Predicting Stock Price and Risk Using a Custom-Built GRU-Based Application

Mohammad Idhom [1], Trimono Trimono [1], Alvin Ryan Dana [1], Akhmad Fauzi [2], Prismahardi Aji Riyantoko [3]

[1] *Data Science Study Program, University of Pembangunan Nasional Jawa Timur, Surabaya, Indonesia*
[2] *Department of Management,University of Pembangunan Nasional “Veteran” Jawa Timur, Surabaya, Indonesia*
[3] *Department of Information and Communication Systems, Okayama University, Okayama 700-8530, Japan*

Abstract **– Price fluctuations and the loss risk are major problems in stock investing that need to be managed quickly and efficiently. The use of technology can improve the efficiency of price prediction and loss risk assessment. This study aims to implement the Gated Recurrent Unit (GRU) and Value-at-Risk (VaR) algorithms for price prediction and loss risk, packaged in a GUI-based application. GRU is designed to efficiently process sequential data by addressing the problem of short-term memory. VaR is a loss prediction method based on historical returns and quantiles of their distributions. There are two novelties in this study: first, the development of a hybrid model that integrates the GRU and VaR models into a single prediction model. The GRU prediction results are used as input values for the VaR model. Second, the development of a GUI-based application to accelerate prediction results. The results indicate that integrating GRU with VaR provides accurate results. This finding refers to the accuracy of the value and its conformity with the actual data. The GUI application consists of four main menus: data input, preprocessing, price prediction, and loss prediction. The GUI application has been proven to improve the prediction process without reducing accuracy.**

Keywords **– Investment, technology, loss risk, Gated Recurrent Unit (GRU), Value at Risk (VaR).**

DOI: 10.18421/TEM153-34
https://doi.org/10.18421/TEM153-34

Corresponding author: Mohammad Idhom,
Data Science Study Program, University of Pembangunan Nasional “Veteran” Jawa Timur, Surabaya
Email: idhom@upnjatim.ac.id

Received: 10 April 2025.
Revised: 06 April 2026.
Accepted: 18 May 2026.
Published: 27 August 2026.

1. Introduction

The rapid advancement of financial technology has transformed investment practices, making stock trading more accessible to a wider audience. As stock markets play a crucial role in economic stability and growth, investors seek innovative tools to navigate the volatility of financial markets effectively [1]. One of the key challenges in stock investment is predicting price movements while assessing the associated risks. In response to this need, various machine learning techniques have been explored to enhance prediction accuracy and risk evaluation.

Stock price predictions can be made using several approaches such as deterministic, stochastic, and machine learning. Traditional methods like ARIMA have been widely used but struggle with complex data patterns [2]. Machine learning algorithms, particularly Gated Recurrent Unit (GRU), can be an alternative to ARIMA models. GRU can handle time series data problems that have non-linear and seasonal patterns at the same time [3]. In addition, GRU also has more efficient computation time compared to other models such as RNN or LSTM forecasting [4].

Loss risk prediction is an important aspect of risk management. The loss risk value will be a reference for investors to manage their investments in order to avoid bankruptcy. VaR is a quantitative risk model that predicts the maximum loss risk over a certain period of time based on a predetermined confidence level [5]. This study implements the historical simulation method for VaR calculation, which utilizes past stock price movements to project future risks without assuming a predefined statistical distribution.

Existing studies have primarily focused on stock price prediction or risk assessment separately [6], [7]. Based on previous research, the use of GRU is still limited to stock price prediction. There has been no integration of the GRU model with risk measures to predict the risk of loss. This study bridges that gap by developing an application that combines GRU-based stock prediction with VaR-based risk assessment, providing a comprehensive tool for investors.

This research emphasizes the development of applications for price prediction and loss risk. This application works by integrating a machine learning model (GRU) with a risk measure model (VaR). The interaction between users and computers is designed in the form of two-way communication. Users act as determinants of how the model works specifically based on the given parameter values. Then, the computer becomes the executor in order to obtain accurate prediction results [6]. The integration of GRU with VaR is expected to become a guideline for investors in determining the direction of policy in financial asset investment risk management.

Users can input their own stock data into the application, allowing for flexible and customizable analysis. The application processes historical stock price data to generate future stock price predictions and risk assessments, helping investors make informed decisions.

There are three main stages in this study, including: (a) Pre-processing; includes preparation of historical data to build a GRU model; (b) Price prediction with the GRU model; the model is built based on historical data from the previous stage. The prediction results are evaluated based on the MAPE value; (c) Loss risk prediction with VaR.

The GRU prediction results become input data for calculating the loss risk value. Given the advantages of Backpropagation Neural Network (BPNN) in accuracy and precision for predictions, as demonstrated in AES applications [7], incorporating similar neural network techniques could further refine for risk assessment model.

The integration of machine learning algorithms and risk measurement models developed in a GUI-based application system is expected to become an accurate prediction tool for investors as a guide for investing. The combination of these two models allows investors to make risk management decisions more quickly from the perspective of price movements and risk value. This study contributes to the financial technology domain by demonstrating the practical application of machine learning in stock market analysis and risk management.

2. Methodology

This research is a quantitative study that uses a hybrid model and builds a GUI-based application. The following is the analysis procedure carried out:

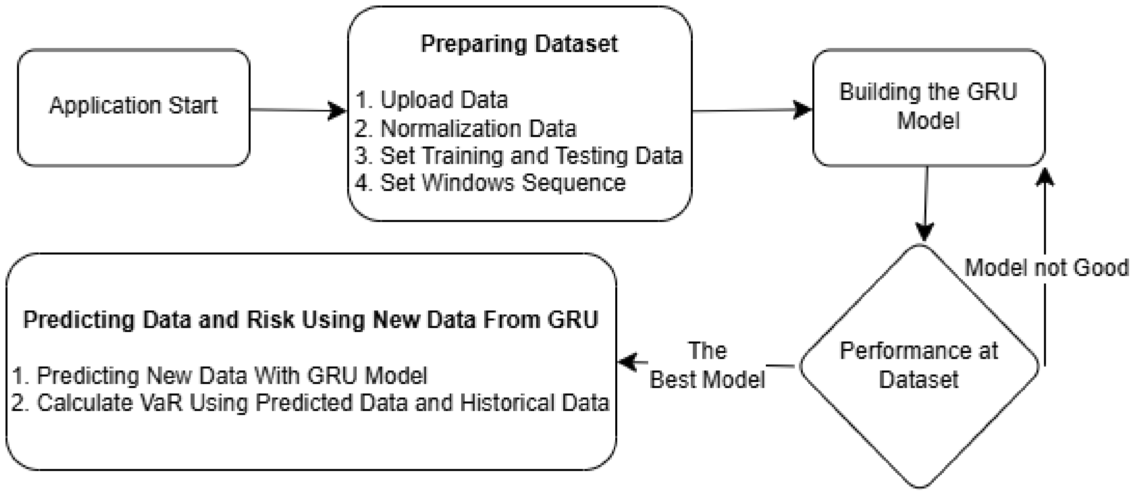


Figure 1. Application workflow

Figure 1 shows the methodology used in developing an application for stock prediction and risk assessment using GRU and VaR. This chapter explains the workflow of the application structured into several key stages, as depicted in the diagram.

2.1. Preparing Dataset

The first step in this research is preparation and pre-processing. This stage aims to ensure that the data meets the model criteria in order to obtain accurate prediction results. The work in this stage includes:

1. Upload Data: User can put any stock data to be used as variable for predicting new stock data and the risk of that stock.
2. Normalization: Scaling the data to ensure consistency and improve the efficiency of the GRU model [8].
3. Data Splitting: Data is divided into two parts, namely training data (for building models) and testing data (for model evaluation).
4. Set Windows Sequence: Dividing the data into sequence form to ensure predicting using GRU is smooth.

Once the dataset is prepared, the next step involves constructing the GRU model using the processed data. The user plays a role in defining model parameters, selecting appropriate hyperparameters, and training the model. The objective is to develop a model that can effectively learn from historical stock data and generate accurate predictions

2.2. Building the GRU Model

Once the dataset is prepared, the next step involves constructing the GRU model using the processed data. The user plays a role in defining model parameters, selecting appropriate hyperparameters, and training the model. The objective is to develop a model that can effectively learn from historical stock data and generate accurate predictions.

The GRU algorithm is composed of two main gates, namely the reset gate and the update gate [9]. The reset gate serves to limit the amount of historical data that needs to be eliminated in modeling. Meanwhile, the update gate determines the new information to be used [10]. This framework enables GRU to analyze time series data with long-term dependency structures without requiring gates with complex performance. The GRU model works by updating the hidden state using the formula below [10]:

$$z_t = \sigma\,(w_z x_t + U_z h_{t-1} + b_z) \qquad (1)$$

where:

z_t : Update Gate
w_z : Weight matrix for x_t
U_z : Weight matrix for the hidden state
σ : Sigmoid activation
b_z : Bias term update gate
h_{t-1} : Previous hidden state
x_t : Input at time t

z_t (update gate) works to regulate the amount of hidden state that needs to be passed on to the next layer [11]. The value is determined based on the sigmoid activation function used on the current input value and the weighted sum value on the previous hidden state.

$$r_t = \sigma\,(w_r x_t + U_r h_{t-1} + b_r) \qquad (2)$$

where:

r_t : Reset Gate parameter
w_r : Weight for determining Reset Gate value
U_r : Weight of value determination on Reset Gate for Hidden State
b_r : Bias value on the Reset Gate

The r_t parameter determines the historical data limit that needs to be removed [12]. r_t is calculated through the sigmoid activation function previously applied to the weighted sum value in the previous hidden state.

$$\widetilde{h_t} = \tanh\,(w_h x_t + U_h(r_t * h_{t-1}) + b_h)) \qquad (3)$$

where:

$\widetilde{h_t}$: Candidate State
b_h : Bias term value on candidate state
W : Weight for calculating the candidate state
U_h : Weight value in determining candidate states
tanh : Tangent activation function

The candidate hidden state $\widetilde{h_t}$, represents the temporary updated state before being adjusted by the update gate [12]. The h_t parameter is calculated using the tanh function (hyperbolic tangent function) applied to the X parameter (total weighted value of the current input) and Z (element-wise product obtained from the reset gate with the previous hidden state).

$$h_t = z_t * h_{t-1} + (1 - z_t) * \widetilde{h_t} \qquad (4)$$

Where h_t denote Hidden State parameter.

The final value of the h_t parameter is determined through interpolation between *M* and *N*, where *M* is the previous hidden state; and *N* is the candidate hidden state [13]. The update gate controls the weighting of these two states, ensuring that relevant past information is retained while incorporating new knowledge.

2.3. Evaluating Model Performance

After training, the model undergoes evaluation to assess its performance. This stage involves testing the prediction results against actual data in order to measure the accuracy of the model. [14]. If the model does not meet the expected performance criteria, refinements and retraining are conducted to improve its predictive capabilities.

In this research, model accuracy will be evaluated using Mean Absolute Percentage Error (MAPE). The main advantage of MAPE is that it provides clear assessment standards that are easy to interpret and does not depend on the type of scale and data units used [15]. MAPE works by comparing the average percentage error between the predicted value and the actual value. Mathematically, MAPE is formulated as follows:

$$MAPE = \sum_{t=1}^{n} \left| \frac{y_i - \hat{y}_i}{\hat{y}_i} \right| \text{x } 100\% \qquad (5)$$

Where:

n : Sample size
$\hat{y}_i$: Predicted value at i-th period
y_i : Actual value at i-th period

The following is the classification of prediction accuracy based on the MAPE value [17]:

Table 1. Prediction accuracy classification based on MAPE values

MAPE value intervals	Accuracy category
<10%	Very accurate
11 – 20 %	Accurate
21 – 50%	Moderately accurate
>51%	Not accurate

Based on the accuracy category in Table 1, it is known that the recommended prediction value is at least in the 10 – 20% interval so that the accuracy results are still included in the accurate category.

2.4. Loss Risk Prediction using VaR Based on Predicted Value from GRU Model

After obtaining the GRU model with optimal parameters, the model is used to determine the predicted price. Through the GUI application that has been developed, users can determine:

1. Predicting New Data with the GRU Model: Running the trained model to produce future stock price estimates.
2. Calculating VaR Using Predicted and Historical Data: Combining predicted and historical stock data to compute Value at Risk (VaR), which provides insights into potential financial risks.

Risk prediction is determined using the Value at Risk (VaR) method with a Historical Simulation approach. VaR is defined as the maximum loss that will occur in a certain period of time with a confidence level of (1-α) % [18].

Specifically, the Historical Simulation approach works by utilizing historical return data without any specific distribution assumptions. Historical simulation is based on the principle that past events are likely to recur and occur again in the future [19]. VaR-HS is formulated as follows:

$$VaR = V_0 \text{ x } P_\alpha \text{ x } \sqrt{t} \tag{6}$$

Where:

VaR : Los risk prediction
V_0 : Initial investment value
$\sqrt{t}$: Holding period
P_α : The α-th quantile of return distribution

$$P_\alpha = a \text{ x } n \tag{7}$$

n : Sample size

By using historical data, VaR provides a quantitative estimate of financial risks associated with stock predictions, allowing for better decision-making in risk management.

The results section describes the system design developed for predicting stock prices while simultaneously analyzing investment risks. This system integrates historical stock data with predictions generated by the Gated Recurrent Unit (GRU) model. The primary objective of this system is to provide a broader understanding of stock price movements and potential future risks.

The system consists of several key components: processing historical data as test data, predicting prices using the GRU model, and calculating risks by integrating historical data with prediction results. This approach is designed to provide accurate analysis and assist users in making informed investment decisions based on comprehensive information. Additionally, the system is equipped with a graphical user interface (GUI) designed to simplify users' understanding of predictions and risk analysis. With this design, the system aims not only to produce accurate forecasts but also to provide clear insights into stock price patterns and risks.

2.5. System Interface and Implementation

The application system is developed using the Tkinter library to build the graphical interface. Named the "Stock Prediction and Investment Risk Analysis Application," this tool assists users in predicting stock prices and analyzing risks. this tool is designed to assist users in predicting stock prices and analyzing potential investment risks based on historical market data. The application works by combining the GRU model with VaR. Each model has a specific task. GRU is tasked with processing historical stock price data as a reference for obtaining price prediction values. VaR is tasked with using the price prediction results from the GRU model to measure the risk of losses that will occur.

The interface of the system is structured with a well-organized layout, making it accessible for users with different levels of experience in financial analysis. It includes various sections such as data input fields, a prediction results display, risk assessment indicators, and a visualization panel. Users can input stock market data, select desired parameters, and run the predictive model seamlessly.

2.6. Main Interface

Figure 2 illustrates the initial display of the Stock Prediction and Investment Risk Analysis Application, which predicts stock prices and risks for blue-chip stocks such as MYOR and TBIG. The application utilizes the GRU model to provide accurate analysis, aiding users in making investment decisions.

At the center, a brief description of the application's purpose is displayed in a gray-colored box. Below, three primary buttons are available: "View Graph" for data visualization, "Prediction Page" for stock price predictions, and "VaR Page" for Value at Risk (VaR) calculations as part of investment risk analysis. The simple yet functional design enhances user comprehension and usability.

Figure 2. Main interface of application

The Python script implementing this GUI displays the application title followed by a frame containing a brief project description, emphasizing stock price and risk prediction using the GRU model. Below this frame, three main navigation buttons provide access to other pages: Upload Data, Prediction Page, and VaR Page. The bottom section of the interface includes a footer displaying application copyright information. This structured and informative design serves as the starting point for users to explore the application's core functionalities.

2.7. Upload Data Feature

Figure 3 presents the "Upload Data & View Statistics" menu, designed for data upload functionality and stock statistics visualization. This page focuses on stock data manipulation and analysis. At the top, four primary buttons are available: "Upload CSV" for selecting stock data files, "Show Graph" for refreshing the visualization with updated data. The "Process Data" menu to to perform pre-processing on the data, and "View CSV" for to display the data that has been uploaded.

Figure 3. Upload and preprocessing menu

This application provides a user-friendly interface for analyzing stock data. Users can upload CSV files, visualize data trends through interactive graphs, and access key statistical insights. The app ensures data accuracy by verifying required columns before generating visual representations. Additionally, it presents essential descriptive statistics, such as the mean and standard deviation, for a comprehensive overview of the uploaded data.

Table 2. Features on the data upload page

Feature	Function	Description
Upload Data	Allows users to upload files or datasets.	Users can upload file types CSV for processing.
Show Data	Displays the uploaded or processed data.	After uploading, users can view the content in a structured format, such as a table.
Windows Data	Splits data into training and testing sets, then sequences it.	Users can choose the train-test split ratio and define how many sequential data points should be included in each window. This is useful for time-series analysis.

As Table 2 describes, the Windows Data feature allows users to preprocess sequential data through normalization, splitting, and windowing. Data normalization using the Min-Max approach serves to reduce data to values between 0 and 1.

This process is carried out using the following equation [16]:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (8)$$

Where:

x_{scaled} : normalization process results data
x_{min} : the smallest data value
x : actual value
x_{max} : the largest data value

This step helps improve model stability and is an essential part of preprocessing, as demonstrated in the study, which evaluates the impact of various data normalization techniques on ANN model performance for electricity consumption prediction [20].

Then, users are asked to determine the training and testing data distribution scheme. Users can choose several combinations of values. For example, 80% and 20%. This means that 80% of the initial data will be allocated as training data, while the remaining 20% will serve as testing data. At this stage, users must ensure that the data distribution is proportional, with the training data proportion being greater than the testing data proportion. Finally, the sliding window method is applied, where the user specifies a sequence length to structure the data. This method creates overlapping windows of consecutive time steps. Each sequence serves as an input, and the next value is the prediction target. This approach is particularly useful for time-series forecasting and machine learning models that require sequential dependencies such as GRU [21].

TEM Journal. Volume 15, Issue 3, pages 2439-2454, ISSN 2217-8309, DOI: 10.18421/TEM153-34, August 2026.

2.8. Prediction Page

Figure 4 illustrates the model configuration page for prediction, designed with a dynamic layout.

At the top, the model configurations can be made via the "Configure Layer" button, with the model architecture details displayed in the "Model Architecture" text area.

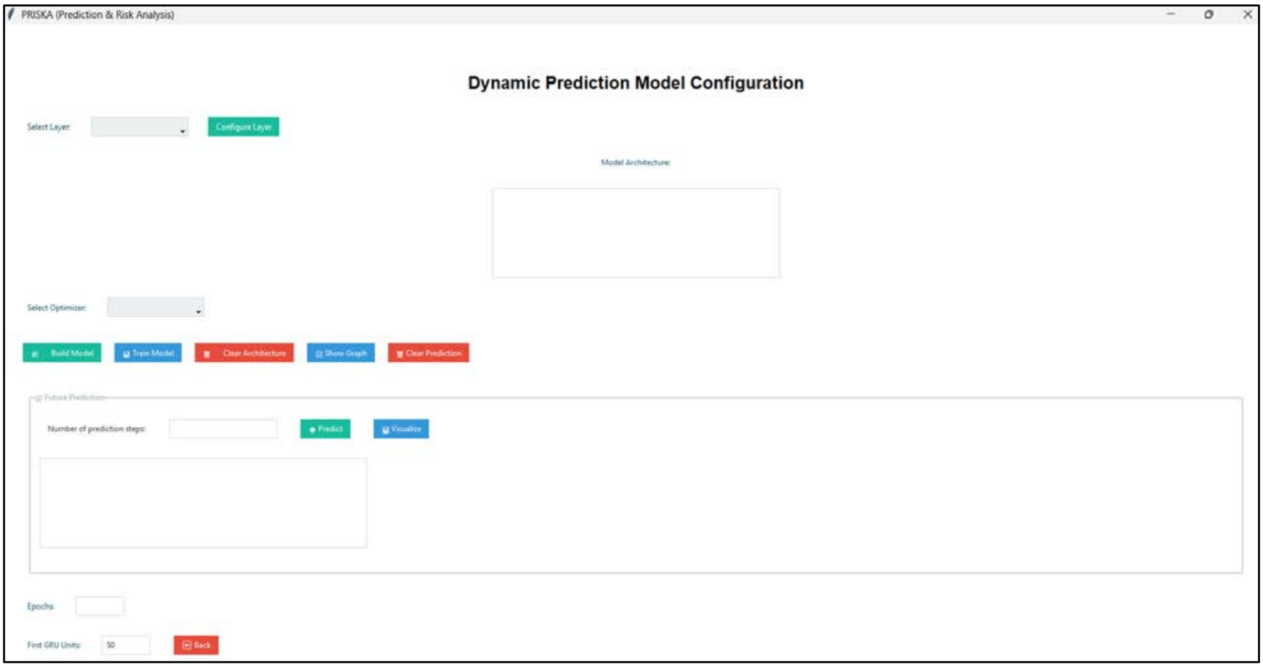


Figure 4. Prediction page

This application offers an intuitive interface for configuring a stock prediction model powered by neural networks. Users can customize the model by selecting layers, choosing an optimizer, and adjusting prediction settings.

The interface allows for seamless model architecture refinement and training, enabling users to build and optimize their predictive models with ease.

Table 3. Features on the prediction page

Feature	Function	Description
Select Layer	Allows users to choose neural network layers.	Users can customize their model by selecting different layers, enabling dynamic model architecture.
Train Model	Trains the selected predictive model.	The system learns from the dataset to make future predictions based on the chosen model structure.
Delete Model	Removes an existing trained model.	Useful for managing different models and versions.
Test Model	Evaluates the performance of the trained model.	Runs the model on test data to measure accuracy and effectiveness.
Show Prediction	Displays prediction results.	Visualizes the model's predictions and performance metrics through graphs and numerical output.
Prediction Analysis	Provides insights into model performance.	Displays key evaluation metrics such as accuracy, loss, and error analysis.
Input Custom Parameters	Allows users to manually enter model parameters.	Users can fine-tune hyperparameters such as learning rate, batch size, and activation functions.
Set GRULayers	Adjusts the number of GRU layers.	Configures the depth of the GRU-based neural network for sequential data processing.
Epochs	Sets the number of training iterations.	Determines how many times the model processes the entire dataset during training.

Referring to Table 3, there are nine features in the application that can be used. These features function in building a GRU model for price prediction. The resulting price predictions then serve as a reference for predicting the risk of loss.

In this feature also showcases the trained model's evaluation results. The visualizations include loss values indicating prediction errors and a comparison between predicted and actual data. This interface enables users to comprehensively assess model performance. The loss is calculated using Mean Squared Error (MSE), which is defined as [22].

$$MSE = \sum_{i=1}^{n} \frac{\sqrt{(x_i - E(i))^2}}{n} \qquad (9)$$

Where:
MSE: Mean Squared Error
x_i : Data Test
$E(i)$: Predicted Data
n : Total of Data

where n is the number of samples, x_i is the actual value, and $E(i)$) is the predicted value. A lower MSE indicates better prediction accuracy. The training process involves backpropagation, which updates the models' parameters based on the computed loss, and this is iterated over the number of epochs set by the user.

The next step in training a GRU is to apply Backpropagation Through Time (BPTT) to compute the gradients and update the weights [23]. BPTT is a method used to propagate errors over time in recurrent networks like GRU, utilizing the chain rule to compute gradients with respect to each parameterc [24], [25].

Users can input the desired number of predictions, which are made using the last window of available test data. The system applies a sliding window approach, where each predicted value is used as input for the next step, repeating until the specified number of predictions is reached.

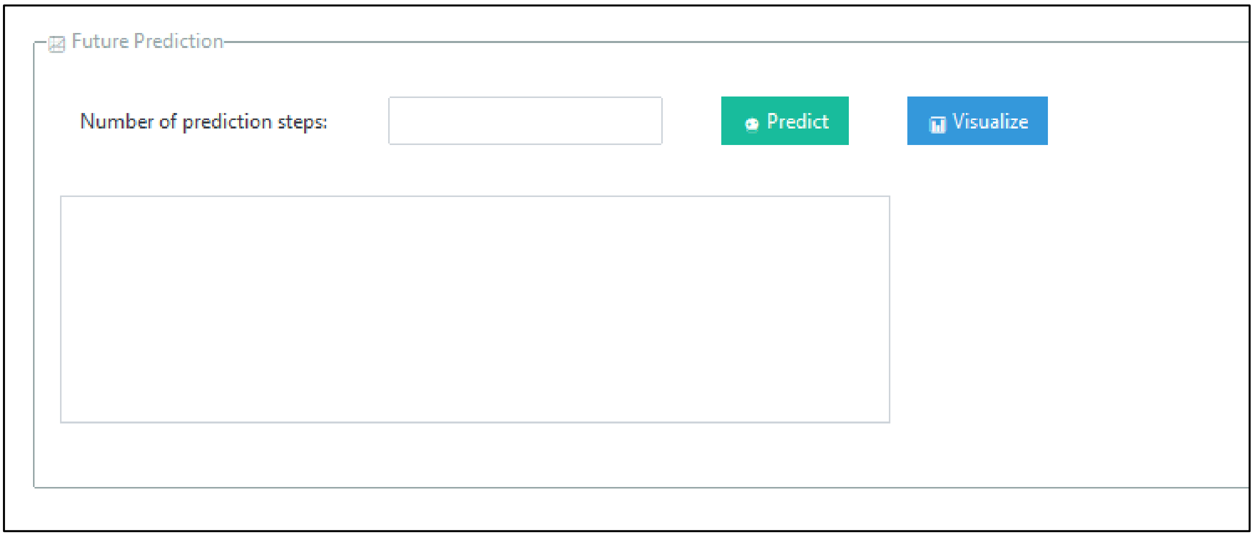


Figure 5. Prediction feature

To obtain the predicted values, users can press the Predict button on the user interface. This button executes the trained model and generates predictions, as shown in Figure 5.

2.9. Value at Risk (VaR) Page

Figure 6 displays the "Investment Risk Analysis" page, designed for investment risk analysis based on return data.

The top section features a histogram chart illustrating return data distribution, helping users understand stock risks. Below the histogram, a red notification warns users that prediction data is unavailable for VaR calculations. Three interactive buttons are provided: "Retrieve Test Data" to load return data, "Calculate VaR" to compute risk values, and "Back" to navigate to the previous page. The lower section includes a results area where VaR values appear after computation.

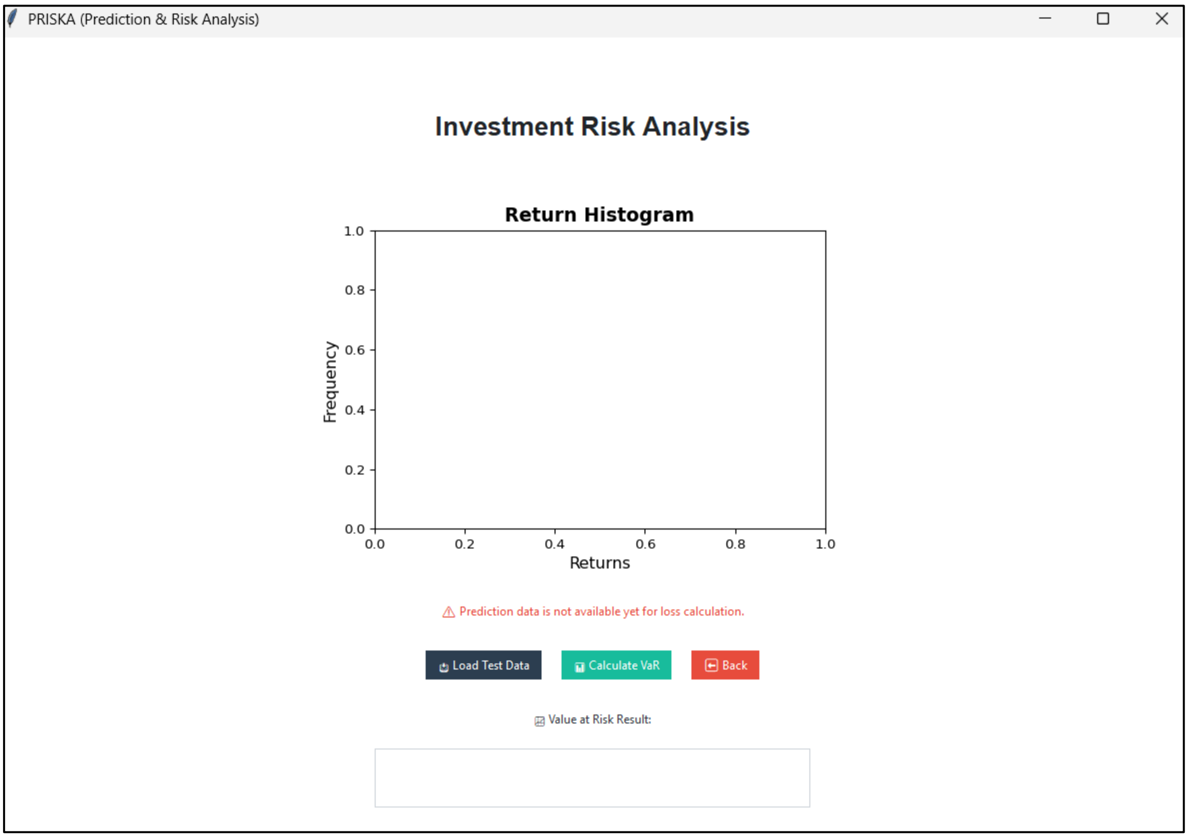


Figure 6. Value at Risk page

Figure 6 shows a display of the risk analysis menu using VaR. The VaR calculation is performed after the price prediction process is complete. The Python script implementing this page uses Tkinter to create a GUI for VaR-based risk analysis. The HalamanRisiko class, a subclass of *tk.Frame*, manages visual elements such as histogram charts, notifications, and result displays.

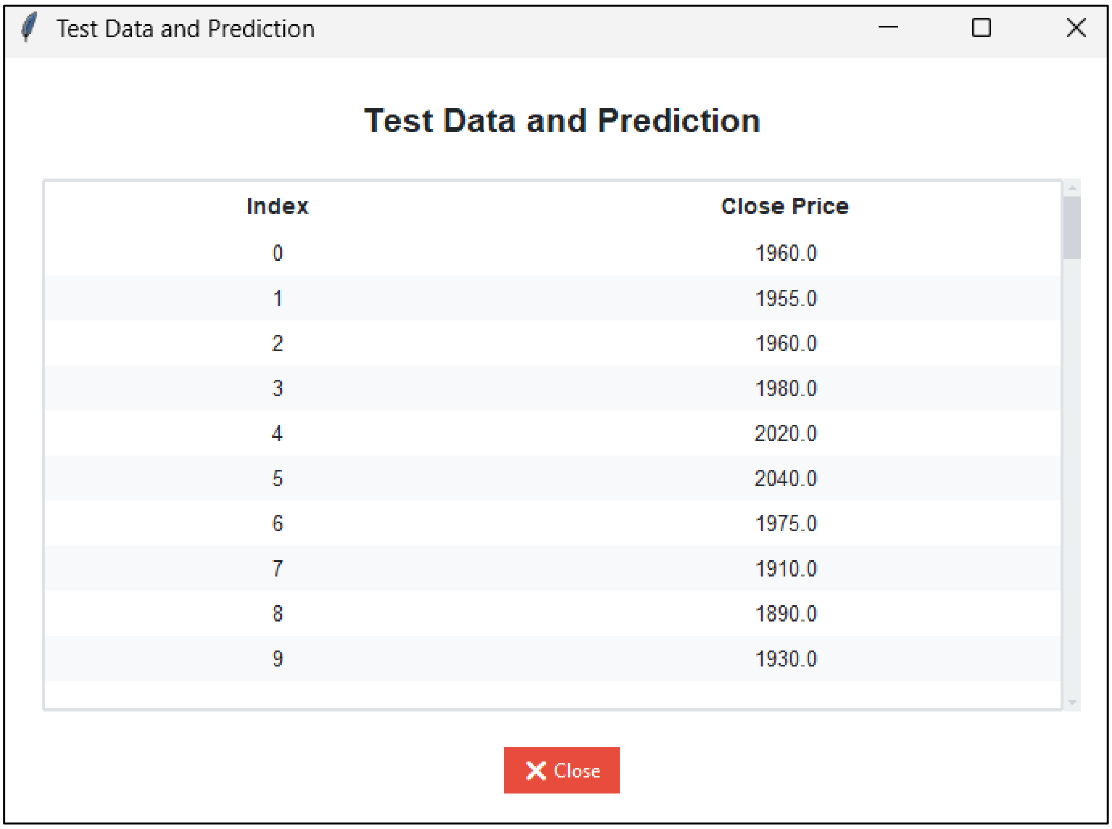
Users retrieve test data using an interactive button and compute VaR using an integrated method. If data is unavailable, a notification appears automatically, and the computed VaR results are displayed in a designated text area.

Table 4. Features on the Value at Risk page in figure 6

Feature	Function	Description
Histogram Returns	Displays the distribution of returns.	A graphical representation showing the frequency of different return values.
Get Test Data	Loads test data for analysis.	Retrieves test dataset for risk analysis and Value at Risk (VaR) calculation.
Calculate VaR	Computes Value at Risk.	Performs risk assessment using statistical methods to estimate potential losses.
Back	Returns to the previous screen.	Allows the user to navigate back without proceeding further.
VaR Result	Displays the calculated VaR value.	Shows the risk estimation result after running the VaR calculation.

In Table 4, there are a selection of features that can be utilized by users when making risk predictions, starting from histogram visualization, to risk prediction values calculated using a historical simulation approach.

Then, users can view the collected data by pressing the View Data button, as shown in the Figure 6. This button allows users to access and review the data visually, making it easier to verify, analyze, and understand the collected information.



Test Data and Prediction

Test Data and Prediction

Index	Close Price
0	1960.0
1	1955.0
2	1960.0
3	1980.0
4	2020.0
5	2040.0
6	1975.0
7	1910.0
8	1890.0
9	1930.0

Close

Figure 7. Data collection used for VaR calculation

Figure 7 is an example of the data display that will be used to determine model parameters in VaR risk prediction. The final output displays a graphical representation of the risk analysis, where the histogram includes a red vertical line indicating the VaR value, along with a percentage label representing the associated risk level. This allows users to visually interpret the financial risk based on the analyzed data. Additionally, the calculated VaR result is shown alongside the graph, providing users with both a numerical and graphical understanding of potential losses over the selected period.

3. Results

In this study, simulations were conducted on Mayoora Ltd shares with the code MYOR.JK, which are classified as bluechip stocks on the Indonesia Stock Exchange. This selection was based on several aspects, including: (i) Stable and high market capitalization over the past 5 years; (ii) High demand among investors, as evidenced by high daily trading volumes. The time range used in the historical data is from July 1, 2019, to July 31, 2024 (1,237 trading days). Data analysis was performed using a GUI application. The first result obtained was a time series plot visualization, which is presented in Figure 8:

Figure 8. Dataset visualization

Referring to several previous studies, the ideal proportion of training and testing data is 80% and 20%. This distribution is considered robust enough for the GRU model to obtain the best accuracy. Therefore, this proportion will be used in this study.

Additionally, a sliding window technique with a 40-day window was implemented to capture short-term trends and patterns within the dataset.

Figure 9 illustrates the data input process within the "View Graph" menu. When the "Upload Data" button is clicked, the application opens a file explorer dialog for users to select the desired stock data file. The system ensures that only CSV files are accepted as the standard format for stock data storage. This feature allows dynamic external data input for real-time stock analysis, ensuring compatibility with supported file formats.

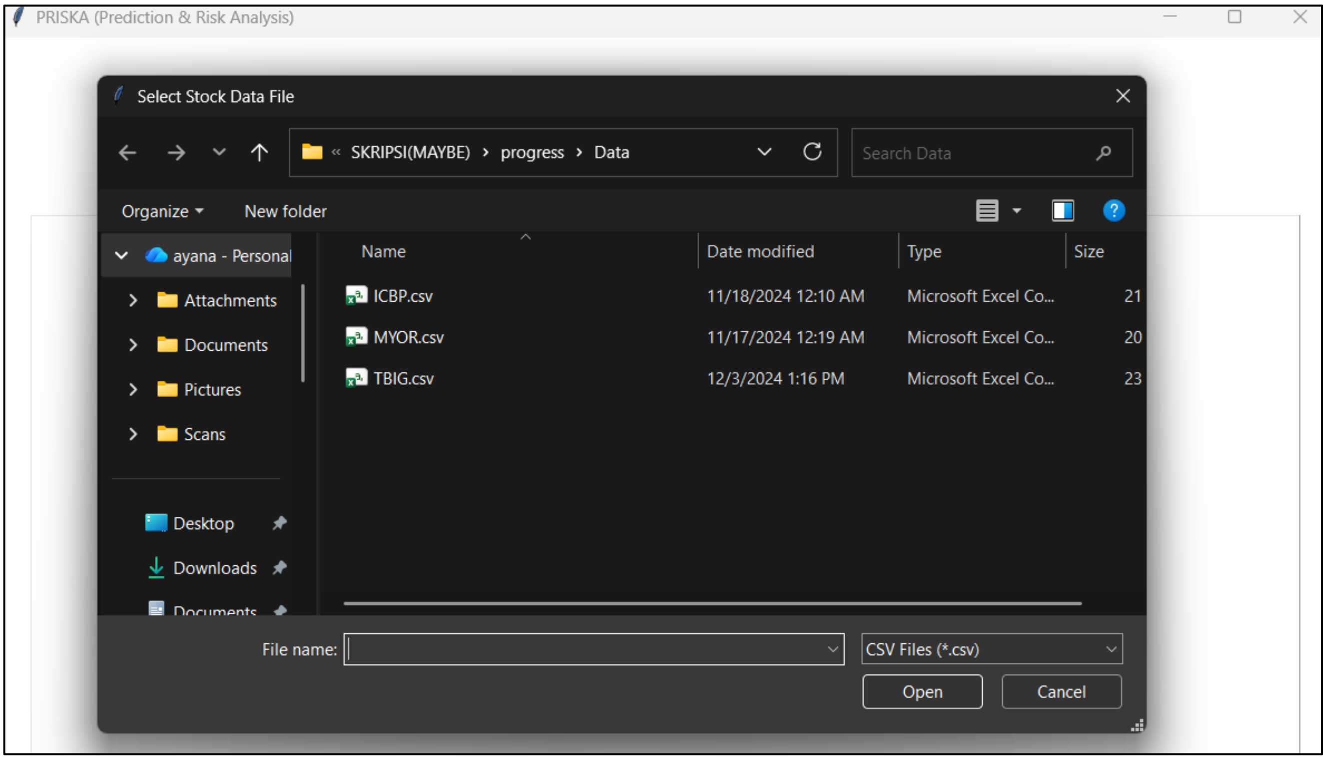


Figure 9. Data imputation process into the application

Figure 10 shows the GUI interface after data is uploaded and the "Update Graph" button is clicked. The displayed line graph illustrates "Closing Prices Over Time," with the X-axis representing dates and the Y-axis representing stock prices.

At the bottom of the time series plot, there is a text box that displays descriptive values of stock price data. These descriptive values allow users to obtain information about the initial characteristics of the data to be analyzed using the GRU and VaR models.

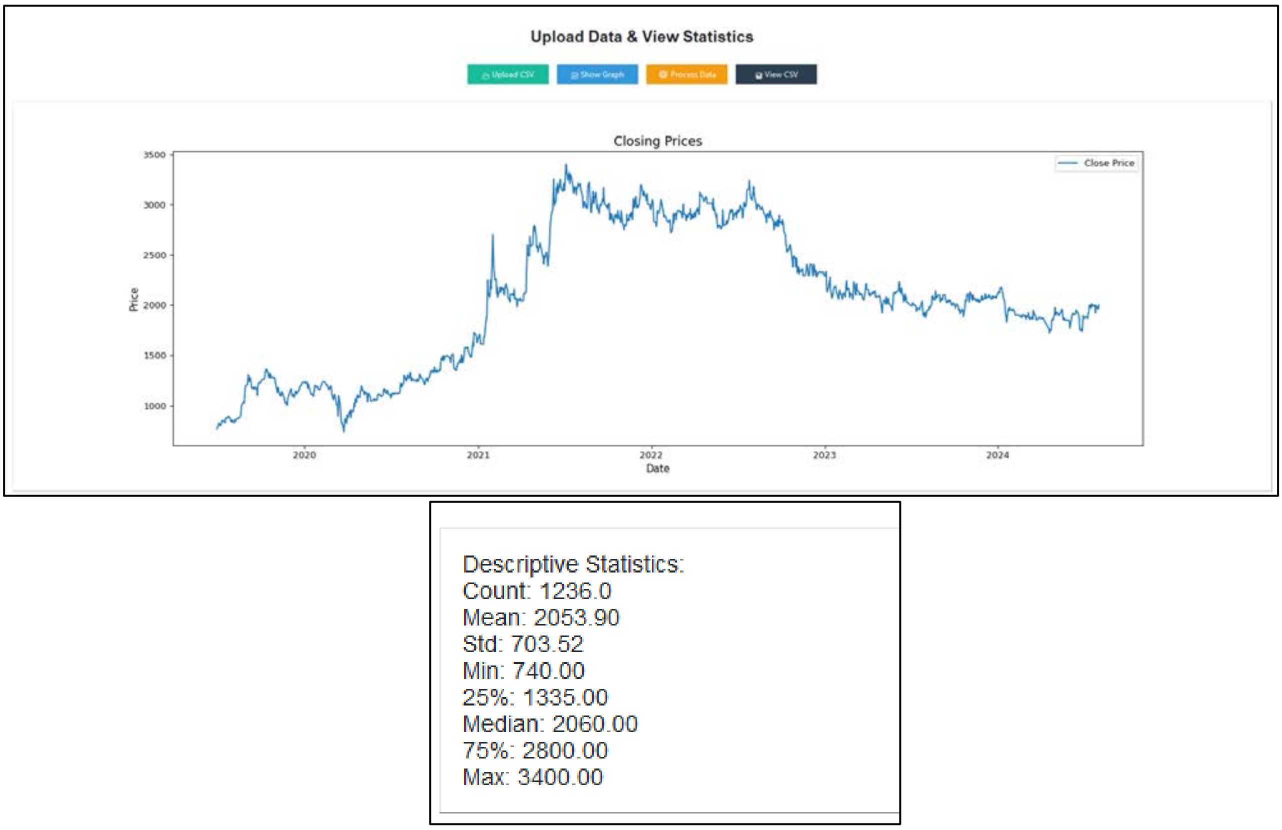


Figure 10. Visualization results in the application

Additionally, the "Windows Data" button processes uploaded data by splitting it into training and testing sets. A sliding window method is applied to both datasets, preparing them for stock price prediction. This feature ensures automatic data preparation, making it ready for predictive modeling

Next, the application was developed to create a Gated Recurrent Unit (GRU) model, allowing users to generate and train their own GRU-based predictions. This model was selected for its efficiency in handling sequential data, particularly in capturing temporal dependencies in stock price movements. The application enables users to customize various parameters to optimize the model's performance based on their specific needs. The GRU model in the application was configured with the following parameters.

In this study, a price prediction simulation was carried out using GRU with the selected parameters attached in Table 5:

Table 5 GRU model implemented in the application

Parameters	**Value**
GRU	(64, Return Sequence = TRUE)
Dropout	(0.2)
Dense	(8, Rel-U)
	(1, Linear)

By integrating this functionality, the application provides a flexible and interactive approach to time-series forecasting, empowering users to analyze and predict stock price trends with greater accuracy. During the model training phase, the GRU model will be optimized using the Adam optimizer with an Epoch value of 50. This optimization aims to help the model improve its ability to learn varying data patterns while maintaining an efficient computational process.

In addition to displaying predicted values, this application also displays output in graphical form, including Training and Validation Loss, as well as a comparison graph between predicted and actual values. The graphical display is shown in Figure 11:

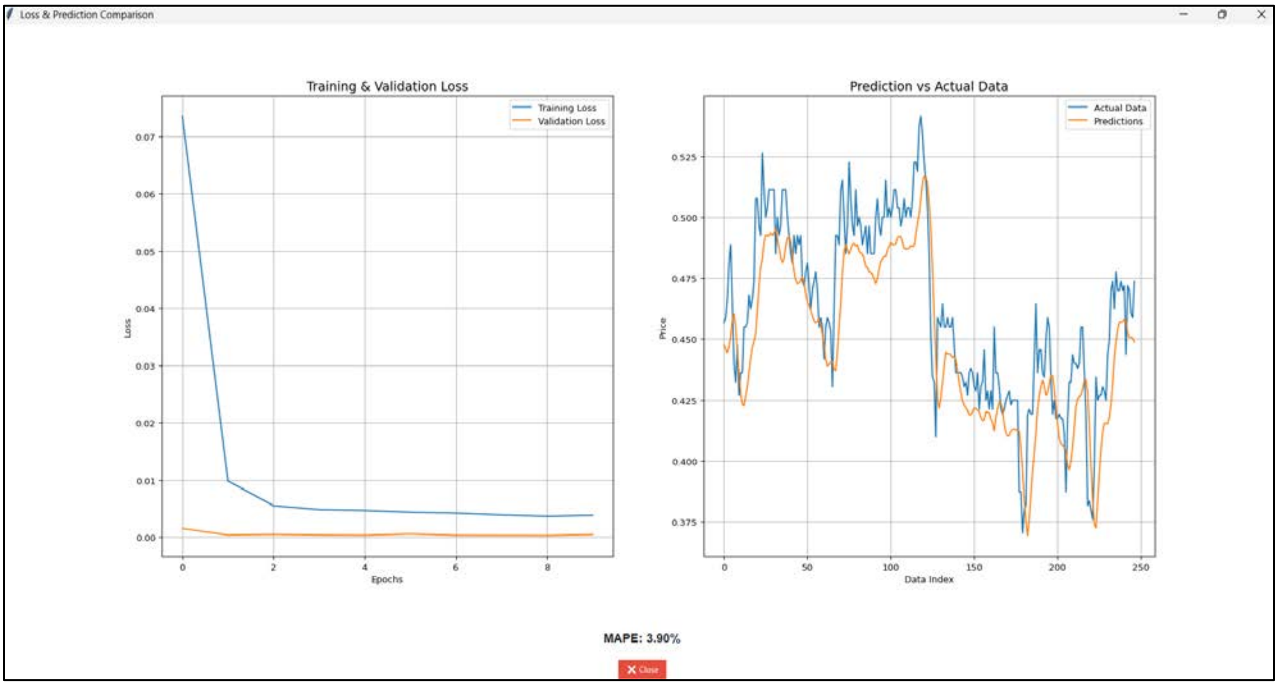


Figure 11. Model performance results in the application

Furthermore, the MAPE value is used as a measure to evaluate the model's accuracy so that users will have definite information on how accurate the prediction results are. The evaluation results indicate that the Adam optimizer achieving the lowest MAPE of 2.33%.

Once the model is trained and optimized, it will be used for predicting stock price data and assessing potential risks using new input data processed through the GRU model.

The predictions will generate stock price forecasts for the next five days, allowing for short-term trend analysis and risk evaluation. The predicted results will be shown in the application, providing users with essential information on expected market movements. This enables users to analyze potential risks and make informed decisions based on the forecasted data.

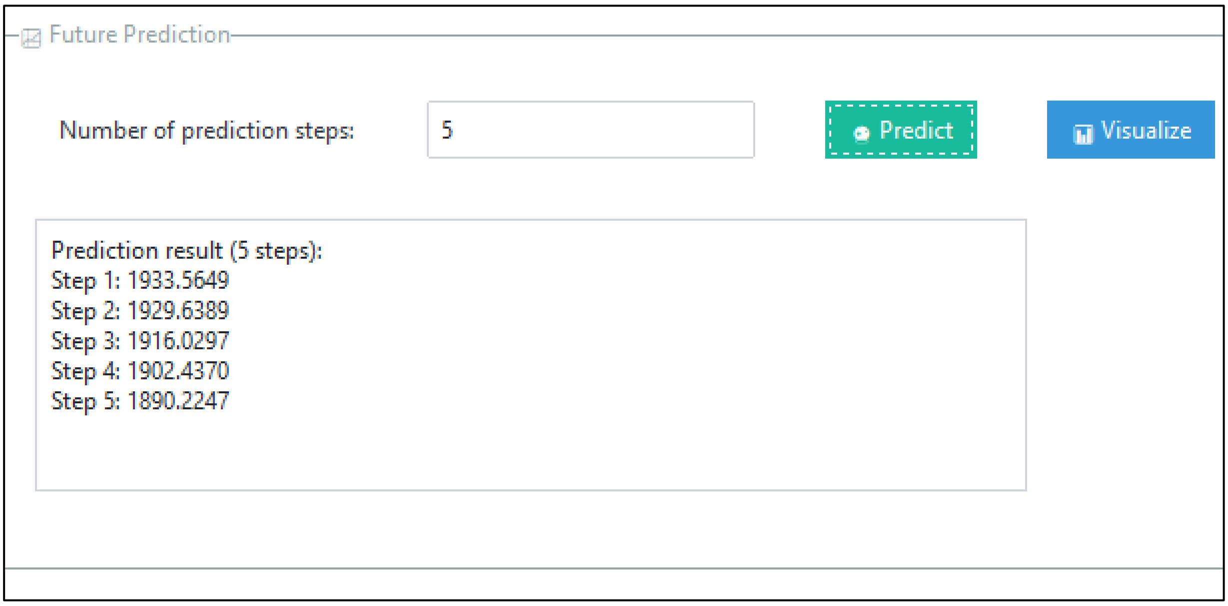


Figure 12. New data prediction results in the Application

Figure 12 is an example of the prediction results provided by GRU for the period after the out-sample date.

Users can adjust the number of prediction results if they require a longer prediction period.

Then, the visualization of the predicted data can be viewed by users by clicking the "Visualization" button. Upon clicking, the system will generate a graphical representation of the stock price trends, enabling users to compare actual test data with the predicted values.

In this visualization, the test data is displayed in blue, representing the actual market movements, while the predicted stock price trends are illustrated using a red dashed line. This intuitive representation allows users to easily identify patterns, deviations, and trends, helping them gauge the accuracy of predictions and assess potential risks.

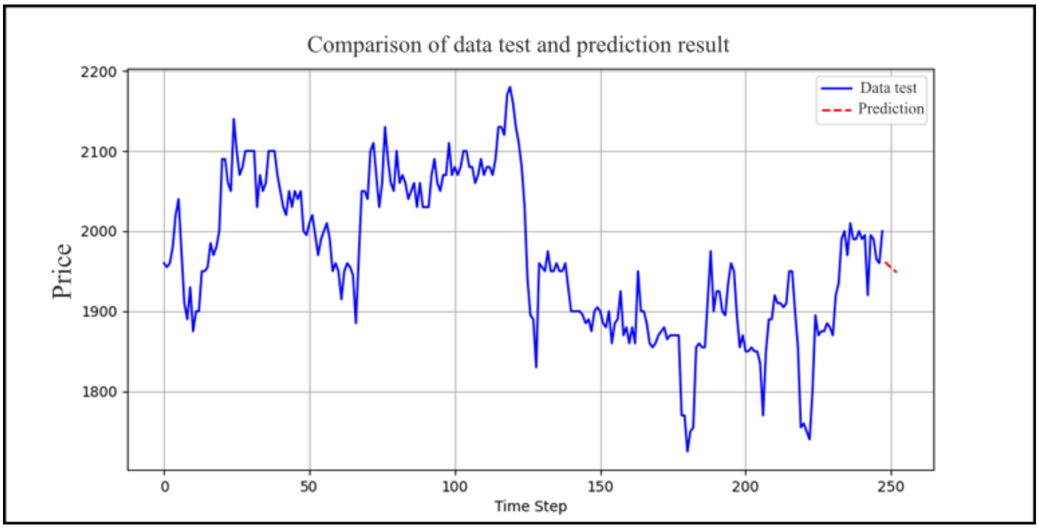


Figure 13. Visualization of Test Data and Predicted Data

Referring to Figure 13, users can compare the predicted results for the period after the out-of-sample date with the testing data. This figure can serve as a reference for observing the price movements that occurred.

The last section of this application contains a loss risk analysis. Risk prediction using the VaR-HS method is carried out by utilizing stock price data predicted by the GRU model. In this simulation, the VaR model parameters are determined as follows:

V_0 = IDR 1.000.000,-
t = 5 days
1- α = 95%

The determination of these parameters is not absolute, meaning that users can replace them with other values according to their needs. Based on the analysis obtained, the quantile value at α = 5% is -0.0283. This means that the predicted risk of loss for t = 1 is no more than 2.83% of the total funds invested. For t = 5, using equation (6), the VaR prediction is 6.3362%. If the initial investment value is IDR 1,000,000, then the nominal predicted loss is IDR 63,362.

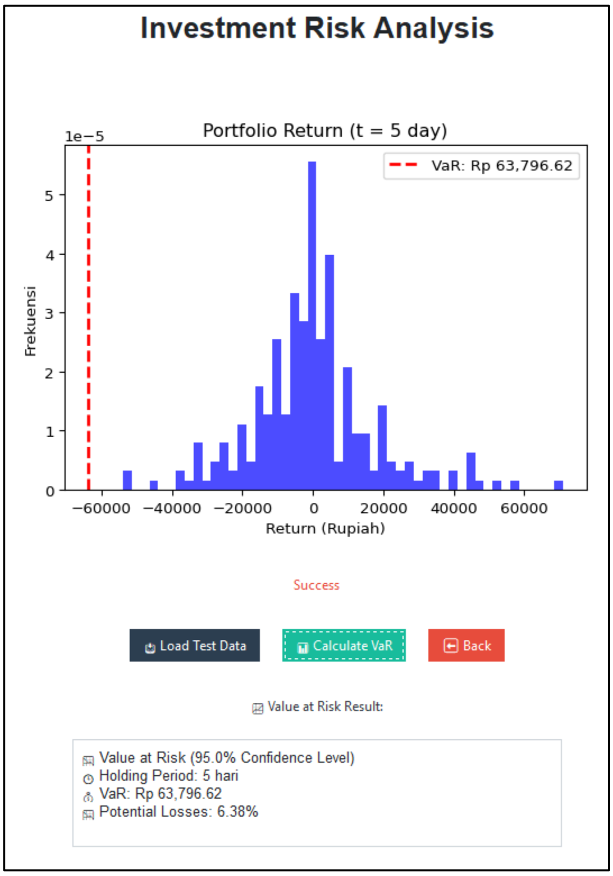


Figure 14. VaR calculation results in the application

Based on Figure 14, the VaR prediction results provide information to users regarding the possible value of the risk of loss that will occur (in percentage), in addition to that, information is also presented on the selected holding period, as well as the confidence level value. Further interpretation based on the VaR prediction value is that, at a 95% confidence level and a holding period of 5 days, the maximum risk of loss that investors may experience if they invest IDR 1,000,000 is IDR 63,362. By combining past trends with forward-looking predictions, the application enhances the accuracy of risk estimation, providing users with valuable insights for decision-making in asset risk management.

4. Discussion

The results obtained in this study demonstrate the effectiveness of the GRU model in predicting stock prices and assessing investment risks. The integration of predictive modeling with risk analysis provides a comprehensive approach for investors seeking data-driven decision-making tools. The system's ability to visualize historical data, predict future trends, and calculate risk values offers valuable insights for stock market analysis.

Furthermore, the graphical user interface enhances usability, allowing users to interact intuitively with the predictive model and risk assessment tools.

The modular design of the system ensures flexibility for future enhancements and improvements, making it adaptable to various stock market conditions. The results underscore the importance of integrating machine learning models with financial analysis tools to improve investment decision-making processes.

5. Conclusion

This research is a quantitative analysis focused on developing a GUI-based application for price prediction and risk of loss, specifically for stock asset data in Indonesia. In theory, price prediction is performed using the GRU model, while risk-of-loss prediction is performed using the VaR model with a historical simulation approach. The two models are combined into a hybrid model, enabling price prediction and risk of loss to be performed in a single analysis. Broadly speaking, the GUI application consists of four parts. The first part covers the application's initial display. The second part contains the data input and the pre-processing. The third part contains the data analysis process using the GRU model to obtain price predictions. The final part is a menu for predicting los risk values. The GUI application has proven effective, simplifying the analysis process through its simpler features and procedures. Furthermore, the application continues to provide accurate predictions, making it highly recommended for investors. One possible development of the application's features is to add options for the risk model used, for example, VaR with Monte Carlo Simulation. This addition is important to improve the accuracy of loss risk predictions.

References:

[1]. Chikwira, C., & Mohammed, J. I. (2023). The impact of the stock market on liquidity and economic growth: Evidence of volatile market. *Economies*, *11*(6), 155. Doi: 10.3390/economies11060155

[2]. Ridwan, M., Sadik, K., & Afendi, F. M. (2023). Comparison of ARIMA and GRU Models for High-Frequency Time Series Forecasting. *Scientific Journal of Informatics*, *10*(3), 389-400.

[3]. Meri Aryati, N. W., et al. (2024). Komparasi Metode LSTM dan GRU dalam Memprediksi Harga Saham. *Jurnal Media Informatika Budidarma*, 8(2), 1131. Doi: 10.30865/mib.v8i2.7342

[4]. Rivas, F., Sierra-Garcia, J. E., & Camara, J. M. (2025). Comparison of LSTM-and GRU-Type RNN networks for attention and meditation prediction on raw EEG data from low-cost headsets. *Electronics*, *14*(4), 707. Doi: 10.3390/electronics14040707

[5]. Maronrong, R., Hermastuti, P., & Muntahak, S. M. I. A. A. (2022). Analisis Value at Risk untuk Keputusan Investasi Menggunakan Simulasi Historis. *Jurnal Akuntansi STEI*, *5*(2), 21-30.

[6]. Idhom, M., Buditjahjanto, I. G. P. A., & Riyantoko, P. A. (2023). Antithesis of Human Rater: Psychometric Responding to Shifts Competency Test Assessment Using Automation (AES System). *Studies in Learning and Teaching*, *4*(2), 329-340. Doi: 10.46627/silet.v4i2.291

[7]. Asto Buditjahjanto, I. G. P., et al. (2022). An Automated Essay Scoring Based on Neural Networks to Predict and Classify Competence of Examinees in Community Academy. *TEM Journal*, *11*(4), 1694–1701. Doi: 10.18421/TEM114-34

[8]. Sinsomboonthong, S. (2022). Performance comparison of new adjusted min-max with decimal scaling and statistical column normalization methods for artificial neural network classification. *International Journal of Mathematics and Mathematical Sciences*, *2022*(1), 3584406. Doi: 10.1155/2022/3584406

[9]. Tripathy, N., Parida, S., & Nayak, S. K. (2023). Forecasting stock market indices using gated recurrent unit (GRU) based ensemble models: LSTM-GRU. *International Journal of Computer and Communication Technology*, *9*(1), 85-90.

[10]. Chen, C., Xue, L., & Xing, W. (2023). Research on improved GRU-based stock price prediction method. *Applied Sciences*, *13*(15), 8813. Doi: 10.3390/app13158813

[11]. Arfianti, U. I., et al. (2021). Sunspot number prediction using gated recurrent unit (GRU) algorithm. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *15*(2), 141-152. Doi: 10.22146/ijccs.63676

[12]. Hamzah, H., et al. (2023). Robust Stock Price Prediction using Gated Recurrent Unit (GRU). *International Journal of Informatics and Computation*, *5*(1), 29. Doi: 10.35842/ijicom.v5i1.56

[13]. Ghadimpour, M., & Ebrahimi, S. B. (2022). Forecasting financial time series using deep learning networks: evidence from long-short term memory and gated recurrent unit. *Iranian Journal of Finance*, *6*(4), 81-94. Doi: 10.30699/ijf.2022.313164.1286

[14]. Li, L. Y., et al. (2022). An Evaluation of the Forecast Performance of Neural Network. In *International Conference on Educational Technology and Administration*, 287-299. Springer Nature Singapore. Doi: 10.1007/978-981-99-2850-7_23

[15]. Jierula, A., et al. (2021). Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data. *Applied Sciences*, *11*(5), 2314. Doi: 10.3390/app11052314

[16]. Khairunisa, N. K., & Hendikawati, P. (2024). Long short-term memory and gated recurrent unit modeling for stock price forecasting. *Jurnal Matematika, Statistika dan Komputasi*, *21*(1), 321-333. Doi: 10.20956/j.v21i1.35930

[17]. Idhom, M., et al. (2023). Time Series Regression: Prediction of Electricity Consumption Based on Number of Consumers at National Electricity Supply Company. *TEM Journal*, *12*(3), 1575–1581. Doi: 10.18421/TEM123-39

[18]. Helena, P. Z., & Puspitasari, L. D. (2024). Value At Risk (VAR) Analysis using Historical and Monte Carlo Methods in Stock Prices of Bank CIMB Niaga, BSI, BJB, Bank Mega, and Bank Bukopin. International *Journal of Global Operations Research*, *5*(3), 170–175. Doi: 10.47194/ijgor.v5i3.320

[19]. Saputra, D., et al. (2023). Value at risk analysis using historical method and Monte Carlo simulation in banking and mining sector companies. *International Journal of Applied Management and Business*, *1*(1), 26-31. Doi: 10.54099/ijamb.v1i1.436

[20]. Kim, Y. S., et al. (2025). Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models. *Sustainable Cities and Society*, *118*, 105570. Doi: 10.1016/j.scs.2024.105570

[21]. Hussain, B., et al. (2021). Intelligent traffic flow prediction using optimized GRU model. *IEEE Access*, *9*, 100736-100746. Doi: 10.1109/ACCESS.2021.3097141

[22]. Kulisz, M., et al. (2024). Optimizing the neural network loss function in electrical tomography to increase energy efficiency in industrial reactors. *Energies*, *17*(3), 681. Doi: 10.3390/en17030681

[23]. Prathama, A. Y. (2018). Pendekatan ANN (Artificial Neural Network) untuk Penentuan Prosentase Bobot Pekerjaan dan Estimasi Nilai Pekerjaan Struktur pada Rumah Sakit Pratama. *Jurnal Teknosains*, *7*(1), 14. Doi: 10.22146/teknosains.30139

[24]. Jabed, M. I. K. (2024). Stock market price prediction using machine learning techniques. *American International Journal of Sciences and Engineering Research*, *7*(1), 1-6. Doi: 10.46545/aijser.v7i1.308

[25]. Musa, Y., Etuk, I. E., & Gulumbe, S. U. (2020). Efficiency and consistency assessment of value at risk methods for selected banks data. *Journal of Advances in Mathematics and Computer Science*, *35*(2), 1-11. Doi: 10.9734/jamcs/2020/v35i230245