anonymized columns are not considered for the fidelity scores since they are necessarily different between the real and synthetic records in order to achieve anonymization.

Missing Values Similarity (MVS) measures how close are the percentages of missing values in the synthetic and real data. This metric is bounded between [0-1], where 1 represents the same percentage of missing data. The table below present the five features with the highest and lowest similarity as shown in Table V.

TABLE V. MVS Result

| Feature | MVS (Higest) | MVS (Lowest) |
|---|---|---|
| atm_id | 1.0 | 1.0 |
| trx_amount | 1.0 | 1.0 |
| nominal | 1.0 | 1.0 |

Statistical Similarity (SS) measures how similar are the synthetic and real data considering five metrics: mean, standard deviation, median, 25% quantile, and 75% quantile. Each similarity is bounded between [0-1], where 1 represents equal values. Only numerical features are considered in this analysis as shown in Table VI.

TABLE VI. SS Result

| Feature | Mean | Std. Dev | Median | Q25% | Q75% |
|---|---|---|---|---|---|
| trx_amount | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| nominal | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |

Mutual Information (MI) measures how much information can be obtained about one feature by observing another. This metric calculates the similarity between real and synthetic MI values for each pair of features. It returns values between [0, 1], where closer to 1 is desirable (i.e., equal MI) as shown in Fig. 2 and Fig. 3.

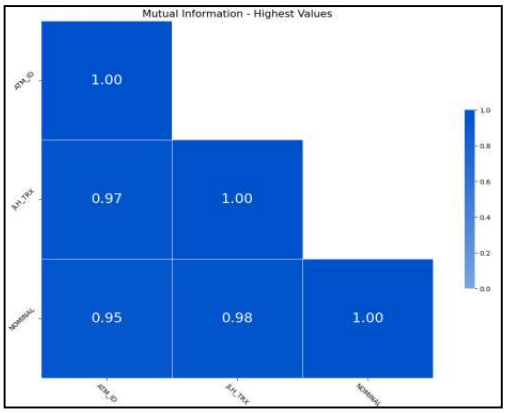


Fig. 2. Highest Value MI Result

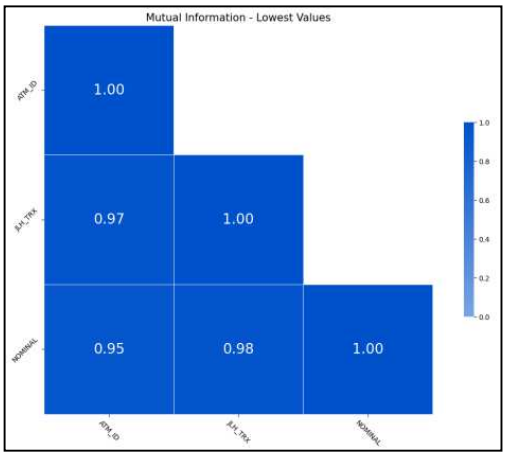


Fig. 3. Lowest Value MI Result

Range Coverage (RC) computes the similarity ratio between the numerical variables domain in the real dataset compared to the synthetic one. The two tables below present the five features with the highest and lowest coverage as shown in Table VII.

TABLE VII. RC Result

| Feature | RC (Higest) | MVS (Lowest) |
|---|---|---|
| nominal | 1.0 | 1.0 |
| trx_amount | 0.8 | 0.8 |

Dimensionality Reduction visualization plots show how closely the distribution of the synthetic data resembles that of the original data on a two-dimensional graph. Principal Component Analysis (PCA) algorithm used to reduce the datasets dimensionality. PCA captures any fundamental difference in the distributions of the datasets. The scatterplots represent depict this difference visually. Ate represent the two first main Eigenvectors that together explain 99.91% of of the total variance of the dataset.

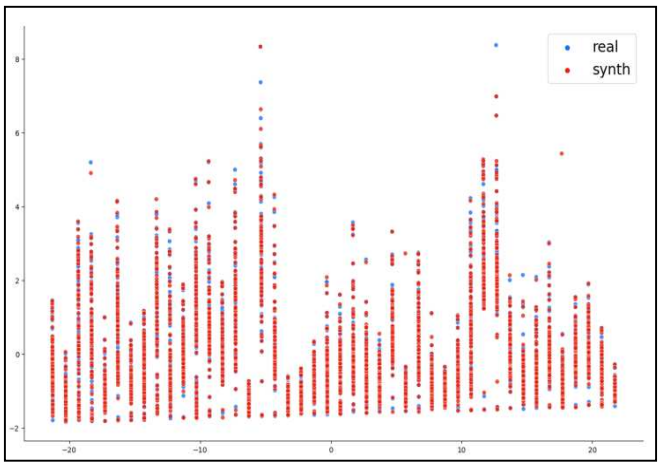


Fig.4 Dimensionality Reduction Visualization

Synthetic Classifier is used to predict metrics highlight the AUROC performance of several machine learning (ML) models that are trained separately on both the synthetic and original datasets. The score is calculated by testing the models on withheld data from the original dataset. The Train Synthetic Test Real (TSTR) score measures the performance of an estimator trained on synthetic data and later evaluated on real data. Train Real Test Real (TRTR) provides the estimator score expected if actual data was available. Suppose the TSTR and TRTR scores are comparable. In that case, the data generated by the synthesizer has a similar predictive performance as the original. Score between [0, 1]. For Machine Learning use, a predictive score above 0.8 is recommended as shown in Table VIII and Table IX.

TABLE VIII. AUROC PERFORMANCE ON SEVERAL ESTIMATORS

| Estimator | Real Data | Synth Data |
|---|---|---|
| Linear Regression | 1.62 | 1.43 |
| Multi-layer Perceptron | 2.79 | 1.59 |
| Decision Tree | 0.15 | 3.62 |
| Ridge | 1.62 | 1.43 |
| Lasso | 1.00 | 1.03 |
| Linear Support Vector | 1.66 | 2.05 |

TABLE IX. PREDICTION METRICS SCORE

| TRTR | TSTR | Score |
|---|---|---|
| 1.5 | 1.9 | 1.3 |

Fig.5 shows that data is divided into train data containing withdrawal transactions from July to Aug and test data containing withdrawal in September 2022 to evaluate the prediction model.

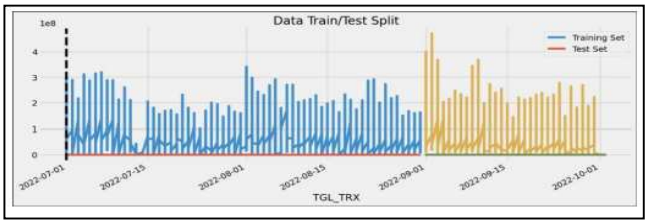


Fig. 5. Train Test Split Data

Fig.6 shows the first test of ML modeling using data that has not been imputed and without data cleaning. The data preprocessing process carried out is only data transformation, because XGBoost can predict data that still has missing values and nan values. the results of this test will be compared with tests that apply data cleaning and synthetic data generation.

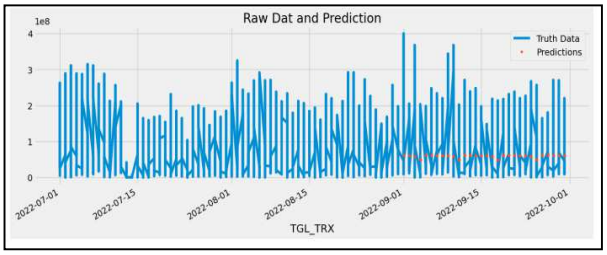


Fig. 6. Prediction Result on Unclean Data

Then the model is evaluated using Mean Absolute Error (MAE), Mean Squared Error (MSE), dan R-squared as follow in Table X.

TABLE X. MODEL 1 PREDICTION METRICS VALUE

| Metric | Value |
|---|---|
| MAE | 4.109786e+07 |
| MSE | 3.170747e+15 |
| R-squared | 1.139997e-02 |

The second test conducted using generated data. Fig.7 displays better prediction result of cash ATMs availability in September 2022.

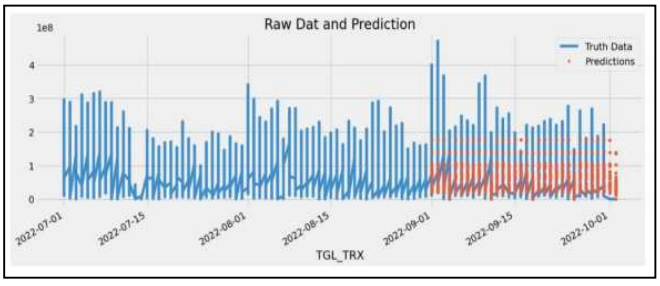


Fig. 7. Prediction Result on Generated Data

Table XI shows that the second model has a better evaluation metric value of Mean Absolute Error (MAE), Mean Squared Error (MSE), dan R-squared.

TABLE XI. MODEL 2 PREDICTION METRICS VALUE

| Metric | Value |
|---|---|
| MAE | 2.579965ee+07 |
| MSE | 1.648314e+15 |
| R-squared | 5.029429e-01 |

TABLE XII. CASH ATM PREDICTION IN SEPTEMBER 2022

| date | true_nominal | error | prediction | atm_id | amount |
|---|---|---|---|---|---|
| 09/01/2022 | 20150000 | 1013698 | 21163698 | xxva02xx | |
| 09/02/2022 | 33200000 | 11425456 | 21774544 | xxva02xx | |
| 09/03/2022 | 9350000 | 11146262 | 20496262 | xxva02xx | |
| 09/04/2022 | 4800000 | 7530686 | 12330686 | xxva02xx | |
| 09/05/2022 | 23600000 | 2497940 | 21102060 | xxva02xx | 91100000 |
| 09/06/2022 | 18750000 | 2352060 | 21102060 | xxva02xx | |
| 09/07/2022 | 19900000 | 988362 | 18911638 | xxva02xx | |
| 09/08/2022 | 11600000 | 9563698 | 21163698 | xxva02xx | |
| 09/09/2022 | 23200000 | 1425456 | 21774544 | xxva02xx | |
| 09/10/2022 | 4950000 | 15546262 | 20496262 | xxva02xx | 78400000 |
| 09/11/2022 | 13850000 | 1519314 | 12330686 | xxva02xx | |
| 09/12/2022 | 15650000 | 5452060 | 21102060 | xxva02xx | |
| 9/13/2022 | 19000000 | 2102060 | 21102060 | xxva02xx | |
| 9/14/2022 | 23550000 | 4638362 | 18911638 | xxva02xx | |
| 9/15/2022 | 20600000 | 563698 | 21163698 | xxva02xx | 92650000 |
| 9/16/2022 | 24250000 | 2475456 | 21774544 | xxva02xx | |
| 9/17/2022 | 4700000 | 15796262 | 20496262 | xxva02xx | |
| 9/18/2022 | 1000000 | 11330686 | 12330686 | xxva02xx | |
| 9/19/2022 | 13300000 | 7802060 | 21102060 | xxva02xx | |
| 9/20/2022 | 18650000 | 2452060 | 21102060 | xxva02xx | 61900000 |

The second model can predict estimated cash every 5$^{th}$ day on average in September 2022. Table XII displays the predicted required cash in one of the ATMs for a month in September 2022 in one of the ATM. which can accommodate a maximum of 200,000,000 IDR.

## IV. CONCLUSION

Based on the conducted research, XGBoost algorithm successfully obtained several conclusions. This research analyzes data by adding zone features to the dataset, and it affects the availability of cash ATMs based on time such as weekdays and holidays, with the results being able to predict when and where the Bank should fill up the money on the ATM machine. Evaluation of model performance is valued using Mean Absolute Error (MAE) 2.57 value, Mean Squared Error (MSE) 1.64 value, and R-squared 5.02 value.

Further research is expected to use more data over a longer period of time and use data from several different ATM withdrawals to be able to compare and analyze for long-term predictions.

## REFERENCES


[1] M. Rafi, M. T. Wahab, M. Khan, and H. Raza, *ATM Cash Prediction Using Time Series Approach*. 2020. doi: 10.1109/iCoMET48670.2020.9073937.

[2] V. Sarveswararao, V. Ravi, and Y. Vivek, "ATM cash demand forecasting in an Indian bank with chaos and hybrid deep learning networks," *Expert Syst. Appl.*, vol. 211, p. 118645, 2023, doi: https://doi.org/10.1016/j.eswa.2022.118645.

[3] A. Riabykh, I. Suleimanov, D. Surzhko, M. Konovalikhin, and V. Ryazanov, "ATM Cash Flow Prediction Using Local and Global Model Approaches in Cash Management Optimization," *Pattern Recognit. Image Anal.*, vol. 32, no. 4, pp. 803–820, 2022, doi: 10.1134/S1054661822040113.

[4] S. I. Serengil and A. Ozpinar, "ATM Cash Flow Prediction and Replenishment Optimization with ANN," *Uluslararası Muhendis. Arastirma ve Gelistirme Derg.*, no. February, pp. 402–408, 2019, doi: 10.29137/umagd.484670.

[5] J. Yoon and D. Jarrett, "Time-series Generative Adversarial Networks," no. NeurIPS, pp. 1–11, 2019.

[6] Hartono, O. S. Sitompul, Tulus, and E. B. Nababan, "Biased support vector machine and weighted-SMOTE in handling class imbalance problem," *Int. J. Adv. Intell. Informatics*, vol. 4, no. 1, pp. 21–27, 2018, doi: 10.26555/ijain.v4i1.146.

[7] A. Ibrahem Ahmed Osman, A. Najah Ahmed, M. F. Chow, Y. Feng Huang, and A. El-Shafie, "Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia," *Ain Shams Eng. J.*, vol. 12, no. 2, pp. 1545–1556, 2021, doi: 10.1016/j.asej.2020.11.011.

[8] A. A. Nababan, Sutarman, M. Zarlis, and E. B. Nababan, "Air Quality Prediction Based on Air Pollution Emissions in the City Environment Using XGBoost with SMOTE," in *2022 IEEE International Conference of Computer Science and Information Technology (ICOSNIKOM)*, 2022, pp. 1–6. doi: 10.1109/ICOSNIKOM56551.2022.10034887.

[9] Herlambang Dwi Prasetyo, Pandu Ananto Hogantara, and Ika Nurlaili Isnainiyah, "A Web-Based Diabetes Prediction Application Using XGBoost Algorithm," *Data Sci. J. Comput. Appl. Informatics*, vol. 5, no. 2 SE-, pp. 49–59, Jul. 2021, doi: 10.32734/jocai.v5.i2-6290.

[10] D. Aulia and H. Murfi, "XGBoost in handling missing values for life insurance risk prediction," *SN Appl. Sci.*, vol. 2, Aug. 2020, doi: 10.1007/s42452-020-3128-y.

[11] H. Sunata, "Komparasi Tujuh Algoritma Identifikasi Fraud ATM Pada PT. Bank Central Asia Tbk," *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, pp. 441–450, Dec. 2020, doi: 10.35957/jatisi.v7i3.471.

[12] S. Li and X. Zhang, "Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm," *Neural Comput. Appl.*, vol. 32, Apr. 2020, doi: 10.1007/s00521-019-04378-4.