
Research article

Investment risk forecasting model using extreme value theory approach combined with machine learning

Melina Melina^{1,*}, Sukono², Herlina Napitupulu² and Norizan Mohamed³

¹ Doctoral Program in Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang 45363, Indonesia

² Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Sumedang 45363, Indonesia

³ Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Kuala Terengganu 21030, Malaysia

* **Correspondence:** Email: melina21001@mail.unpad.ac.id; Tel: +6282120214656.

Abstract: Investment risk forecasting is challenging when the stock market is characterized by non-linearity and extremes. Under these conditions, VaR estimation based on the assumption of distribution normality becomes less accurate. Combining extreme value theory (EVT) with machine learning (ML) produces a model that detects and learns heavy tail patterns in data distributions containing extreme values while being effective in non-linear systems. We aimed to develop an investment risk forecasting model in the capital market with non-linear and extreme characteristics using the VaR method of the EVT approach combined with ML ($\text{VaR}_{\text{GPD-ML}(\alpha)}$). The combination of methods used is a multivariate time series forecasting model with RNN, LSTM, and GRU algorithms to obtain ML-based returns. The EVT method of the POT approach was used to model extremes. The VaR method was used for investment risk estimation. The backtesting method was used to validate the model. Our results showed that determining the threshold based on the normal distribution will identify extreme values with the ideal number, minimum bias, and distribution of extreme data following GPD. The $\text{VaR}_{\text{GPD-ML}(\alpha)}$ model was valid in all samples based on backtesting at $\alpha = 0.95$ and $\alpha = 0.99$. Generally, this model produces a greater estimated value of investment risk than the $\text{VaR}_{\text{GPD}(\alpha)}$ model at the 95% confidence level.

Keywords: backtesting; extreme value theory; GRU; LSTM; machine learning; multivariate; non-linear; RNN; VaR

Mathematics Subject Classification: 60G25, 62M10, 62M20, 62M45, 62P05, 90C90, 91G70

1. Introduction

In 2007–2008, the global financial crisis showed that the estimation of the value at risk (VaR) model using the assumption of distribution normality was considered less accurate [1]. After this crisis, the stock market continued to fluctuate along with events that caused the data distribution to become extreme. Over time, many extreme events have caused the stock market to fluctuate, such as the COVID-19 pandemic [2–4], geopolitical issues related to the Russia-Ukraine conflict [5,6], and the trade war between the United States and China [7,8]. These events cause high volatility in global stock markets. These conditions usually make it difficult to estimate investment risk because the data is not normally distributed, there are non-linear relationships between variables, and the data distribution contains extreme values.

The most common method for estimating the level of investment risk is the VaR model [9]. This method is considered less effective in uncertainty conditions when the data has high volatility, and the measurement of the risk level with the assumption of distribution normality becomes less accurate [10]. Although the concept is simple, the estimation of investment risk with the VaR method is very complex, where the very high volatility in the stock market causes the data to be characterized by heteroscedasticity, non-linear, and heavy-tailed. This method also cannot accurately detect extreme values and often fails to provide an appropriate measure of risk during periods of extreme stock price fluctuations [11]. The VaR method with extreme value theory (EVT) can be used to estimate the risk of an investment against data containing heavy tail patterns in stock return data [12–14]. EVT is a model that analyzes data deviations from the mean value of the probability distribution to detect and study heavy tail patterns of the distribution with extreme values [15]. It is usually used to model extreme events. Extreme events are defined as events that rarely occur but have a very large influence, so the lack of available data in modeling causes it to be difficult to identify the possibility of extreme events [16].

The EVT method has been applied in various fields where extreme values may appear such as the industrial field [17], the meteorological field [18], and the financial field [19–21]. The EVT method provides a powerful framework for studying the behavior of extreme observations formally. It focuses directly on the tail of the sample distribution, potentially performing better than other approaches in terms of predicting unexpected extreme changes. Models using a combination of several methods generally perform better than the application of a single method [22]. The estimation of investment risk with the VaR method of the EVT and GARCH combination was conducted by Bali & Neftci [23] who proposed a conditional extreme value approach to calculate VaR by determining the location and scale parameters of GPD as a function of past information. In addition, McNeil & Frey [24] proposed a method for estimating VaR and related risk measures describing the tails of the conditional distribution of heteroskedastic financial return series by combining the pseudo-maximum-likelihood adjustment of the GARCH model to estimate volatility and EVT to estimate the tails of the innovation distribution of the GARCH model. Singh et al. [25] modeled extreme market risk for the ASX-All Ordinaries index (Australia) and the S&P-500 Index (USA). The results showed that EVT can be applied to financial market return series to predict static VaR, CVaR or expected shortfall, and expected return levels as well as daily VaR using a dynamic approach based on GARCH (1,1) and EVT. Karmakar & Paul [26] applied the CGARCH-EVT-Copula model to estimate intraday VaR and CVaR portfolios using

high-frequency data. The backtesting method shows that the CGARCH-EVT-Copula type model has relatively better performance than other competing models.

However, research on investment risk forecasting by applying EVT combined with machine learning (ML) algorithms is very poorly explored. Several researchers have discussed the advantages of ML-based models in estimating risk have been conducted by Ren et al. [27] and this study presents an effective ML model for estimating extreme risks in the American stock market. Moreover, an enhanced AdaBoost algorithm, which combines class-weighted and time-weighted parameters designed to evaluate its performance in predicting extreme stock market crises, is proposed. The optimal model significantly improves the classification performance, especially for risk examples. Karim et al. [28] investigated the potential for extreme risk spillovers across advanced stock markets using a ML approach. A methodology that combines EVT with artificial neural networks to measure the likelihood and magnitude of risk spillovers among twenty-three major developed stock markets for the period covering January 1991 to July 2022 was used. The results revealed significant evidence of risk spillover across markets based on the level of trade integration between countries. The study offers important insights into the dynamics of risk spillover in the stock market and the benefits of incorporating ML techniques into risk management strategies. Furthermore, Blom et al. [29] aimed to improve methods for predicting the VaR of currency investments using ML by proposing a semiparametric and parsymonic model of risk value forecasting, based on quantitative regression and ML methods, combined with the generally available market price of options contracted from the foreign exchange rate interbank market. The proposed ensemble model achieves good estimation in all quantiles.

The large number and complexity of non-linear relationships in the data patterns that drive the dynamics of stock price movements make forecasting financial market behavior a very difficult task. Artificial neural networks (ANNs) are one of the most popular methods used by researchers to solve non-linear problems in various fields [30]. This method is also useful in multivariate cases. ML-based time series forecasting models categorized as linear and non-linear models are recurrent neural networks (RNNs) [31]. RNNs, long short-term memory (LSTM), and gated recurrent units (GRUs) models are non-parametric models, modeling time series data with good accuracy that do not require assumptions of stationarity, normality, and heteroscedasticity in their application [32]. Several studies that confirm the superiority of these models, such as the research of Ahmed et al. [33] predict sales data in financial markets more accurately using time series prediction using a combination of poly-linear regression with LSTM. Ricchiuti & Sperli [34] propose an advisory neural network framework using LSTM-based informative stock analysis for daily investment advice. Wang et al. [35] aimed to estimate the volatility of the Chinese stock market due to international crude oil shocks. Eight individual models, including ANNs, RNNs, LSTM, GRUs, multiple linear regression, support vector regression, bidirectional GRUs, and least absolute shrinkage and selection operator models, are constructed. Most of the combination models can effectively improve forecasting accuracy and conclude that most of the combination models are robust in volatility forecasting.

Oil and gold are two of the most actively traded commodities worldwide, and their price movements have important implications for economies and financial markets. Fluctuations in global oil prices increase stock market volatility and uncertainty as stock markets are particularly vulnerable to oil price fluctuations [36]. The collapse of global stock and oil markets during the COVID-19 pandemic caused investors to turn to safer assets to avoid losses, such as gold [37]. Currency exchange rates are one of the variables that affect fluctuations in the stock market. Such as the effect

of increased volatility of USD/CNY exchange rate uncertainty, resulting in increased volatility of uncertainty in the oil market to the Chinese stock market during bearish periods [38]. Exchange rate shocks can affect stock market returns when the market is sluggish; therefore, exchange rate flexibility has an important role in determining stock market returns [39,40].

The high volatility in the stock market makes it difficult to forecast investment risks because the data is not normally distributed, there is a non-linear relationship between variables, and the data distribution also contains extreme values. Based on the description above, the research gap identified is that there is no research on investment risk forecasting with the VaR method using EVT combined with ML. Motivated by this research gap, it is essential to build an investment risk forecasting model in the stock market that is non-linear and extreme using the VaR method of the EVT approach combined with ML. This is a model that can accommodate non-linear, extreme characters, and consider variables that affect stock fluctuations. The goal is to improve the accuracy of VaR predictions in extreme events. The model input is multivariate data. The data used in this study are historical data of closing prices of the Jakarta Stock Exchange (JKSE), the Kuala Lumpur Stock Exchange (KLSE), the Philippine Stock Exchange (PSEi), and the Stock Exchange of Thailand (SET). Variables that affect stock fluctuations are considered in this model, such as crude oil prices, world gold prices, and currency exchange rates against the US Dollar. Therefore, the model is more sensitive to stock market dynamics. The method used in this study involves using an ML-based multivariate time series forecasting model to forecast the closing price of the composite stock index, which is then used to derive ML-based returns. RNNs, LSTM, and GRUs algorithms are utilized due to the sequential nature of the time series data, a way to combine these sets of information was required. Of all the potential non-linear techniques, the most intuitive are RNNs, including LSTM and GRUs. These algorithms can capture historical trend patterns and predict future values with high accuracy [41]. These methods are highly sophisticated algorithms for time series and are the most widely applied artificial intelligence models in stock market prediction [42]. The EVT method with a peak-over threshold approach is used to model the extremes of the return data distribution. The threshold is determined based on the normal distribution. A $\text{VaR}_{\text{GPD-ML}(\omega)}$ model based on generalized Pareto distribution (GPD) is combined with ML for investment risk forecasting. Finally, the backtesting method is used to validate the proposed model.

The novelty of this research is that the investment risk forecasting model with the EVT approach is useful in univariate cases and combines with ML, which is reliable in multivariate instances [43]. We determine the threshold based on the normal distribution results in the optimal value of u to identify the ideal number of extreme values, minimum bias, and the distribution of extreme values that exceed the threshold following the GPD. This research provides new insights into the application of ML for investment risk forecasting and risk management. This research benefits financial institutions, investors, brokers, financial companies, governments, academics, and researchers.

2. Materials and methods

2.1. Dataset

We use multi-variable data. The dataset consists of historical closing price data of the JKSE, KLSE, PSEi, and SET composite stock indices. Variables that affect the price movement of the composite stock index are also needed, such as historical data on the closing price of crude oil,

currency exchange rates, and world gold prices. Daily historical stock data was downloaded from the www.finance.yahoo.com website for the period between January 7, 2019, and February 16, 2024.

2.2. Methods

Figure 1 shows the stages carried out in this study.

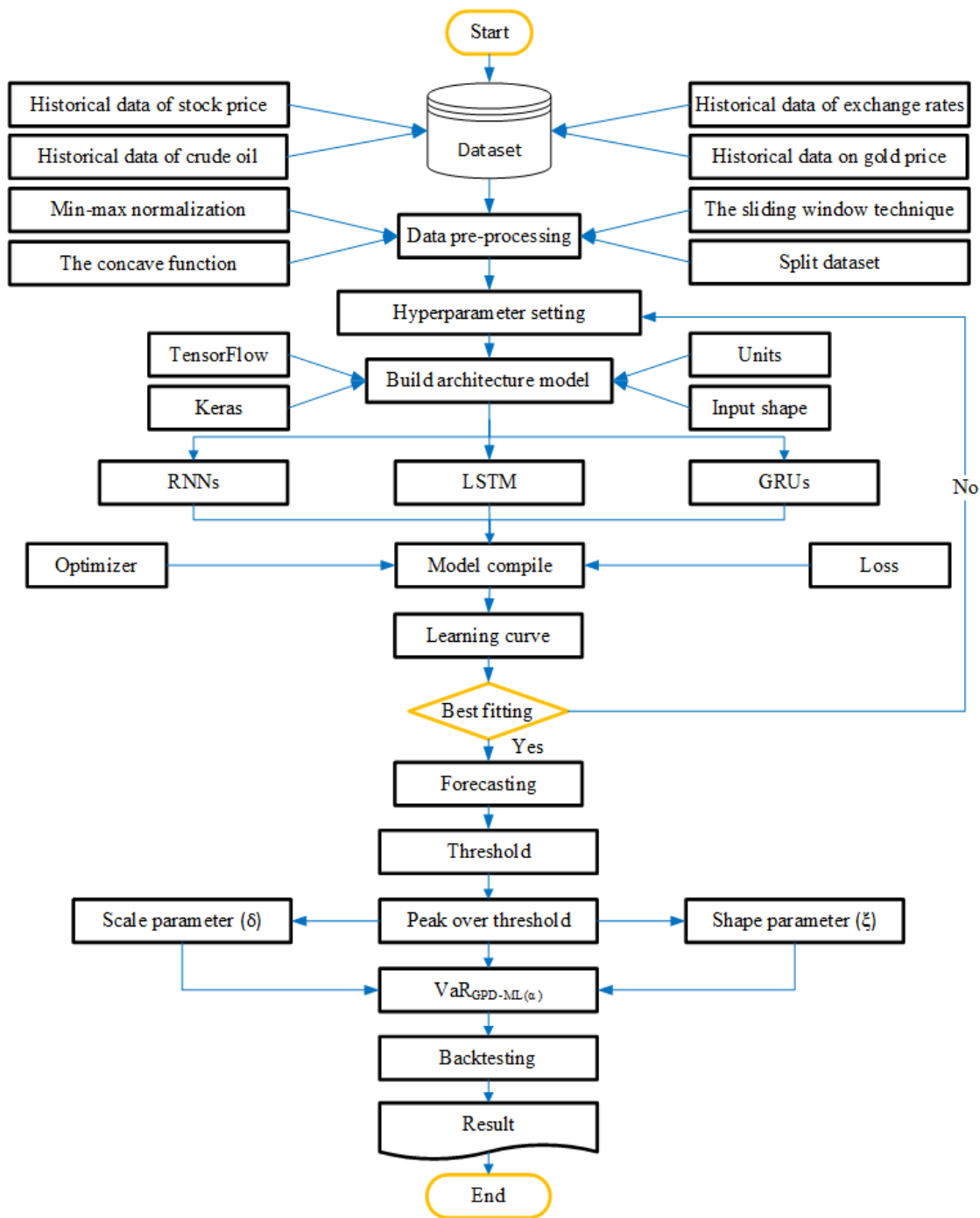


Figure 1. Research methodology.

In this study, computational calculations are conducted using RStudio software. Building an ML-based multivariate time series forecasting model in this study involves high-level neural network APIs such as ‘Keras’ and ‘TensorFlow’. Keras and TensorFlow packages are available in RStudio software. The first step is data pre-processing, building a model architecture to be used to forecast historical prices. The best model is selected based on the learning curve (LC). The model output is used to calculate the return. Extreme values are those that are above the threshold. Parameter estimation is performed to obtain shape and scale parameters used for VaR_{GPD-ML}(α). Backtesting is performed to validate the model.

2.2.1. The concave function

The data obtained tends to be incomplete because the stock market activities are closed on Saturdays, Sundays, and holidays. Missing values are estimated using a concave function. The missing value estimate is calculated using Eq (1) [44]:

$$\ddot{x}_i = \frac{x_{i-1} + x_{i+n}}{2}, \quad (1)$$

where \ddot{x}_i is the estimated value, x_{i-1} is the value available in the previous period, x_{i+n} is the value available in the next period.

2.2.2. Min-max normalization

Data normalization is used to scale data in the interval 0 to 1. This is generally useful to ease the computation of ML algorithms. Each value is converted to the interval [0,1] based on the maximum and minimum values of the dataset using Eq (2) [45,46]:

$$\bar{x} = \frac{x - \min(x)}{\max(x) - \min(x)}. \quad (2)$$

Then, to reverse the value to its original integer form, the min-max renormalization is used, by reversing the role of Eq (2). The min-max renormalization is written as Eq (3):

$$x = \bar{x} \cdot \max(x) - \bar{x} \cdot \min(x) + \min(x), \quad (3)$$

where x is the original data value, $\min(x)$ is the value of the minimum of the data set, $\max(x)$ is the maximum value of the data set, and \bar{x} is the scaling result data.

2.2.3. Artificial neural networks

ANNs are methods that exploit the architecture of the human brain to perform tasks that conventional algorithms cannot. ANNs were first introduced by McCulloch & Pitts [47]. ANNs can learn and model non-linear relationships between variables. This is achieved by connecting neurons in various patterns, allowing the output of some neurons to become inputs for others, and applying an activation function before producing the output. Mathematically, the ANNs model is written as a triplet $(\varphi; x; w)$ as in Figure 2 [48].

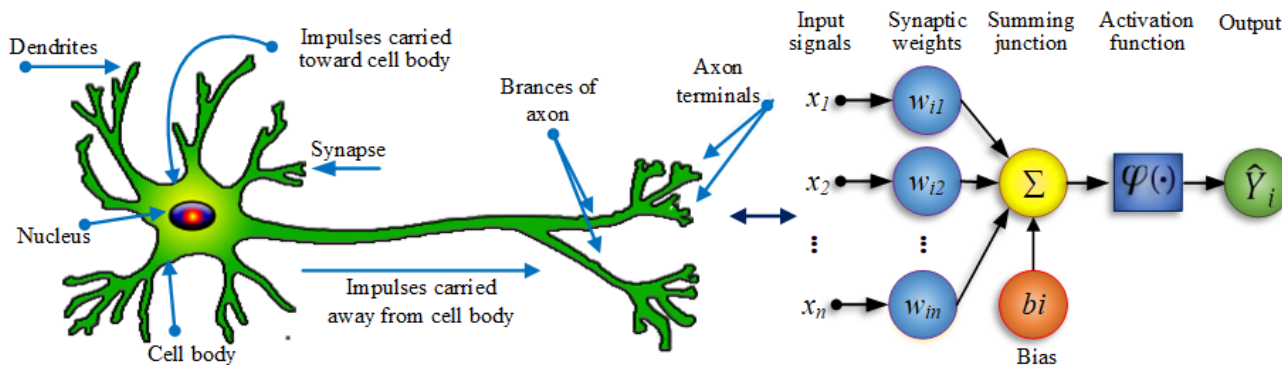


Figure 2. ANNs concept.

Figure 2 shows the concept of ANNs, where the dendrites of biological neural networks represent the input into the ANNs, the cell nucleus is the node, the synapse represents the weights, and the axon represents the output. Based on Figure 2, the ANNs are written as Eq (4):

$$\hat{Y}_i = \varphi(b_i + \sum_{j=1}^n w_{i,j}x_j), \tag{4}$$

where $\{x_1, x_2, x_3, \dots, x_n\}$ are the inputs, $\{w_{i1}, w_{i2}, w_{i3}, \dots, w_{in}\}$ are the weight of neuron i , b_i is the bias, $\varphi(\cdot)$ is the activation function, and \hat{Y}_i is the output.

2.2.4. The recurrent neural networks (RNNs)

One of the reliable time series forecasting methods in univariate and multivariate cases is recurrent neural networks. The RNNs are a type of artificial neural network designed to process sequential data or time series data. RNNs can store the memory of previous information in time series data. With this ability to remember, RNNs can recognize data patterns well, so they can make accurate predictions. They work similarly to the Jordan networks [49] and Elman networks [50]. Storing information from the past is used by repeating the architecture and performing mathematical calculations sequentially, where the output data is stored and reused as input data. Figure 3 shows compressed and unfolded basic RNNs [51].

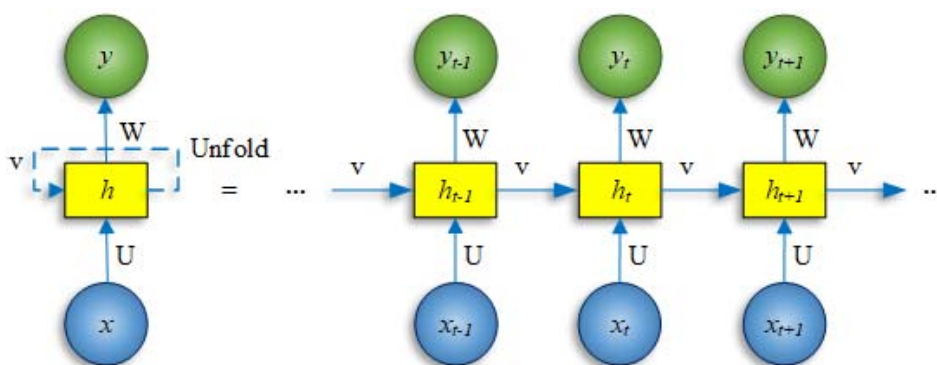


Figure 3. Basic the recurrent neural networks.

In Figure 3, the left image is a circuit diagram of a recurrent connection labeled v , where the RNNs are in an unrolled position into the full network (compressed). The right image shows the RNNs that have been unfolded into the full network so that the sequence becomes complete. Mathematically the hidden state is calculated by Eq (5):

$$h_t = \varphi_h(w_h x_t + U_h h_{t-1} + b_h), \quad (5)$$

and the output value y_t is obtained using Eq (6):

$$y_t = \varphi_y(w_y h_t + b_y), \quad (6)$$

where x_t is the input vector, h_t is the hidden state, y_t is the output vector, b_h is the bias, φ is the activation, and U, W are the parameter matrices.

2.2.5. The long short-term memory (LSTM)

As the circuit size and complexity increase, issues such as gradient bursting, gradient missing, and computational inefficiency have added to the complexity of using the RNNs. To address these issues, several approaches have been proposed such as the neural network architecture proposed by Lang et al. [52] and LSTM. The LSTM was introduced by Hochreiter & Schmidhuber as a solution to the missing gradient problem [53,54]. The LSTM has memory cells that effectively act as long-term memory storage and can update information. Figure 4 shows the memory unit of the LSTM.

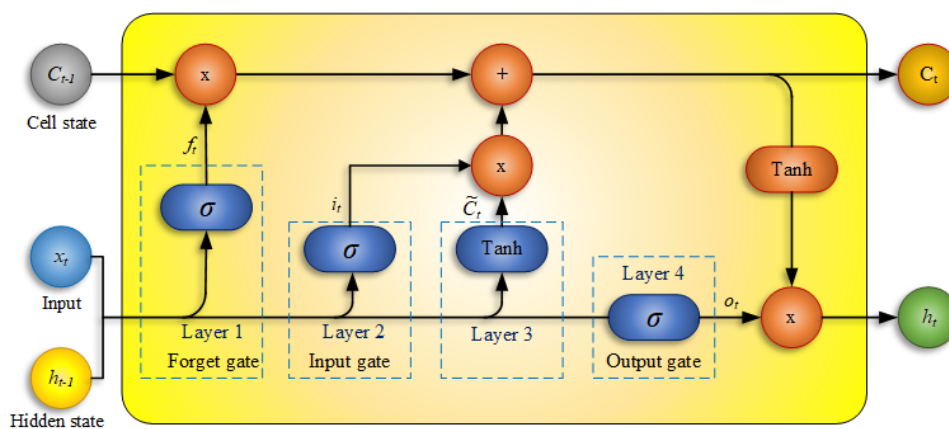


Figure 4. The LSTM unit memory.

The LSTM consists of three gates, namely the input gate, forget gate, and output gate. The LSTM performs four steps in performing its processes. First is a forget gate that decides whether the input and output will be passed to the cell state or not. These values are passed to a sigmoid function, outputting with values of 0 and 1. A value of 1 means the previous information is kept, otherwise, a value of 0 means the previous information is deleted. This forget gate value is written as Eq (7) [55]:

$$f_t = \sigma_g(w_f x_t + U_f h_{t-1} + b_f). \quad (7)$$

The second is the input gate, which serves to determine how important the current input is to the process of forming a new cell \tilde{C}_t , as shown in the equation below [56]:

$$i_t = \sigma_g(w_i x_t + U_i h_{t-1} + b_i). \quad (8)$$

$$\tilde{c}_t = \tanh(w_c x_t + U_c h_{t-1} + b_c), \quad (9)$$

and this new cell state is now a long-term memory state and will be used in subsequent processes.

Third is the cell state gate to update the old cell state C_{t-1} with the new cell state \tilde{c}_t , which is mathematically written as the equation below:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{c}_t. \quad (10)$$

The multiplication of f_t with C_{t-1} aims to remove the information that has been determined at the forget gate stage, and then a new value i_t is added and multiplied by \tilde{c}_t .

Fourth is the output gate that decides which cell state to generate. This output will be based on the cell state, with two activation functions sigmoid and tanh. The sigmoid activation function decides what information can go through the output gate and then the cell state is multiplied after being activated by the tanh activation function. It is mathematically written as the following equation:

$$o_t = \sigma_g(w_o x_t + U_o h_{t-1} + b_o), \quad (11)$$

and

$$h_t = o_t \odot \tanh(C_t), \quad (12)$$

where i_t is input gate time t , C_t is new cell state time t , \tilde{C}_t is new cell state time t , o_t is output gate, f_t is forget gate, h_t is hidden state time t , w_i is input weight, w_c is candidate layer weight, w_f is forget layer weight, σ_g is the activation of the sigmoid function, tanh is the activation of hyperbolic tangent function, and w_o is output layer weight.

2.2.6. The gated recurrent units (GRUs)

The GRUs are a variation of LSTM and both have a similar design. The GRUs support hidden states, using an update gate and reset gate to solve the vanishing gradient problem. The GRUs have a special mechanism that determines that the hidden state (h_t) must be updated and h_t must be reset. The cell diagram of GRUs with reset gate and update gate is shown in Figure 5 [57].

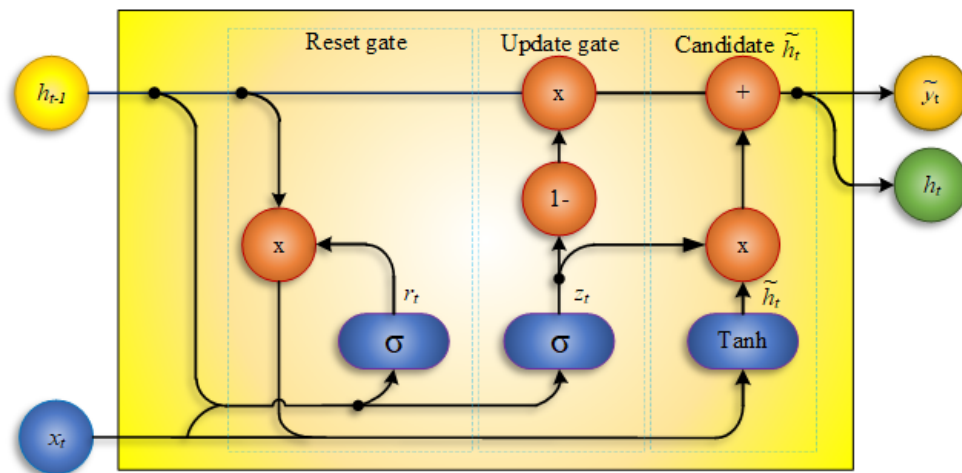


Figure 5. The diagram of the GRUs cell.

The GRUs have only one h_t ; therefore, the GRUs architecture is simpler than LSTM. Illustrating the GRUs process shown in Figure 5, mathematically the reset gate is written as the following equation [58]:

$$r_t = \sigma(w_r x_t + U_r h_{t-1} + b_r), \quad (13)$$

and the update gate equation is as follows:

$$z_t = \sigma(w_z x_t + U_z h_{t-1} + b_z), \quad (14)$$

the hidden state candidate equation is as follows:

$$\tilde{h}_t = \tanh(W_h x_t + U_h [r_t \odot h_{t-1}] + b_h), \quad (15)$$

the hidden state equation is as follows:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t, \quad (16)$$

where r_t is the reset gate, z_t is the update gate, x_t is the input vector, \tilde{h}_t is the candidate hidden state, h_t and \tilde{y}_t are the output vector, b_h is the bias, σ is a logistic activation functions, \tanh is hyperbolic tangent activation functions, W_r , W_z , and W_h are weight parameters, and Operator \odot is the Hadamard product.

2.2.7. Model accuracy

The accuracy of the model used must be evaluated to determine how good the model output is. The model accuracy evaluation used is the mean absolute error (MAE), root mean squared error (RMSE), and the mean absolute percentage error (MAPE) [59]. Residual or forecasting error is the difference between the actual value and the forecasting value, written as Eq (17) [60]:

$$e_i = Y_i - \hat{Y}_i. \quad (17)$$

MAE is commonly used to measure prediction error in time series analysis. MAE is the average of the absolute difference between the actual value and the forecast value. MAE is written as the

following equation:

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_i|. \quad (18)$$

RMSE calculates the average of the squared difference between the actual and predicted values, and then the square root is taken. The smaller the RMSE value, the better the performance of the model. RMSE is written as Eq (19) [61]:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2}. \quad (19)$$

A measure of forecasting accuracy used in addition to MAE and RMSE is MAPE, where MAPE calculates the average prediction error as a percentage of the actual value (20):

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{e_i}{Y_i} \right|, \quad (20)$$

where e_i is the prediction error, n is the amount of data, Y_i is the actual value, and \hat{Y}_i is the forecast value.

The judgment of forecasting accuracy is, $MAPE \leq 0.1$ judgment: Highly accurate, $0.1 \leq MAPE \leq 0.2$ assessment: Good forecast, $0.2 < MAPE \leq 0.5$ assessment: Reasonable forecast, and $MAPE > 0.5$ assessment: Inaccurate forecast [62].

2.2.8. Return

Return is the rate of profit or loss obtained by investors from investment. Return is calculated by Eq (21):

$$X_i = \frac{P_i}{P_{i-1}} - 1, \quad (21)$$

where X_i is the return at time i , P_i is the price at time i , and P_{i-1} is the price at the previous period.

2.2.9. Extreme value theory

The EVT method is used to identify, detect, and study heavy tail patterns in data distributions. The basis of the EVT approach focuses on statistical behavior [63].

Theorem 1. $X_1, X_2, X_3, \dots, X_n$ are a sequence of independent random variables with common distribution function F , and let:

$$M_n = \max\{X_1, X_2, X_3, \dots, X_n\}, \quad (22)$$

where n is the number of observations, M_n is the maximum value over n observations.

Theorem 2. If there exists a constant sequence $\{a_n > 0\}$ and $\{b_n\}$ such that:

$$P_r = \left\{ \frac{M_n - b_n}{a_n} \leq Z_n \right\} \rightarrow G(z) \quad \text{as } n \rightarrow \infty, \quad (23)$$

where G is a non-degenerate distribution function, then G belongs to the families:

$$\text{Type of Gumbel family: } G(z) = \exp \left\{ -\exp \left[-\left(\frac{z-b}{a} \right) \right] \right\}, \quad -\infty < z < \infty; \quad (24)$$

$$\text{Type of Fréchet family: } G(z) = \begin{cases} 0, & z \leq b, \\ \exp \left\{ -\left(\frac{z-b}{a} \right)^{-\alpha} \right\}, & z > b; \end{cases} \quad (25)$$

$$\text{Type of Weibull family: } G(z) = \begin{cases} \exp \left\{ -\left(\frac{z-b}{a} \right)^\alpha \right\}, & z < b, \\ 1, & z \geq b, \end{cases} \quad (26)$$

for parameters, $\alpha > 0$, b and in the case of families Fréchet and Weibull, $\alpha > 0$.

Each family type has scale (a) and location (b) parameters. The Fréchet and Weibull families have the shape parameter α . These three families have different forms of behavior towards extreme value behavior, corresponding to different forms of tail behavior for the distribution function F of X_i . However, in practice, only one of them is chosen to be applied to estimate the most suitable distribution parameters. Therefore, an appropriate method is needed to select the type of one of these three family types that best fits the data at hand. Finally, a better analysis is offered to combine all three into one model that has another form of the distribution function. For a non-degenerate distribution function G , then G is the generalized extreme value (GEV) of the distribution. Applying the Fisher-Tippett theorem to the distribution of extreme value sample data based on the GEV distribution with cumulative distribution function [64,65]:

$$G(z) = \begin{cases} \exp \left(- \left[1 + \xi \left[\frac{z-\mu}{\delta} \right]^{-\frac{1}{\xi}} \right] \right), & \text{if } \xi \neq 0, \\ \exp \left[-\exp \left[- \left[\frac{z-\mu}{\delta} \right] \right] \right], & \text{if } \xi = 0, \end{cases} \quad (27)$$

where μ is the location parameter, δ is the scale parameter, and ξ is the shape parameter. Defined on the set $z: \left\{ 1 + \xi \left(\frac{z-\mu}{\delta} \right) > 0 \right\}$, its parameters fulfill the requirements $-\infty < \mu < \infty$, $\delta > 0$, and $-\infty < \xi < \infty$.

The location parameter (μ), scale parameter (δ), and shape parameter (ξ) are estimated using the maximum likelihood estimation (MLE) method [66]. The result is an estimate of the of the μ parameter, δ parameter, and ξ parameter. GEV is divided into three types based on the value of ξ , including the Gumbel distribution when $\xi = 0$, the Fréchet distribution at $\xi > 0$, and the Weibull distribution at $\xi < 0$. The value of ξ has an infinite limit, the larger the value of ξ , the greater the possibility of extreme values, usually leading to heavier distributions. The type with the heaviest tail is the Fréchet distribution. This is shown by the explanation that when $\xi < 0$, then the extreme values are finite, and if $\xi > 0$ then the extreme values are infinite.

There are two EVT methods, namely block maxima (BM) and peak over threshold (POT). The BM method identifies extreme values based on the maximum value of data grouped by specific blocks. This method produces only one extreme value in each block, ignoring the role of other data. Therefore, this method will require a large amount of data for the extreme value identification

process. This method tends to be considered inefficient. The fundamental difficulty in extreme value analysis is the limited amount of data for model estimation. The POT approach is considered to be more efficient in data utilization [67]. The POT method identifies extreme data behavior patterns based on values that exceed the threshold (u). Data that exceeds u is an extreme value. The value of u is estimated by the mean residual life plot (MRLP) method [68].

2.2.10. VaR GPD combined with machine learning

VaR is the maximum expected loss value of investment in a certain period with a certain level of confidence from the concept of a normal curve. There are two types of VaR values, namely a positive value means that the investment activities carried out generate profits. Moreover, a negative VaR value indicates that the investment activities carried out experience a loss.

Let X_1, X_2, \dots, X_n denote the actual return sequence on time $1, 2, \dots, n$. Suppose also X'_1, X'_2, \dots, X'_n is the sequence of returns predicted using the ML model on time $1, 2, \dots, n$. The ML-based returns are the sum of the actual return and the predicted return:

$$\bar{X}_i = X_i + X'_i, \quad i = 1, 2, \dots, n. \quad (28)$$

Thus, the ML-based returns sequence can be denoted as $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n$. Extreme value can be defined as the maximum of a random variable $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_n$, which are assumed to be independent and identically distributed (iid). By replacing the variable X with \bar{X} , the POT model approach focuses on estimating the distribution function F_u of the values of \bar{X} that exceed u . For a random variable \bar{X} with an unknown distribution function F , the conditional excess distribution function is defined as:

$$F_u(\bar{y}) = P(\bar{X} - u \leq \bar{y} \mid \bar{X} > u) = \frac{F(\bar{y}+u) - F(u)}{1 - F(u)} = \frac{F(\bar{x}) - F(u)}{1 - F(u)}, \quad (29)$$

for $0 \leq \bar{y} < \bar{x}_F - u$, where $\bar{x}_F \leq \infty$ is the right endpoint of the underlying distribution F .

Theorem 3. *Balkema & Haan, (1974), and Pickands., (1975). For a large class of underlying distribution functions F and large u , F_u is well approximated by the GPD [69,70]:*

$$\lim_{u \rightarrow \sigma} \sup_{0 \leq \bar{x} < \bar{x}_F - u} |F_u(\bar{y}) - G_{\xi, \delta}(\bar{y})| = 0, \quad (30)$$

where u is threshold, F_u is the distribution of the excess value over a u , and \bar{x}_F is the right endpoint of the underlying distribution F .

The distribution function $G_{\xi, \delta}(\bar{y})$ is the GPD given by [71]:

$$G_{\xi, \delta}(\bar{y}) = \begin{cases} 1 - \left[1 + \xi \frac{(\bar{x} - u)}{\delta} \right]^{-\frac{1}{\xi}}, & \text{if } \xi \neq 0, \\ 1 - \exp \left[-\frac{(\bar{x} - u)}{\delta} \right], & \text{if } \xi = 0, \end{cases} \quad (31)$$

where δ is the scale parameter, ξ is the shape parameter, $\delta > 0$, and $(\bar{x} - \mu) \geq 0$.

In the POT method, parameter estimation using MLE to obtain the parameter δ , and parameter

ξ required in the calculation of $\text{VaR}_{\text{GPD-ML}(\alpha)}$. If n is the number of ML-based returns observations and N_u is the number of observations that exceed u , then the empirical estimate of $F(u)$ is equal to $\frac{n-N_u}{N_u}$ [72]. Empirical distribution function to estimate $\bar{F}(u)$:

$$\bar{F}(u) = \frac{N_u}{n} = 1 - F(u), \quad (32)$$

by substituting Eq (30) into Eq (29), the approximation $F(\bar{x})$, for $\bar{x} > u$, can be obtained

$$F_u(\bar{y}) = \frac{F(\bar{x}) - F(u)}{1 - F(u)}. \quad (33)$$

$$F(\bar{x}) = F(u) + (G_{\xi, \delta}(\bar{y}))\bar{F}(u). \quad (34)$$

$$F(\bar{x}) = 1 - F(u) + (G_{\xi, \delta}(\bar{y}))\bar{F}(u), \quad (35)$$

similar

$$F(\bar{x}) = 1 + \bar{F}(u)[G_{\xi, \delta}(\bar{y}) - 1]. \quad (36)$$

$$F(\bar{x}) = 1 + \frac{N_u}{n} \left[1 - \left(1 + \xi \frac{(\bar{x} - u)}{\delta} \right)^{-\frac{1}{\xi}} - 1 \right], \quad (37)$$

similar

$$F(\bar{x}) = 1 - \frac{N_u}{n} \left[\left[1 + \xi \frac{(\bar{x} - u)}{\delta} \right]^{-\frac{1}{\xi}} \right]. \quad (38)$$

The high quantile estimator the VaR, for $\alpha \geq \frac{n-N_u}{n}$ can be obtained by reversing the Eq (38):

$$\alpha = 1 - \frac{N_u}{n} \left[1 + \xi \frac{q_{\alpha(F)} - u}{\delta} \right]^{-\frac{1}{\xi}}. \quad (39)$$

$$\left[1 + \xi \frac{q_{\alpha(F)} - u}{\delta} \right]^{-\frac{1}{\xi}} = \frac{n}{N_u} (1 - \alpha). \quad (40)$$

$$\xi \frac{q_{\alpha(F)} - u}{\delta} = \left[\frac{n}{N_u} (1 - \alpha) \right]^{-\xi} - 1. \quad (41)$$

Finally

$$VaR_{GPD-ML(\alpha)} = q_{\alpha}(F) = u + \frac{\hat{\delta}}{\hat{\xi}} \left[\left[\frac{n}{Nu} (1 - \alpha) \right]^{-\hat{\xi}} - 1 \right], \quad (42)$$

where n is the number of ML-based returns observations, Nu is the observations that exceed the threshold, $\hat{\delta}$ is the scale parameter estimate, and $\hat{\xi}$ is the shape parameter estimate, α is the confidence level of VaR.

The simplest method for estimating investment risk is to use the empirical quant of the return distribution. This method is often called VaR historical simulation ($VaR_{HS(\alpha)}$) [73]. For the analysis, we use the $VaR_{HS(\alpha)}$ method as a comparison method.

2.2.11. Backtesting

Backtesting is conducted to determine the validity of the VaR model, using the method developed by Kupiec [74] which is based on the failure rate with the log-likelihood ratio (LR) approach. The exact likelihood ratio statistic is written as Eq (43):

$$LR = -2\ln[(1 - p)^{n-f} p^f] + 2\ln \left(1 - \frac{f}{n} \right)^{n-f} \left(\frac{f}{n} \right)^f, \quad (43)$$

where p is the confidence level for VaR, n is the number of data, and f is the number of failures.

The VaR estimation results will be retested with the actual return value if the $X_i > VaR$ value is considered failed. A model is rejected if it produces too few or too many failures.

3. Results

To achieve the objectives of this research, the methodological stages are applied in detail. The dataset used includes the closing price of the composite stock index, crude oil price, currency exchange rate against the dollar, and world gold price. The first step is data pre-processing, followed by data analysis based on descriptive statistics. Stock returns are calculated based on the closing price data of the composite stock index, further forecasting the closing price of the combined stock index with variables that affect stock fluctuations using RNNs, LSTM, and GRUs. The results of forecasting the closing price of the composite stock index are used to obtain the ML-based \bar{X}_i . QQ plots and histograms are used to observe heavy tails in the data distribution of X_i and \bar{X}_i . Extreme values are identified using the POT method. Parameter estimation is employed to calculate $VaR_{GPD-ML(\alpha)}$. Furthermore, the model is validated using the backtesting method. Finally, the conclusion of this study is outlined based on the research results.

3.1. Data pre-processing

Data pre-processing is done because the dataset may contain noise, incomplete, inconsistent, or missing data. The aim is to produce a complete dataset and improve the quality of the dataset. Initial field studies show that the stock market operating time is a working day, meaning any day other than Saturday, Sunday, or public holidays. If missing data is found between Monday and Friday, the value of the data is estimated using Eq (1). Table 1 presents the pre-processing of the data.

Table 1. Data pre-processing.

Variables	Symbol	Period	Missing	Observation
X ¹	IDX Composite (JKSE)	1260	75	1335
	FTSE Bursa Malaysia (KLSE)	1253	82	1335
	PSEi Index (PSEi)	1247	83	1335
	SET Index (SET)	1239	96	1335
X ²	Crude oil	1289	46	1335
X ³	Gold	1289	46	1335
X ⁴	Exchange rate (USD/IDR)	1334	1	1335
	Exchange rate (USD/MYR)	1333	2	1335
	Exchange rate (USD/PHP)	1334	1	1335
	Exchange rate (USD/BTH)	1334	1	1335

Applying RNNs, LSTM, and GRUs algorithms to data with drastic ranges will produce less accurate output. The min-max normalization method changes the data set to a scale of [0,1], which is carried out using Eq (2). This process is carried out because some ML algorithm activation functions can only process normalized data. Apart from that, min-max normalization will reduce the complexity of the ML algorithm.

3.2. Machine learning-based returns

In this research, the calculation of \bar{X}_i begins with the formation of a new return data distribution. This data distribution was obtained from the results of forecasting return values from the best ML-based multivariate time series forecasting model. Therefore, Eq (21) is modified by considering the residual value of the ANNs and written as Eq (44):

$$X'_i = \frac{(X_i - e_i)}{(X_{i-1} - e_{i-1})} - 1, \quad (44)$$

if Eq (17) is substituted into Eq (44), the following is obtained:

$$X'_i = \frac{(X_i - \{Y_i - \hat{Y}_i\})}{(X_{i-1} - \{Y_{i-1} - \hat{Y}_{i-1}\})} - 1, \quad (45)$$

because $X_i = Y_i$, $X_{i-1} = Y_{i-1}$ similar

$$X'_i = \frac{\hat{Y}_i}{\hat{Y}_{i-1}} - 1, \quad (46)$$

where X_i is the actual return period i , X'_{i-1} is the return forecasting value, X_{i-1} is the actual return in the previous period, e_i is the residual in period i , and e_{i-1} is the residual in the previous period, and \hat{Y}_i is the forecasting result in period i , \hat{Y}_{i-1} is the forecasting result in the previous period.

The ML-based \bar{X}_i model is the X_i value from Eq (21) plus the X'_i value from Eq (46) which contains the non-linear elements of the ANNs output as shown in Eq (28).

3.3. Data analysis

The dataset of this research is the historical closing price of the JKSE, KLSE, PSEi, and SET composite stock indices. In addition, we use historical data on the closing price of crude oil, the closing price of gold, and the closing price of each country's exchange rate. Figure 6 shows the close price fluctuations of JKSE, KLSE, PSEi, and SET.

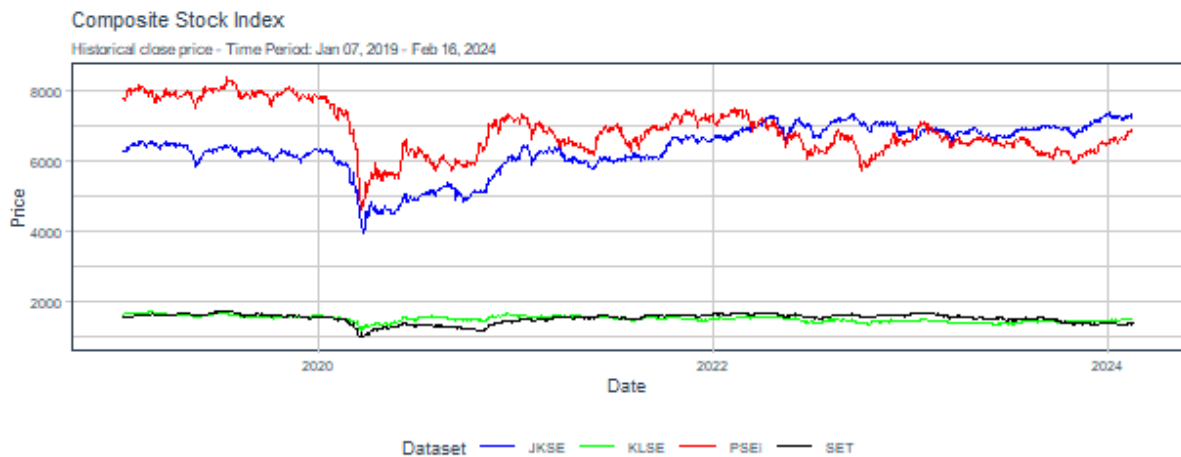


Figure 6. Composite stock index (CSI).

Figure 6 is a graph of the closing price movements of the JKSE, KLSE, PSEi, and SET composite stock indices from four countries, namely Indonesia, Malaysia, the Philippines, and Thailand (adjusted for each country's currency). This graph shows that there was a decline in closing prices on all stock exchanges in March 2020, when the COVID-19 pandemic was first announced by WHO. March 2020 was the period of the highest volatility in 2020. This global pandemic triggered extreme changes in stock market performance, causing the composite stock index to drop dramatically, and fluctuate greatly in line with the development of the pandemic. This condition illustrates that the stock market is characterized by high volatility and non-linear characteristics. Table 2 below presents the descriptive statistics of the closing prices of the CSIs of the four countries.

Table 2. Descriptive statistics of the main data.

Symbol	Min	Median	Mean	Max	SD	TNNs test	ADF test
JKSE	3938	6445	6339	7360	688.892	0.05489	0.4456
KLSE	1220	1531	1529	1731	85.022	0.03132	0.0391
PSEi	4623	6721	6844	8365	689.973	0.20040	0.2668
SET	1024	1580	1541	1741	133.114	0.00774	0.4637

Table 2 shows that all standard deviation (SD) of the composite stock indices are smaller than the mean value, indicating that the data is homogeneous. The results of the min and max analysis, all of the composite stock indices fluctuate greatly, especially the KLSE has a large enough difference between min and max, which is 70.48%, while JKSE, PSEi, and SET have smaller min and max ranges, below 60%.

Terasvirta neural networks (TNNs) tests are conducted to characterize the data with the `terasvirta.test` function available in the RStudio application. The results show that KLSE and SET are non-linear in character, with a p-value < 0.05. In contrast, the JKSE and PSEi have linear data characteristics. These results illustrate that the dataset of close price fluctuations of the composite stock index used represents data with linear and non-linear characteristics.

Stationary test using augmented Dickey-Fuller (ADF) Test. Only KLSE was identified as stationary with a p-value < 0.05. On the other hand, JKSE, PSE, and PSEi produced p-values > 0.05. This indicates that the time series data is not stationary.

3.4. Time series forecasting model based on machine learning

The most important thing to do when building an ML-based multivariate time series forecasting model is to organize the data in such a way that the model knows the order of the input variable data, as well as the target variable data during the learning and testing stages. In this study, the sliding window (SW) technique is used to obtain patterns in time series data. This technique allows data analysis to identify valuable patterns in time series data while reducing the complexity of the algorithms used. The SW technique generally makes the forecasting model produce much better accuracy because the model is retrained after shifting only one block of data [75]. Figure 7 shows the SW used in this study.

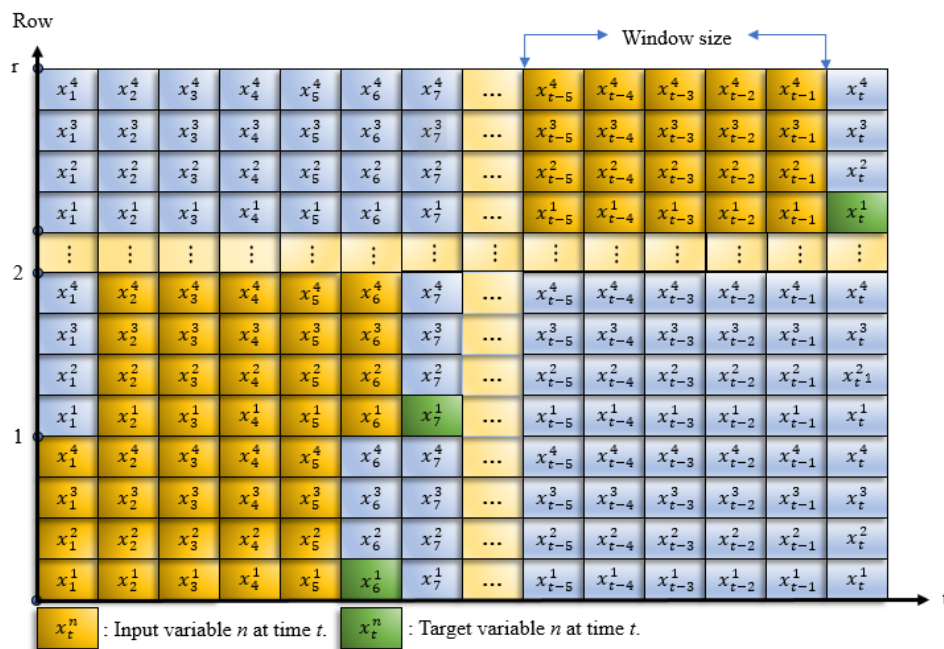


Figure 7. The sliding window is multivariate.

The window size (k) is set according to the working days, which consist of five days a week, so $k = 5$. In the ML-based multivariate time series forecasting model, the value of $k = 5$ means that every five periods of data will obtain the input variable x_t^n , and the next one period of data will obtain the output variable y_t^n or target data. Parameter n is the number of variables used. This study uses four variables for model input, namely x_t^1 is the historical close of the composite stock index, x_t^2

is the world crude oil price, x_t^3 is the gold price, and x_t^4 is the exchange rate of each country against the US Dollar. This multivariate SW process runs continuously until the data period ends (t).

Min-max normalization to produce the number ranging between 0 and 1 using Eq (2) is applied to the dataset. The dataset is divided by a ratio of 80:20, 80% learn data, and 20% test data. The training dataset functions to determine weights and biases when training the model, while the testing dataset is used as an evaluator of the model being built. The validation composition of each dataset is set at the time of the hyperparameter setting by providing a split validation value. In building the ML-based multivariate time series forecasting model, the input batch frame is formed in an array shape (sample-batch-size, window size, number of variables, target). Sample-batch-size is the number of r partitioned samples, where $r = \text{length}(\text{data}) - k$, the number of variables is $n = 4$, and the target is the value to be used as the output reference, in this case, target = 1. Based on the 80:20 data composition, the number of training data samples is 1068, i.e., period 1 to period 1068, and the number of test data is 267 periods, starting from period 1069 to period 1335. The total partitions (r) generated are 1063 for the learn data and 262 for the test data. So, the input frame for the training data is array shape (1063, 5, 4, 1) and the input frame for the test data is array shape (262, 5, 4, 1).

Experiments were conducted to find the best model architecture by changing the values of hyperparameter variables. Hyperparameter settings are performed to obtain best-fitting model results when building the model architecture. The optimal number of hidden layers, the optimal number of units in each layer, the activation function, the appropriate lost validation function, and the best optimizer greatly affect the results of the learning process. Epoch is the number of iterations during model training. The loss function is used to measure the effectiveness of the model in making predictions at each epoch, seen from the error difference between prediction and actual, in this case, the mean squared error (MSE) is set as the loss function. Root mean square propagation (RMSprop) and “Adam” are used as optimization algorithms. The following Table 3 shows the hyperparameter settings.

Table 3. Hyperparameter setting.

Parameter	Value	Status
Array shape (train)	(1063, 5, 4, 1)	Fixed
Array shape (test)	(262, 5, 4, 1)	Fixed
ANNs type	RNNs, LSTM, and GRUs	Experiment
Hidden layers (HL)	1, 2	Experiment
Neurons / Units	10, 50	Experiment
Optimizer	Adam, RMSprop	Experiment
Batch-size	32, 64	Experiment
Validation split	0.2	Fixed
Max epochs	20, 30, 50	Experiment

The LC is used to evaluate the model training process, if the best-fitting condition has not been achieved, the hyperparameter is reset until best-fitting is achieved. After repeated experiments with changing the hyperparameter settings, The parameter set used was HL = 2 with the number of neurons = 50 each HL, loss select MSE, batch size = 32, and max epochs = 20. Based on these settings, the LC diagnostics of all models resulted in best-fitting models. The following shows the consecutive LCs of the selected models based on the best accuracy value, as shown in Figure 8.

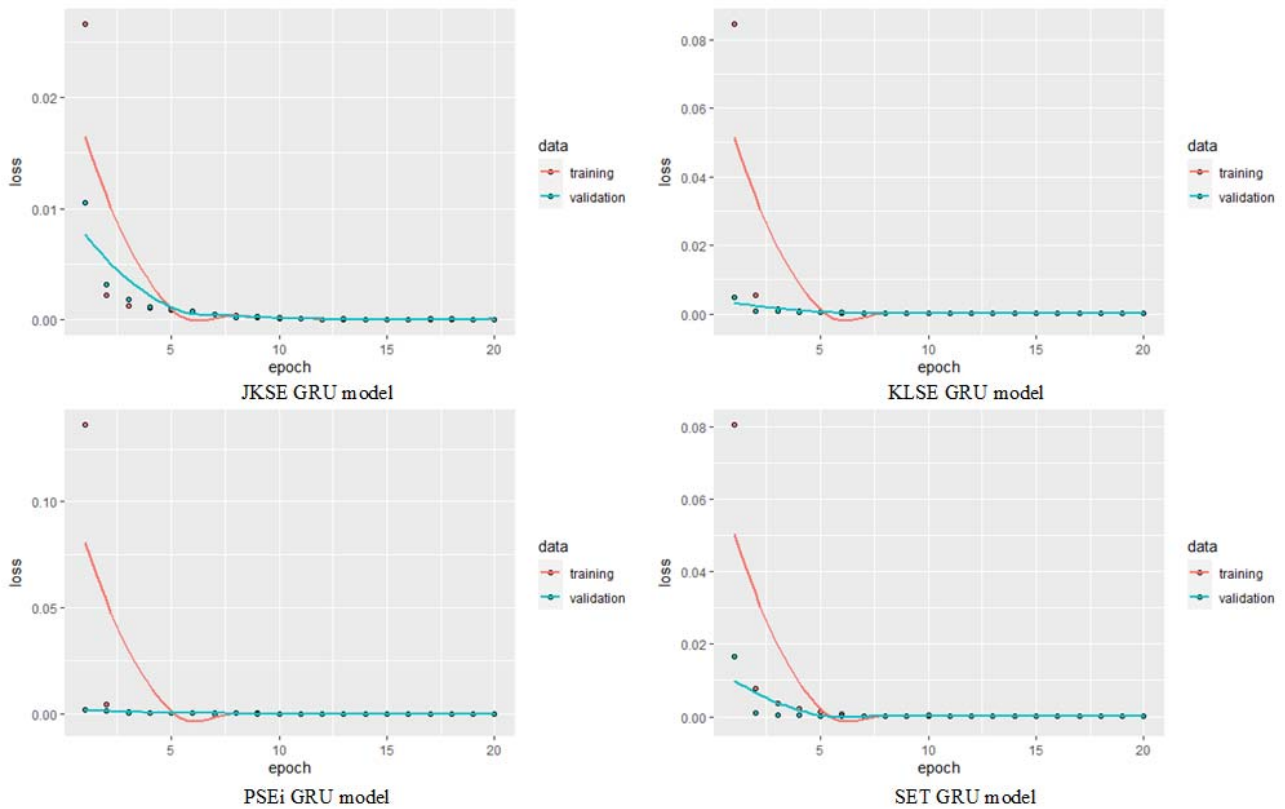


Figure 8. The learning curve of the selected model.

Figure 8 shows that all models can quickly learn the training data pattern with $epoch \leq 20$. Based on the observation and evaluation of the LC, it can be concluded that there are no under-fitting or over-fitting models on the training dataset. If under-fitting or over-fitting occurs, the hyperparameter is reset, until the model can capture the pattern of training data and validation data. Under-fitting occurs when the model is unable to get a low enough error value on the training set and validation set. Over-fitting occurs when the accuracy value on the training data is high while the accuracy value on validation is low. This usually occurs if the model is trained for too long so that the validation loss plot decreases to a certain point and starts to increase again. A best-fitting LC is the goal of the learning algorithm and falls between over-fitting and under-fitting models. Best-fitting is identified by the training and validation loss decreasing to a point of stability with minimal difference between the two final loss values. Based on the LC evaluation, the selected model showed indications that the training stage worked well. It is characterized by the training and validation data loss functions that decrease dramatically and achieve a good fit before the end of the epoch.

Based on the LC evaluation, the best-fitting model architecture was finally selected. This model is chosen to forecast test data. Figure 9 shows the forecasting results of the RNNs, LSTM, and GRUs models using the JKSE dataset.



Figure 9. Forecasting graphs of RNNs, LSTM, and GRUs models using JKSE data.

Figure 9 shows the forecasting graph using the JKSE data. All models can read the pattern of the training data. The GRUs model forecasting results are closer to the training data than the RNNs and LSTM models. Likewise, when the GRUs model forecasts the test data, it outperforms the RNNs and LSTM models. Figure 10 shows the output of the RNNs, LSTM, and GRUs models when using KLSE data.



Figure 10. Forecasting graphs of RNNs, LSTM, and GRUs models using KLSE data.

Figure 10 shows the output of the RNNs, LSTM, and GRUs models using the KLSE data. At the training stage, all models can capture the patterns of the training data. The RNNs model is superior to the LSTM on the KLSE data, while the GRUs model seems to remain the most optimal among the three. Figure 11 displays the forecasting results of the three models using PSEi data.



Figure 11. Forecasting graphs of RNNs, LSTM, and GRUs models using PSEi data.

Figure 11 shows the output of the RNNs, LSTM, and GRUs models using the PSEi data. All models learned the training data well, so there is no dominant model among the three. Likewise, when these three models forecast test data, they produce forecasts very close to the test data. Figure 12 displays the forecasting results of the three models using SET data.

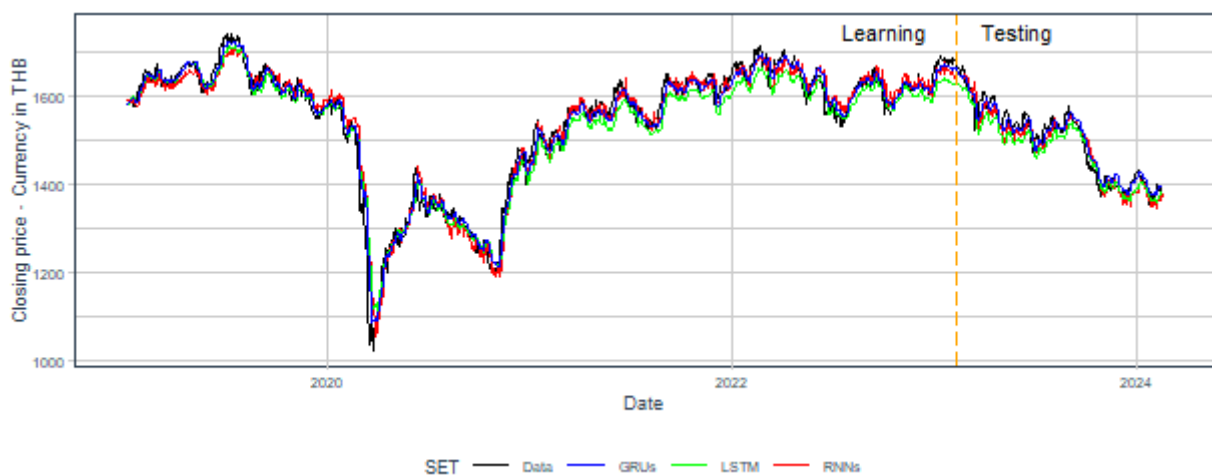


Figure 12. Forecasting graphs of RNNs, LSTM, and GRUs models using SET data.

Figure 12 visually shows the output of the RNNs, LSTM, and GRUs models using the SET data. In the training stage, the graph of the GRUs model looks almost the same as the data graph. This indicates that the GRUs model is dominant in this dataset. Likewise, at the testing stage, the GRUs model has better accuracy than the RNNs and LSTM models. Figures 9–12 have displayed the training and testing result graphs. All models are best-fitting models based on the LC diagnostics. The model is initially trained using the learning data. Then, the model is controlled based on the LC observations by changing the hyperparameter settings until it produces the best-fitting. The

best-fitting model is used to forecast the test data. The results of this test data forecasting is used as a new dataset for the calculation of X' . In the training phase (left vertical line graph), it can be seen that the model managed to read the data pattern well. All model outputs produce values that are close to the target variable values and follow the pattern of the learning data distribution. Likewise, when the selected model is used to forecast the test data (right-hand vertical line graph), the model successfully follows the target data pattern. Furthermore, model performance is measured by the MAPE, RMSE, and MSE methods. Table 4 presents the accuracy of the RNNs, LSTM, and GRUs models.

Table 4. Model accuracy.

Symbol	RNNs			LSTM			GRUs		
	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE
JKSE	0.0173	0.0087	0.0062	0.0163	0.0082	0.0061	0.0113	0.0056	0.00410
KLSE	0.0139	0.0134	0.0100	0.0226	0.0208	0.0162	0.0109	0.0110	0.00789
PSEi	0.0163	0.0188	0.0130	0.0175	0.0197	0.0137	0.0148	0.0165	0.01160
SET	0.0122	0.0162	0.0121	0.0176	0.0166	0.0129	0.0107	0.0110	0.00760

Table 4 shows the model accuracy values. Based on the MAPE, RMSE, and MAE values, the GRUs model whose architecture is simpler than LSTM has the best accuracy in all samples. All forecasting models produce highly accurate accuracy based on MAPE values. Furthermore, this GRUs model will be used to forecast with test data. The forecasting results will be min-max renormalized using Eq (3). Table 5 presents X_i , X'_i , and \bar{X}_i .

Table 5. Comparison X_i and \bar{X}_i .

Symbol		Period					
		1075	1076	1077	1078	...	1335
JKSE	X_i	0.0000072	-0.0001446	-0.0030911	-0.0092293	...	0.0044178
	X'_i	-0.0000118	-0.0001929	-0.0005698	-0.0015931	...	0.0006556
	\bar{X}_i	-0.0000046	-0.0003375	-0.0036608	-0.0108224	...	0.0050733
KLSE	X_i	-0.0049587	-0.0023292	0.0003733	-0.0067910	...	0.0033827
	X'_i	0.0019965	0.0000160	-0.0008551	-0.0013014	...	0.0016659
	\bar{X}_i	-0.0029622	-0.0023133	-0.0004818	-0.0080924	...	0.0050486
PSEi	X_i	-0.0054124	-0.0051482	0.0084281	-0.0149582	...	-0.0012961
	X'_i	-0.0006959	-0.0019877	-0.0025911	-0.0001923	...	0.0018835
	\bar{X}_i	-0.0061082	-0.0071360	0.0058369	-0.0151505	...	0.0005874
SET	X_i	-0.0039921	0.0036447	0.0065996	-0.0054836	...	-0.0007208
	X'_i	-0.0013941	-0.0011770	-0.0006119	0.0017013	...	0.0002137
	\bar{X}_i	-0.0053862	0.0024678	0.0059877	-0.0037823	...	-0.0005072

Table 5 presents the values of X_i , X'_i , and \bar{X}_i using test data. The model will produce output starting from period 1075 to period 1335. There are two models generated in this study. The first model is the $\text{VaR}_{\text{GPD}(\alpha)}$ model using X_i data. The second model is the $\text{VaR}_{\text{GPD-ML}(\alpha)}$ model where this model will use \bar{X}_i data.

4. Discussion

4.1. Return characteristics

Descriptive statistics analysis was carried out to add information and variable relationships from return data. Table 6 presents descriptive statistics of X and \bar{X} .

Table 6. Descriptive statistics of the return data.

Symbol		Min	Mean	Max	SD	Skewness	Kurtosis
JKSE	X	-0.0213853	0.0002525	0.0171358	0.005581595	-0.2852144	4.309061
	X'	-0.0071181	0.0001813	0.0056336	0.002099109	-0.2461606	3.103821
	\bar{X}	-0.0223225	0.0004338	0.0145267	0.005801407	-0.5799707	4.399383
KLSE	X	-0.0196929	0.0001350	0.0144654	0.004437276	0.0941133	4.678326
	X'	-0.0085410	0.0000763	0.0056590	0.001788984	-0.3954494	6.238534
	\bar{X}	-0.0247946	0.0002112	0.0129547	0.004827826	-0.3090902	5.754314
PSEi	X	-0.0230900	0.0000632	0.0247400	0.007899659	-0.1258554	3.113695
	X'	-0.0103700	-0.0000056	0.0111200	0.003295881	0.1769271	3.499875
	\bar{X}	-0.0282700	0.0000576	0.0254700	0.008398925	-0.2010673	3.316509
SET	X	-0.0312637	-0.0006605	0.0269770	0.007185647	0.0663270	4.906188
	X'	-0.0118895	-0.0006341	0.0055026	0.003116718	-0.2612103	2.834681
	\bar{X}	-0.0359890	-0.0012950	0.0181410	0.007862045	-0.4426459	4.105686

Table 6 presents the descriptive statistics of the return dataset. Comparing the SD values of X and \bar{X} , all SD values of $X' < X < \bar{X}$. The SD values in all samples are greater than the mean, identifying that there is a lot of variation in the data and there may be extreme values in the data distribution. The JKSE X , JKSE \bar{X} , KLSE \bar{X} , PSEi X , PSEi \bar{X} , and SET \bar{X} data yield negative skewness values, indicating that the distribution is skewed to the left, whereas the KLSE X and SET X data yield positive skewness values. The kurtosis values are greater than three, indicating that the data distribution has more values in the tail compared to the normal distribution, identifying a heavy tail in the data distribution.

4.2. Diagnostic plots

Table 6 shows skewness values that vary between positive and negative. In addition to the skewness values, the identification of extreme values is visualized using the normal QQ plot and histogram. In the QQ plot and histogram graphs, \bar{X} is symbolized by X-ML. Figure 13 displays the QQ plot X and X-ML of the JKSE, KLSE, PSEi, and SET data.

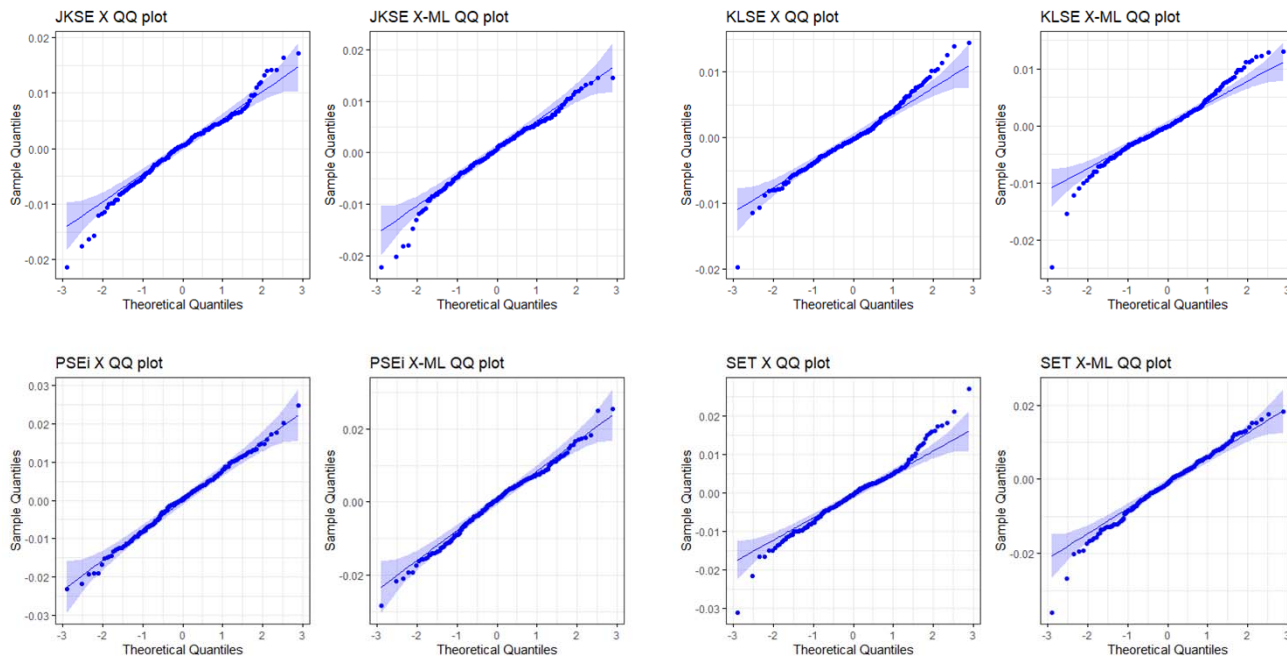


Figure 13. QQ plot of the JKSE, KLSE, PSEi, and SET data.

Figure 13 visually shows that most of the data distribution looks abnormal, except for the PSEi X and PSEi X-ML model data. The data of these two models are identified as normal data distribution where the plot shows a pattern close to the diagonal line. In contrast to the other data, there is a heavy tail in the data distribution. Figure 14 displays a histogram with the normal curve of X and \bar{X} values.

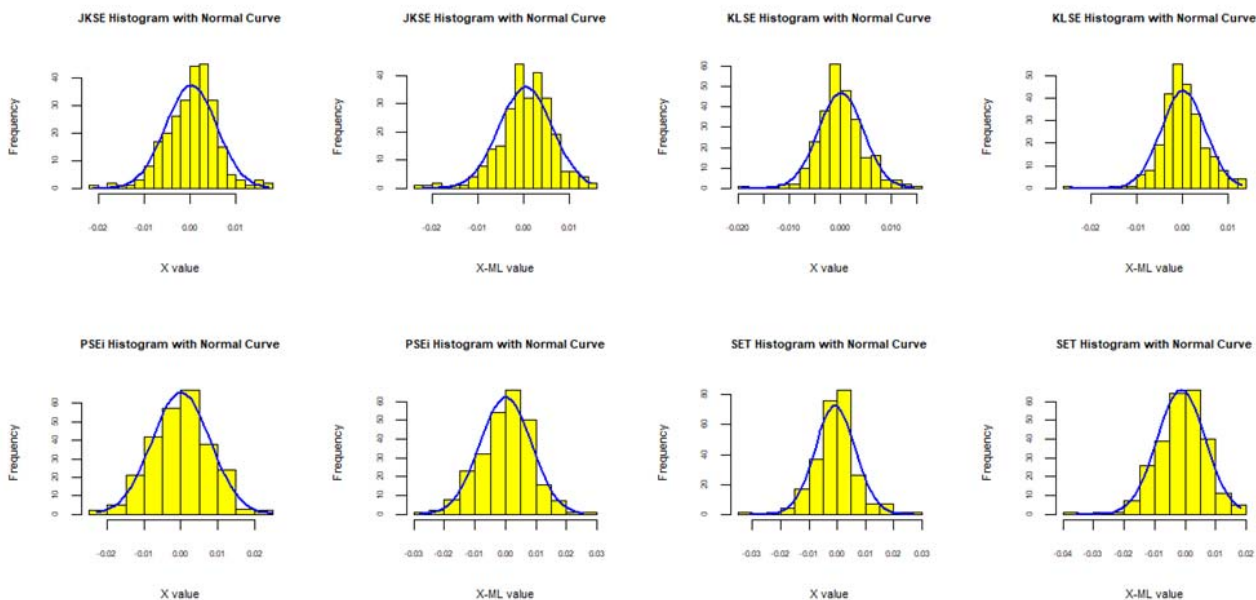


Figure 14. Histograms of the JKSE, KLSE, PSEi, and SET data.

Figure 14 shows histograms with normal curves of JKSE, KLSE, PSEi, and SET based on X and \bar{X} values. Based on the visual observation, there is a heavy tail in the distribution of X and \bar{X} , where it does not appear that the values are symmetrically distributed, with the distance between the right and left tails of the distribution being equal. Only the PSEi data for both X and \bar{X} appear to be symmetrically distributed, which identifies the data as normally distributed.

4.3. Threshold and parameter estimates

The POT method identifies extreme patterns of data behavior by determining the value of u . Data that exceeds u is an extreme value. The value of u is determined as optimally as possible to produce a minimum error rate, to have a small bias, and to have an ideal variance value when estimating the model. To obtain reliable estimates, the number of exceedances required is at least 50 to 150 to produce stable parameter estimates. This means that for the number of data $n \leq 1000$, the threshold value is approximated by.

Theorem 4. *The threshold based on normal distribution*

Let $X_1, X_2, X_3, \dots, X_n$ are iid random variables, with unknown distribution function F . The POT model approach identifies values exceeding u as extreme values. The value of u with minimal bias and a sufficient number of extreme samples for $n < 1000$ is well approximated:

$$u \approx \mu + \sqrt{\frac{1}{N} \sum_i^n (X_i - \mu)^2}, \quad (47)$$

where u is the threshold, n is the number of data, $\{X_1, X_2, X_3, \dots, X_n\}$ are the data values, and μ is the mean of the data values.

Proof. Let X_1, X_2, \dots, X_n be iid random variables with an unknown CDF $F(x)$. In the POT model approach, The threshold should be chosen such that it identifies the extreme value with the ideal amount and minimum bias. Assuming a normal distribution, let the data X follow a normal distribution, i.e., $X \sim N(\mu, \sigma^2)$, with a mean parameter μ and standard deviation σ within the range $(\mu - \sigma, \mu + \sigma)$. To analyze extreme values, the threshold is set as $u = \mu + \sigma$. Normalize the data using the Z -score, defined as:

$$Z = \frac{X - \mu}{\sigma}, \quad (48)$$

since $X \sim N(\mu, \sigma^2)$, Z follows a standard normal distribution, i.e., $Z \sim N(0,1)$. The threshold $u = \mu + \sigma$ can be rewritten as:

$$Z = \frac{(\mu + \sigma) - \mu}{\sigma} = 1. \quad (49)$$

Therefore, the probability that X exceeds the threshold $u = \mu + \sigma$ is the same as the probability that $Z \geq 1$:

$$P(X \geq \mu + \sigma) = P(Z \geq 1), \quad (50)$$

using the CDF of the standard normal distribution:

$$P(Z \geq 1) = 1 - P(Z \leq 1) = 1 - \Phi(1), \quad (51)$$

where $\Phi(1)$ is the CDF of the standard normal distribution at $Z = 1$. From the standard normal distribution table:

$$\Phi(1) = 0.8413, \quad (52)$$

thus

$$P(Z \geq 1) = 1 - \Phi(1) = 0.1587. \quad (53)$$

This means that the probability of a value X being above the threshold $= \mu + \sigma = 0.1587$, or approximately 15.87%. Therefore, it is proven that the threshold $\mu + \sigma$ minimizes bias with a sufficient number of samples, which is about 15.87% above the threshold.

4.4. Threshold experiment

Experiments were conducted to obtain the value of u with minimal bias and a sufficient number of extreme samples. In this experiment, the setting of u is made with four models. The u_1 model states that the extreme value is the value above the mean value, therefore $u_1 = \mu$. The u_2 model is the value of u based on the observation of the MRLP. The u_3 model is the extreme value is the value above the mean + SD; therefore, $u_3 = \mu + \sigma$. The u_4 model states that the extreme value is the value above mean + twice the SD; thus, $u_4 = \mu + 2\sigma$, where u is the threshold, μ is the mean, and σ is the standard deviation.

Estimating the value of u using the MRLP method by looking at the value of u that is close to linear. Figure 15 displays the MRLP of the X and \bar{X} data distributions.

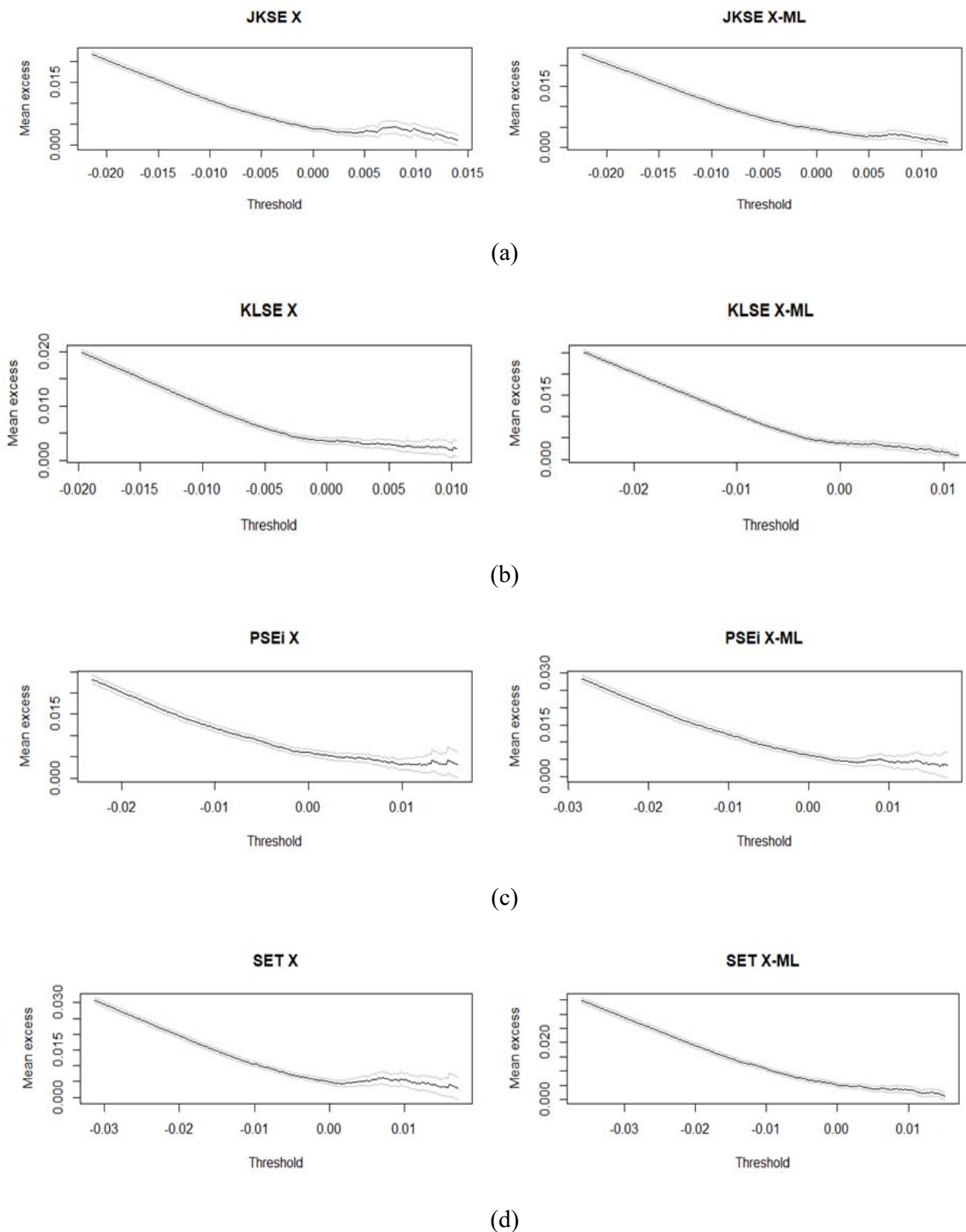


Figure 15. Mean residual life plot: (a) JKSE; (b) KLSE; (c) PSEi; and (d) SET.

Table 7 presents the results of the experiments conducted to obtain the ideal u value, as well as the estimated δ and shape ξ .

Table 7. The parameter estimation.

Symbol	u model	n	u	Nu	Percentage	δ	ξ	Status
JKSE X	u_1	261	0.00025249	141	54.02%	0.00449056	-0.15128876	Too low
	u_2	261	0.005	41	15.71%	0.00277539	0.12457724	Ideal
	u_3	261	0.00583408	30	11.49%	0.00349917	-0.03960151	Ideal
	u_4	261	0.01141568	8	3.07%	0.00595227	-1.04059067	Too high
JKSE \bar{X}	u_1	261	0.00043378	136	52.11%	0.00569456	-0.33742725	Too low
	u_2	261	0.005	52	19.92%	0.00310365	-0.10562157	Ideal
	u_3	261	0.00623519	31	11.88%	0.00472750	-0.48832701	Ideal
	u_4	261	0.01203660	5	1.92%	0.00250317	-1.00525948	Too high
KLSE X	u_1	261	0.00013498	121	46.36%	0.00421928	-0.17932436	Too low
	u_2	261	0.004	42	16.09%	0.00416035	-0.29147247	Ideal
	u_3	261	0.00457226	38	14.56%	0.00365306	-0.23417870	Ideal
	u_4	261	0.00900953	8	3.07%	0.00668316	-1.22495709	Too high
KLSE \bar{X}	u_1	261	0.00021124	121	46.36%	0.00496352	-0.28462496	Too low
	u_2	261	0.004	48	18.39%	0.00538449	-0.55239240	Ideal
	u_3	261	0.00503906	39	14.94%	0.00484277	-0.55778751	Ideal
	u_4	261	0.00986689	8	3.07%	0.00441038	-1.42832318	Too high
PSEi X	u_1	261	0.00006319	133	50.96%	0.00766454	-0.26180449	Too low
	u_2	261	0.007	49	18.77%	0.00476065	-0.14938315	Ideal
	u_3	261	0.00796285	41	15.71%	0.00445067	-0.13242676	Ideal
	u_4	261	0.01586251	5	1.92%	0.00411562	-0.22284767	Too high
PSEi \bar{X}	u_1	261	0.00005764	141	54.02%	0.00743901	-0.22313744	Too low
	u_2	261	0.008	35	13.41%	0.00531591	-0.12794773	Ideal
	u_3	261	0.00845656	31	11.88%	0.00566219	-0.17101014	Ideal
	u_4	261	0.01685549	6	2.30%	0.00146834	0.73315525	Too high
SET X	u_1	261	-0.00066051	131	50.19%	0.00732443	-0.22988621	Too low
	u_2	261	0.007	25	9.58%	0.00762504	-0.27965514	Ideal
	u_3	261	0.00652514	27	10.34%	0.00754547	-0.26388436	Ideal
	u_4	261	0.01371078	10	3.83%	0.00450188	-0.12583437	Too high
SET \bar{X}	u_1	261	-0.00129463	134	51.34%	0.00902711	-0.43933976	Too low
	u_2	261	0.007	35	13.41%	0.00498006	-0.33897700	Ideal
	u_3	261	0.00656741	36	13.79%	0.00573323	-0.41704069	Ideal
	u_4	261	0.01442946	5	1.92%	0.00724593	-1.95245071	Too high

Table 7 shows that the u_1 model produces too many extreme values, the average number of extreme values identified is 50.67% of the total data, this happens because the value of u is too low. This value will potentially produce bias. The u_2 model determines the value of u based on observations from the MRLP. Visually, the u value obtained from MRLP observations is a value close to SD with two decimal places. This model produces an average number of extrinsic values of 15.66% of the data. The u_3 model determines the value of u based on the mean + σ value. This

model identifies extreme values, including 13.07% on average, of the data. The u_3 model produces u with better precision, values with eight decimal places compared to the u_2 model. The u_4 model produced an average number of extreme values of 2.63% of the data. This is a very small gain when estimating the model. Based on the experiments conducted, two models produce the ideal number of extreme values, namely the u_2 model and the u_3 model. The u_3 model is chosen for determining the value of the u because it produces a larger value of the u , where the higher the u value, the more extreme data follows the GPD. Extreme values that exceed the threshold based on the u_3 model will be checked to determine whether the data exceedances follow the GPD using the Kolmogorov-Smirnov test. The hypothesis criteria are:

H_0 : Data that exceeds the threshold is GPD.

H_1 : Data that exceeds the threshold is not GPD.

Table 8 shows the Kolmogorov-Smirnov test for data exceeding the threshold based on the u_3 model.

Table 8. Kolmogorov-Smirnov test of exceedances data based on u_3 model.

Symbol	u_3	Nu	p-value	Hypothesis
JKSE X	0.005834084	30	0.3672	H_0 is accepted
JKSE \bar{x}	0.006235188	31	0.3593	H_0 is accepted
KLSE X	0.004572256	38	0.9986	H_0 is accepted
KLSE \bar{x}	0.005039064	39	0.6716	H_0 is accepted
PSEi X	0.007962850	41	0.7741	H_0 is accepted
PSEi \bar{x}	0.008456561	31	0.9778	H_0 is accepted
SET X	0.006525136	27	0.9980	H_0 is accepted
SET \bar{x}	0.006567412	36	0.7747	H_0 is accepted

Based on Table 8, the Kolmogorov-Smirnov test obtained p-values > 0.05 for extreme data exceeding the threshold using the u_3 model. This result indicates that there is not enough evidence to reject hypothesis H_0 , therefore the data that exceeds the threshold is GPD.

4.5. VaR forecasting

Table 7 shows the parameter estimates. The parameter estimate δ describes the diversity of extreme values. The parameter ξ describes the tail behavior of the extreme data, and the greater the value of ξ , the greater the chance of extreme occurrence, and $\xi > 0$ extremes are not limited, and $\xi < 0$ extremes are limited distribution. $VaR_{GPD-ML(\alpha)}$ is obtained based on Eq (42), using the value of u from the u_3 model, the estimated value of parameter δ , and parameter ξ . The calculation results of the $VaR_{GPD(\alpha)}$ model and $VaR_{GPD-ML(\alpha)}$ model are presented in Table 9.

Table 9. Approximate parameters and VaR forecasting.

Symbol	u_3	Parameters		Model	VaR		
		δ	ξ		0.90	0.95	0.99
JKSE	0.005834084	0.003499169	-0.039601513	VaR _{GPD(α)}	0.006320044	0.008699339	0.013978394
	0.006235188	0.004727499	-0.488327005	VaR _{GPD-ML(α)}	0.007015331	0.009571204	0.013024818
KLSE	0.004572256	0.003653063	-0.234178703	VaR _{GPD(α)}	0.005885905	0.008026365	0.011840161
	0.005039064	0.004842766	-0.557787506	VaR _{GPD-ML(α)}	0.006781583	0.009006807	0.011800082
PSEi	0.007962850	0.004450674	-0.132426763	VaR _{GPD(α)}	0.009914008	0.012690503	0.018234225
	0.008456561	0.005662193	-0.171010144	VaR _{GPD-ML(α)}	0.009416560	0.013010300	0.019881080
SET	0.006525136	0.007545473	-0.263884364	VaR _{GPD(α)}	0.006779798	0.011516907	0.019684119
	0.006567412	0.005733231	-0.417040690	VaR _{GPD-ML(α)}	0.008292838	0.011310846	0.015712943

Table 9 shows that the confidence level greatly affects the estimated value of VaR. The greater the confidence level, the greater the value of VaR. This describes the greater the risk that will be faced by investors. This level of risk describes the profits and losses experienced by investors. Table 10 presents the output of the investment risk forecasting model using the VaR_{HS(α)}, VaR_{GPD(α)}, and VaR_{GPD-ML(α)} models.

Table 10. VaR estimates.

Symbol	Model	Confidence Level (α)		
		0.90	0.95	0.99
JKSE	VaR _{HS(α)}	-0.006687400	-0.009120857	-0.015895400
	VaR _{GPD(α)}	0.006320044	0.008699339	0.013978394
	VaR _{GPD-ML(α)}	0.007015331	0.009571204	0.013024818
KLSE	VaR _{HS(α)}	-0.004881989	-0.006678174	-0.009514125
	VaR _{GPD(α)}	0.005885905	0.008026365	0.011840161
	VaR _{GPD-ML(α)}	0.006781583	0.009006807	0.011800082
PSEi	VaR _{HS(α)}	-0.010520770	-0.012701800	-0.019099420
	VaR _{GPD(α)}	0.009914008	0.012690503	0.018234225
	VaR _{GPD-ML(α)}	0.009416560	0.013010300	0.019881080
SET	VaR _{HS(α)}	-0.009766997	-0.011342440	-0.016628710
	VaR _{GPD(α)}	0.006779798	0.011516907	0.019684119
	VaR _{GPD-ML(α)}	0.008292838	0.011310846	0.015712943

Table 10 shows the output of the investment risk forecasting model using normal distribution VaR, which produces a negative VaR value. A negative VaR value indicates a possible loss in the value of the investment at the level of confidence used. Historical VaR values produce negative values because the approach of this method focuses on the left tail of the distribution of the return data. The left tail of the distribution reflects the worst-case loss scenarios or extreme values on the downside. Therefore, the VaR value is usually negative because it indicates a potential loss. If using the normal distribution VaR method, at a confidence level of 90%, the maximum loss estimate value occurs in the country of the Philippines. The VaR_{HS(0.90)} value for PSEi was -0.010520770. This means that in market conditions that are assumed to be normal with moderate volatility and no

extreme events, the maximum loss is estimated to be around 1.05% of the value of the investment made. At a 95% confidence level, the largest estimated loss occurred in the Philippines country with an estimated value of $\text{VaR}_{\text{HS}(0.95)}$ of -0.012701800. Likewise, at the 99% confidence level, the highest value is the estimated investment risk in the Philippines with $\text{VaR}_{\text{HS}(0.99)}$ being -0.012701800. The results of the normal distribution VaR estimate do not take into account extreme events that tend to occur in financial markets. This makes normal VaR results sometimes less reflective of true risk, especially when there is high volatility and stock fluctuations that are outside the normal distribution.

In contrast to $\text{VaR}_{\text{HS}(\alpha)}$, the VaR model with the EVT approach focuses on measuring risk for rare events that have a major impact on investment activities by modeling the tail of the distribution from the return data. Table 10 shows the results of the $\text{VaR}_{\text{GPD}(\alpha)}$ method with the EVT approach and the $\text{VaR}_{\text{GPD-ML}(\alpha)}$ model with the ML combination EVT approach. At the 90% confidence level, generally, the estimation of the $\text{VaR}_{\text{GPD-ML}(0.90)}$ model produces a larger estimate than the $\text{VaR}_{\text{GPD}(0.90)}$ model, except for the PSEi model. At the 95% confidence level, generally, the $\text{VaR}_{\text{GPD-ML}(0.95)}$ model also produces a greater estimate of investment risk than the $\text{VaR}_{\text{GPD}(0.95)}$ model, except for SET. In contrast, at the 99% confidence level, the $\text{VaR}_{\text{GPD}(0.99)}$ model generally produces higher risk estimates, except for the PSEi model. It can be concluded that at 90% and 95% confidence levels, the $\text{VaR}_{\text{GPD-ML}(\alpha)}$ model is more sensitive to rare but high-impact events, such as market crises or high volatility, so this model provides more protection to investors in the event of more extreme conditions than the $\text{VaR}_{\text{GPD}(\alpha)}$ model.

At the 90% confidence level, the highest level of investment risk among these four countries occurs in the Philippines. The estimated value of $\text{VaR} = 0.009914008$ is obtained from the $\text{VaR}_{\text{GPD}(0.90)}$ model. This VaR value estimates that if an investor invests in the Philippines, it is likely that the investor will experience a maximum profit of 0.99% of the invested value.

At the 95% confidence level, the highest level of investment risk occurs in the Philippines, with the $\text{VaR}_{\text{GPD-ML}(0.95)}$ model, the estimated $\text{VaR} = 0.0130103$ is obtained. This VaR value illustrates that if an investor invests in the Philippines, at a confidence level of 95%, it is likely to experience a maximum profit of 1.30% of its investment value.

At the 99% confidence level, the highest estimated investment risk level among these four countries is the Philippines. The estimated value of $\text{VaR} = 0.019881080$ is obtained from the $\text{VaR}_{\text{GPD-ML}(0.99)}$ model, meaning that if investors invest in the Philippines, at a 99% confidence level will experience a maximum profit of 1.99% of the investment value.

The VaR method of the EVT approach, focusing only on the tail part of the distribution, is considered better at representing extreme events. This method specifically estimates the risk by paying attention to the behavior of the data at the tail of the distribution. Therefore, this approach is more sensitive to rare events that have a major impact, such as market crises. The advantage of this method is that it can capture the risk of extreme events that are not reflected in the normal distribution and is more accurate in markets with high volatility or when there is a risk of extreme price spikes. The $\text{VaR}_{\text{GPD-ML}(0.95)}$ model tends to get a larger VaR value, as it estimates the potential loss from extreme events. The positive values produced by the $\text{VaR}_{\text{GPD}(\alpha)}$ and $\text{VaR}_{\text{GPD-ML}(\alpha)}$ models show that under extreme conditions, the estimated investment risk can exceed a higher limit than the normal distribution estimate. Therefore, when the data shows high volatility as shown in Figure 6, using the $\text{VaR}_{\text{GPD-ML}(\alpha)}$ model is more recommended, as it better reflects the investment potential and risk of extreme events.

4.6. Backtesting

Backtesting is conducted to validate the model using the Kupiec method. The estimated VaR value is compared with the actual return data, which in this study is the test data. If the actual return $>$ VaR is found, it is considered a failure. The LR value is calculated by Eq (43), and then the LR value is compared with the chi-square degrees of freedom one for each VaR confidence level based on the chi-square table with a degree of freedom of one $\alpha = 90\%$ is 2.70554, $\alpha = 95\%$ is 3.84146, and $\alpha = 99\%$ is 6.6349. The model with $VaR_{(\alpha)}$ at a 90% confidence level is valid, if the LR value $<$ 2.70554, and vice versa the $VaR_{(\alpha)}$ model at a 90% confidence level is invalid if $LR >$ 2.70554. The $VaR_{(\alpha)}$ model at the 95% confidence level is valid, if the LR value is $<$ 3.84146, and otherwise the $VaR_{(\alpha)}$ model is invalid if $LR >$ 3.84146. The same calculation applies to the model with $VaR_{(\alpha)}$ at the 99% confidence level declared valid if $LR <$ 6.6349 otherwise the model is declared invalid if $LR >$ 6.6349. Table 11 shows the backtesting results.

Table 11. The backtesting results.

Symbol	Model	α	VaR	f	LR	Status
JKSE	$VaR_{GPD(\alpha)}$	0.90	0.006320044	28	0.1505	Valid
		0.95	0.008699339	13	0.0002	Valid
		0.99	0.013978394	5	1.7431	Valid
	$VaR_{GPD-ML(\alpha)}$	0.90	0.007015331	18	3.9715	Invalid
		0.95	0.009571204	12	0.0913	Valid
		0.99	0.013024818	6	3.2536	Valid
KLSE	$VaR_{GPD(\alpha)}$	0.90	0.005885905	27	0.0341	Valid
		0.95	0.008026365	11	0.3573	Valid
		0.99	0.011840161	3	0.0562	Valid
	$VaR_{GPD-ML(\alpha)}$	0.90	0.006781583	20	1.7089	Valid
		0.95	0.009006807	8	2.3726	Valid
		0.99	0.011800082	3	0.0562	Valid
PSEi	$VaR_{GPD(\alpha)}$	0.90	0.009914008	29	0.3467	Valid
		0.95	0.012690503	12	0.0913	Valid
		0.99	0.018234225	2	0.1566	Valid
	$VaR_{GPD-ML(\alpha)}$	0.90	0.009416560	32	1.3927	Valid
		0.95	0.013010300	11	0.3573	Valid
		0.99	0.019881080	2	0.1566	Valid
SET	$VaR_{GPD(\alpha)}$	0.90	0.006779798	30	0.0522	Valid
		0.95	0.011516907	14	0.0712	Valid
		0.99	0.019684119	2	0.1566	Valid
	$VaR_{GPD-ML(\alpha)}$	0.90	0.008292838	22	0.7519	Valid
		0.95	0.011310846	14	0.0712	Valid
		0.99	0.015712943	7	5.1069	Valid

Backtesting with the Kupiec method is based on the failure ratio f with a log-likelihood ratio approach. At a 90% confidence level, $n = 261$, this method produces a valid model if $19 \leq f \leq 34$. At the 95% confidence level, the model is valid if $7 \leq f \leq 20$. At a 99% confidence level, this method

produces a valid model if $f \leq 7$, and beyond these conditions, the model is declared invalid.

Table 11 shows the backtesting results, which state that the $\text{VaR}_{\text{GPD-ML}(\alpha)}$ model is generally valid in all samples, except at the 90% confidence level the $\text{VaR}_{\text{GPD-ML}(0.90)}$ model produces too few f and causes this model to be invalid. At the 95% and 99% confidence levels, the $\text{VaR}_{\text{GPD-ML}(\alpha)}$ model is valid in all data samples. At the 95% confidence level, this model generally produces a smaller f value when compared to the $\text{VaR}_{\text{GPD}(0.95)}$ model. In general, the $\text{VaR}_{\text{GPD-ML}(0.95)}$ model is valid in all samples at the 95% confidence level, and generally, the 95% confidence level is used as a reference in most studies.

5. Conclusions

In the ML-based multivariate time series forecasting model, applying multivariate SW techniques will make it easier for the model to read the training data pattern so that the model is highly accurate based on the acquisition of MAPE in all samples in this study. Reviewing the LC of the model during training can be used to diagnose problems in learning, such as under-fitting models or over-fitting models so that the model can be controlled to obtain the best-fitting model.

In this research, the results of historical forecasting of composite stock index closing price data using the RNNs, LSTM, and GRUs methods are used to find ML-based returns. The GRUs model significantly outperforms the RNNs and the LSTM models in all data samples based on the accuracy values obtained.

The investment risk forecasting model uses the EVT approach, identifying extreme data behavior patterns by determining the u value. In this study, the value of u is determined based on the value of the threshold based on the normal distribution, so that the value of u with the ideal number of extreme values can be obtained accurately and with precision. The threshold based on the normal distribution produces the u value with better accuracy, i.e., u values with five or more digits behind the comma while producing optimal u values for identifying the ideal number of extreme values, and the extreme value distribution identified is GPD.

Based on the backtesting results, the $\text{VaR}_{\text{GPD-ML}(\alpha)}$ model produces a valid model in all samples at 95% and 99% confidence levels. The proposed model generally results in a greater estimate of the value of investment risk than the $\text{VaR}_{\text{GPD}(\alpha)}$ model approach at a 95% confidence level. It can be concluded that at 90% and 95% confidence levels, the $\text{VaR}_{\text{GPD-ML}(\alpha)}$ model is more accurate at capturing the risk of extreme events that are not reflected in the normal distribution when the market is non-linear with high volatility or when there is a risk of extreme price spikes; thus, this model provides more protection to investors in the event of more extreme conditions than the $\text{VaR}_{\text{GPD}(\alpha)}$ model.

Author contributions

M. Melina: Conceptualization (equal), methodology (equal), writing-original draft (lead), software (lead); Sukono: Supervision (lead), formal analysis (lead), methodology (lead), validation (lead); H. Napitupulu: Data curation (lead), conceptualization (supporting), writing-review and editing (equal); N. Mohamed: Investigation (lead), formal analysis (equal), formal analysis (equal). All authors have read and approved the final version of the manuscript for publication.

Acknowledgments

The authors of this article would like to thank Universitas Padjadjaran for handling the article processing charge for the publication of this article.

This study has been funded by Universitas Padjadjaran through the Academic Leadership Grant (ALG) scheme, grant number: 1549/UN6.3.1/PT.00/2023.

Conflict of interest

The authors declare no conflict of interest.

References

1. Y. Cao, Extreme risk spillovers across financial markets under different crises, *Econ. Model.*, **116** (2022), 106026. <https://doi.org/10.1016/j.econmod.2022.106026>
2. H. Y. Liu, A. Manzoor, C. Wang, L. Zhang, Z. Manzoor, The COVID-19 outbreak and affected countries stock markets response, *Int. J. Environ. Res. Public Health.*, **17** (2020), 2800. <https://doi.org/10.3390/ijerph17082800>
3. D. Chaerani, H. Napitupulu, A. Z. Irmansyah, A systematic literature review on optimization modeling to electricity strategy business during Covid-19 pandemic, *Eng. Lett.*, **31** (2023), 1–20.
4. S. S. Jeris, R. D. Nath, Covid-19, oil price and UK economic policy uncertainty: Evidence from the ARDL approach, *Quant. Financ. Econ.*, **4** (2020), 503–514. <https://doi.org/10.3934/QFE.2020023>
5. Y. Fang, Z. Shao, The Russia-Ukraine conflict and volatility risk of commodity markets, *Financ. Res. Lett.*, **50** (2022), 103264. <https://doi.org/10.1016/j.frl.2022.103264>
6. K. Yang, Y. Wei, S. Li, J. M. He, Geopolitical risk and renewable energy stock markets: An insight from multiscale dynamic risk spillover, *J. Clean. Prod.*, **279** (2021), 123429. <https://doi.org/10.1016/j.jclepro.2020.123429>
7. F. He, B. Lucey, Z. Wang, Trade policy uncertainty and its impact on the stock market-evidence from China-US trade conflict, *Financ. Res. Lett.*, **40** (2021), 101753. <https://doi.org/10.1016/j.frl.2020.101753>
8. Y. Shi, L. Wang, J. Ke, Does the US-China trade war affect co-movements between US and Chinese stock markets? *Res. Int. Bus. Financ.*, **58** (2021), 101477. <https://doi.org/10.1016/j.ribaf.2021.101477>
9. J. P. Morgan, *Risk metrics technical document*, New York: RiskMetrics, 1996.
10. M. R. Nieto, E. Ruiz, Frontiers in VaR forecasting and backtesting, *Int. J. Forecast.*, **32** (2016), 475–501. <https://doi.org/10.1016/j.ijforecast.2015.08.003>
11. X. Yu, Z. Zhao, X. Zhang, Q. Y. Zhang, Y. L. Liu, C. Sun, Deep-learning-based open set fault diagnosis by extreme value theory, *IEEE Trans. Ind. Informatics.*, **18** (2022), 185–196. <https://doi.org/10.1109/TII.2021.3070324>
12. G. Petneházi, Quantile convolutional neural networks for value at risk forecasting, *Mach. Learn. Appl.*, **6** (2021), 100096. <https://doi.org/10.1016/j.mlwa.2021.100096>
13. A. F. Rossignolo, M. D. Fethi, M. Shaban, Market crises and Basel capital requirements: Could Basel III have been different? Evidence from Portugal, Ireland, Greece and Spain (PIGS), *J. Bank. Financ.*, **37** (2013), 1323–1339. <https://doi.org/10.1016/j.jbankfin.2012.08.021>

14. R. Gençay, F. Selçuk, Extreme value theory and value-at-risk: Relative performance in emerging markets, *Int. J. Forecast.*, **20** (2004), 287–303. <https://doi.org/10.1016/j.ijforecast.2003.09.005>
15. A. J. McNeil, D. Mathematik, Extreme value theory for risk managers a general introduction to extreme risk, *Intern. Model. CAD II.*, **3** (1999), 1–22.
16. S. Hubbert, *Essential mathematics for market risk management*, West Sussex, United Kingdom: John Wiley & Sons, Ltd, 2012. <https://doi.org/10.1002/9781118467213>
17. X. Ma, D. Zheng, G. Ding, J. M. Wang, “Extreme utilization” theory and practice in gas storages with complex geological conditions, *Pet. Explor. Dev.*, **50** (2023), 419–432. [https://doi.org/10.1016/S1876-3804\(23\)60397-0](https://doi.org/10.1016/S1876-3804(23)60397-0)
18. G. Evin, P. D. Sielenou, N. Eckert, P. Naveau, P. Hagenmuller, S. Morin, Extreme avalanche cycles: Return levels and probability distributions depending on snow and meteorological conditions, *Weather Clim. Extrem.*, **33** (2021), 100344. <https://doi.org/10.1016/j.wace.2021.100344>
19. F. M. Longin, From value at risk to stress testing: The extreme value approach, *J. Bank. Financ.*, **24** (2000), 1097–1130. [https://doi.org/10.1016/S0378-4266\(99\)00077-1](https://doi.org/10.1016/S0378-4266(99)00077-1)
20. P. Embrechts, M. V. Wüthrich, Recent challenges in actuarial science, *Annu. Rev. Stat. Appl.*, **9** (2022), 119–140. <https://doi.org/10.1146/annurev-statistics-040120-030244>
21. A. J. McNeil, *Calculating quantile risk measures for financial return series using extreme value theory*, Zürich, 1998. <https://doi.org/10.3929/ethz-a-004320029>
22. X. Ji, J. Wang, Z. H. Yan, A stock price prediction method based on deep learning technology, *Int. J. Crowd Sci.*, **5** (2021), 55–72. <https://doi.org/10.1108/IJCS-05-2020-0012>
23. T. G. Bali, S. N. Neftci, Disturbing extremal behavior of spot rate dynamics, *J. Empir. Financ.*, **10** (2003), 455–477. [https://doi.org/10.1016/S0927-5398\(02\)00070-1](https://doi.org/10.1016/S0927-5398(02)00070-1)
24. A. J. McNeil, R. Frey, Estimation of tail-related risk measures for heteroscedastic financial time series: An extreme value approach, *J. Empir. Financ.*, **7** (2000), 271–300. [https://doi.org/10.1016/S0927-5398\(00\)00012-8](https://doi.org/10.1016/S0927-5398(00)00012-8)
25. A. K. Singh, D. E. Allen, P. J. Robert, Extreme market risk and extreme value theory, *Math. Comput. Simul.*, **94** (2013), 310–328. <https://doi.org/10.1016/j.matcom.2012.05.010>
26. M. Karmakar, S. Paul, Intraday portfolio risk management using VaR and CVaR: A CGARCH-EVT-Copula approach, *Int. J. Forecast.*, **35** (2019), 699–709. <https://doi.org/10.1016/j.ijforecast.2018.01.010>
27. T. Ren, S. Li, S. Zhang, Stock market extreme risk prediction based on machine learning: Evidence from the American market, *North Am. J. Econ. Financ.*, **74** (2024), 102241. <https://doi.org/10.1016/j.najef.2024.102241>
28. S. Karim, M. Shafiullah, M. A. Naeem, When one domino falls, others follow: A machine learning analysis of extreme risk spillovers in developed stock markets, *Int. Rev. Financ. Anal.*, **93** (2024), 103202. <https://doi.org/10.1016/j.irfa.2024.103202>
29. H. M. Blom, P. E. Lange, M. Rissstad, Estimating value-at-risk in the EURUSD currency cross from implied volatilities using machine learning methods and quantile regression, *J. Risk Financ. Manag.*, **16** (2023), 1–23. <https://doi.org/10.3390/jrfm16070312>
30. M. Li, Financial investment risk prediction under the application of information interaction firefly algorithm combined with graph convolutional network, *PLoS One*, **18** (2023), 1–18. <https://doi.org/10.1371/journal.pone.0291510>

31. C. Hamzaçebi, D. Akay, F. Kutay, Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting, *Expert Syst. Appl.*, **36** (2009), 3839–3844. <https://doi.org/10.1016/j.eswa.2008.02.042>
32. K. E. Arun Kumar, D. V. Kalaga, C. M. S. Kumar, M. Kawaji, T. M. Brenza, Forecasting of COVID-19 using deep layer recurrent neural networks (RNNs) with gated recurrent units (GRUs) and long short-term memory (LSTM) cells, *Chaos Soliton. Fract.*, **146** (2021), 110861. <https://doi.org/10.1016/j.chaos.2021.110861>
33. S. Ahmed, R. K. Chakraborty, D. L. Essam, W. P. Ding, Poly-linear regression with augmented long short term memory neural network: Predicting time series data, *Inf. Sci. (Ny)*, **606** (2022), 573–600. <https://doi.org/10.1016/j.ins.2022.05.078>
34. F. Ricchiuti, G. Sperli, An advisor neural network framework using LSTM-based informative stock analysis, *Expert Syst. Appl.*, **259** (2025), 125299. <https://doi.org/10.1016/j.eswa.2024.125299>
35. J. Wang, X. Y. Wang, X. Wang, International oil shocks and the volatility forecasting of Chinese stock market based on machine learning combination models, *North Am. J. Econ. Financ.*, **2023**, 102065. <https://doi.org/10.1016/j.najef.2023.102065>
36. G. Kotsopolis, K. N. Konstantakis, D. L. Stamos, P. G. Michaelides, Oil prices and developing stock markets: Evidence from East Asia, *Dev. Sustain. Econ. Financ.*, **1** (2024), 100006. <https://doi.org/10.1016/j.dsef.2024.100006>
37. Z. Liu, J. Hu, S. Zhang, Z. P. He, Risk spillovers among oil, gold, stock, and foreign exchange markets: Evidence from G20 economies, *North Am. J. Econ. Financ.*, **74** (2024), 102249. <https://doi.org/10.1016/j.najef.2024.102249>
38. L. Chen, F. Wen, W. Li, H. Yin, L. L. Zhao, Extreme risk spillover of the oil, exchange rate to Chinese stock market: Evidence from implied volatility indexes, *Energy Econ.*, **107** (2022), 105857. <https://doi.org/10.1016/j.eneco.2022.105857>
39. K. Gokmenoglu, B. M. Eren, S. Hesami, Exchange rates and stock markets in emerging economies: New evidence using the Quantile-on-Quantile approach, *Quant. Financ. Econ.*, **5** (2021), 94–110. <https://doi.org/10.3934/QFE.2021005>
40. M. Zolfaghari, B. Sahabi, Impact of foreign exchange rate on oil companies risk in stock market: A Markov-switching approach, *J. Comput. Appl. Math.*, **317** (2017), 274–289. <https://doi.org/10.1016/j.cam.2016.10.012>
41. Z. Lyu, A. Ororbia, T. Desell, Online evolutionary neural architecture search for multivariate non-stationary time series forecasting, *Appl. Soft Comput.*, **145** (2023), 110522. <https://doi.org/10.1016/j.asoc.2023.110522>
42. C. Y. Lin, J. A. Lobo-Marques, Stock market prediction using artificial intelligence: A systematic review of systematic reviews, *Soc. Sci. Humanit. Open.*, **9** (2024), 100864. <https://doi.org/10.1016/j.ssaho.2024.100864>
43. Melina, Sukono, H. Napitupulu, N. Mohamed, A conceptual model of investment-risk prediction in the stock market using extreme value theory with machine learning: A semisystematic literature review, *Risks*, **11** (2023), 1–24. <https://doi.org/10.3390/risks11030060>
44. A. Mittal, A. Goel, *Stock prediction using twitter sentiment analysis*, Stanford University, CS229, 2012, 1–5.
45. I. Abdullayev, E. Akhmetshin, I. Kosorukova, E. Klochko, W. Cho, G. P. Joshi, Modeling of extended osprey optimization algorithm with Bayesian neural network: An application on Fintech to predict financial crisis, *AIMS Math.*, **9** (2024), 17555–17577. <https://doi.org/10.3934/math.2024853>

46. J. H. Lee, J. K. Hong, Comparative performance analysis of vibration prediction using RNN techniques, *Electronics*, **11** (2022), 3619. <https://doi.org/10.3390/electronics11213619>
47. W. S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.*, **5** (1943), 115–133. <https://doi.org/10.1007/BF02478259>
48. S. Haykin, *Neural networks and learning machines*, New York: Pearson Education, Inc, 2009.
49. M. I. Jordan, *Attractor dynamics and parallelism in a connectionist sequential machine*, IEEE Press, 1990.
50. J. L. Elman, Finding structure in time, *Cogn. Sci.*, **14** (1990), 179–211. [https://doi.org/10.1016/0364-0213\(90\)90002-E](https://doi.org/10.1016/0364-0213(90)90002-E)
51. S. H. Sung, J. M. Kim, B. K. Park, S. J. Kim, A study on Cryptocurrency log-return price prediction using multivariate time-series model, *Axioms*, **11** (2022), 1–17. <https://doi.org/10.3390/axioms11090448>
52. K. J. Lang, A. H. Waibel, G. E. Hinton, A time-delay neural network architecture for isolated word recognition, *Neural Networks*, **3** (1990), 23–43. [https://doi.org/10.1016/0893-6080\(90\)90044-L](https://doi.org/10.1016/0893-6080(90)90044-L)
53. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
54. S. Zaheer, N. Anjum, S. Hussain, A. D. Algarni, J. Iqbal, S. Bourouis, et al., A multi parameter forecasting for stock time series data using LSTM and deep learning model, *Mathematics*, **11** (2023), 590. <https://doi.org/10.3390/math11030590>
55. K. Zhang, X. Huo, K. Shao, Temperature time series prediction model based on time series decomposition and Bi-LSTM network, *Mathematics*, **11** (2023), 2060. <https://doi.org/10.3390/math11092060>
56. K. Smagulova, A. P. James, A survey on LSTM memristive neural network architectures and applications, *Eur. Phys. J. Spec. Top.*, **228** (2019), 2313–2324. <https://doi.org/10.1140/epjst/e2019-900046-x>
57. Y. Su, C. C. J. Kuo, On extended long short-term memory and dependent bidirectional recurrent neural network, *Neurocomputing*, **356** (2019), 151–161. <https://doi.org/10.1016/j.neucom.2019.04.044>
58. B. C. Mateus, M. Mendes, J. T. Farinha., R. Assis, A. M. Cardoso, Comparing LSTM and GRU models to predict the condition of a pulp paper press, *Energies*, **14** (2021), 6958. <https://doi.org/10.3390/en14216958>
59. M. S. Alam, M. Murshed, P. Manigandan, D. Pachiyappan, S. Z. Abduvaxitovna, Forecasting oil, coal, and natural gas prices in the pre-and post-COVID scenarios: Contextual evidence from India using time series forecasting tools, *Resour. Policy.*, **81** (2023), 103342. <https://doi.org/10.1016/j.resourpol.2023.103342>
60. M. Melina, A. Sambas, S. Firman, Artificial neural network-based machine learning approach to stock market prediction model on the Indonesia stock exchange during the COVID-19, *Eng. Lett.*, **30** (2022), 988–1000.
61. M. G. M. Ghazal, Y. A. Tashkandy, O. S. Balogun, M. E. Bakr, Exponentiated extended extreme value distribution: Properties, estimation, and applications in applied fields, *AIMS Math.*, **9** (2024), 17634–17656. <https://doi.org/10.3934/math.2024857>
62. C. C. Aggarwal, *Neural networks and deep learning*, Cham: Springer International Publishing, 2018.

63. S. Coles, *An introduction to statistical modeling of extreme values*, London: Springer-Verlag London limited, 2001. <https://doi.org/10.1007/978-1-4471-3675-0>
64. A. Ourir, W. Snoussi, Markets liquidity risk under extremal dependence: Analysis with VaRs methods, *Econ. Model.*, **29** (2012), 1830–1836. <https://doi.org/10.1016/j.econmod.2012.05.036>
65. S. I. Hussain., S. Li, Modeling the distribution of extreme returns in the Chinese stock market, *J. Int. Financ. Mark. Institutions Money*, **34** (2015), 263–276. <https://doi.org/10.1016/j.intfin.2014.11.007>
66. M. H. Pham., C. Tsokos, B. J. Choi, Maximum likelihood estimation for the generalized pareto distribution and Goodness-of-Fit test with censored data, *J. Mod. Appl. Stat. Meth.*, **17** (2019), eP2608. <https://doi.org/10.22237/jmasm/1553261471>
67. M. Melina, Sukono, H. Napitupulu, N. Mohamed, Modeling of machine learning-based extreme value theory in stock investment risk prediction: A systematic literature review, *Big Data*, 2024, 1–20. <https://doi.org/10.1089/big.2023.0004>
68. A. J. Mcneil, T. Saladin, *The peaks over thresholds method for estimating high quantiles of Loss distributions*, In: Proceedings of 28th International ASTIN Colloquium, Zurich: ETH Zentrum, 1998, 23–43.
69. A. A. Balkema, L. Haan, Residual life time at great age, *Ann. Probab.*, **2** (1974), 792–804. <https://doi.org/10.1214/aop/1176996548>
70. J. Pickands, Statistical inference using extreme order statistics, *Ann. Stat.*, **3** (1975), 119–131. <https://doi.org/10.1214/aos/1176343003>
71. A. F. Rossignolo, M. D. Fethi, M. Shaban, Value-at-risk models and Basel capital charges: Evidence from emerging and frontier stock markets, *J. Financ. Stab.*, **8** (2012), 303–319. <https://doi.org/10.1016/j.jfs.2011.11.003>
72. C. Omari, S. Mundia, I. Ngina, Forecasting value-at-risk of financial markets under the global pandemic of COVID-19 using conditional extreme value theory, *J. Math. Financ.*, **10** (2020), 569–597. <https://doi.org/10.4236/jmf.2020.104034>
73. K. Kuuster, S. Mittnik, M. S. Paoletta, Value-at-risk prediction: A comparison of alternative strategies, *J. Financ. Econom.*, **4** (2006), 53–89. <https://doi.org/10.1093/jjfinec/nbj002>
74. P. Kupiec, Techniques for verifying the accuracy of risk measurement models, *J. Deriv.*, **3** (1995), 73–84.
75. Z. Zhan, S. K. Kim, Versatile time-window sliding machine learning techniques for stock market forecasting, *Artif. Intell. Rev.*, **57** (2024), 209. <https://doi.org/10.1007/s10462-024-10851-x>



AIMS Press

© 2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)