# A Comparative Study of Deep Learning Models for Bitcoin Price Prediction Using NeuralProphet, RNN, and LSTM

**Tan Khai Lian**
Institute of Computer Science and Digital Innovation, UCSI University, Kuala Lumpur, Malaysia
1001953804@ucsiuniversity.edu.my

**Ismail Ahmad Al-Qasem Al-Hadi**
Institute of Computer Science and Digital Innovation, UCSI University, Kuala Lumpur, Malaysia
Ismailahmed@ucsiuniversity.edu.my

**Mohammad Ahmed Alomari**
Department of Cyber Security, Faculty of Artificial Intelligence and Cyber Security (FAIX), 76100 Durian Tunggal, Melaka, Malaysia
alomari@utem.edu.my (corresponding author)

**Mohammed Nasser Al-Andoli**
Faculty of Computing and Informatics, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia
nasser.alandoli@mmu.edu.my (corresponding author)

**Muhammed Basheer Jasser**
School of Computing and Artificial Intelligence, Faculty of Engineering and Technology, Sunway University, No. 5 Jalan Universiti, Bandar Sunway, 47500 Petaling Jaya, Selangor, Malaysia
basheerj@sunway.edu.my

**AbdulGuddoos S. A. Gaid**
Communication & Computer Engineering Department, Faculty of Engineering & Information Technology, Taiz University, Taiz, Yemen
quddoos.gaid@taiz.edu.ye

*Received: 18 April 2025 | Revised: 30 June 2025, 17 July 2025, and 5 August 2025 | Accepted: 8 August 2025*

*DOI: https://doi.org/10.48084/etasr.11590*

## ABSTRACT

**Bitcoin has recently emerged as a leading asset in the cryptocurrency market. However, its significant price volatility presents challenges for accurate prediction. Due to this volatility, forecasting Bitcoin prices accurately is difficult and complicates decision-making for investors and traders in the cryptocurrency space. This research compares the accuracy of three prediction models: Long Short-Term Memory (LSTM), Recurrent Neural Network (RNN), and Facebook's NeuralProphet, introduced in 2021, focusing on improving Bitcoin price forecasting accuracy. The study uses daily Bitcoin prices from the past five years to assess model performance. Results indicate that the LSTM model outperforms both NeuralProphet and RNN in prediction accuracy. This comparison holds substantial economic significance, as accurate predictions can assist investors and traders in making informed decisions within the cryptocurrency market.**

***Keywords-Bitcoin; cryptocurrency; Long Short-Term Memory (LSTM); Recurrent Neural Network (RNN); NeuralProphet***

# I. INTRODUCTION

Bitcoin, a revolutionary digital currency introduced in 2009, has reshaped the financial landscape by functioning without a central authority, utilizing a decentralized network called blockchain [1]. This advancement guarantees transparency, security, and resistance to censorship. The finite supply of 21 million coins has notably contributed to Bitcoin's value, leading to its designation as "digital gold" and a secure store of value [2]. The cryptocurrency's pseudonymous transactions have spurred discussions about privacy concerns and potential links to illicit activities. The price history of Bitcoin has been characterized by significant volatility. Advocates commend its potential to revolutionize finance, whereas skeptics raise concerns about volatility, regulatory challenges, and environmental impact. Bitcoin's influence extends to acting as a benchmark for the wider cryptocurrency market, drawing attention from individuals, investors, and institutions globally. As Bitcoin continues to integrate into various industries, emphasis is placed on caution and comprehensive research due to the inherent risks and volatility within the cryptocurrency market. Informed decision-making remains crucial for investors considering involvement in this dynamic and evolving space.

Deep Neural Network (DNN) is a type of Artificial Neural Network (ANN) that consists of multiple interconnected layers of neurons. Its "deep" structure involves numerous hidden layers between input and output layers, setting it apart from traditional neural networks. Within a DNN, each neuron processes inputs from the preceding layer, conducts calculations, and transmits results to the subsequent layer. This iterative process continues across layers until the final layer generates an output [3]. Neurons within each layer utilize non-linear activation functions on weighted inputs, enabling the network to capture intricate patterns in data. During training, optimization algorithms such as gradient descent adjust neuron weights and biases to minimize differences between network predictions and desired outputs [4]. DNNs have exhibited remarkable achievements across various domains, encompassing computer vision, natural language processing, speech recognition, and reinforcement learning [5, 6]. Their capacity to acquire hierarchical representations from data has rendered them highly effective in handling complex, information-intensive tasks.

Various studies demonstrate the potential of deep learning models, such as NeuralProphet, Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) for Bitcoin price prediction. However, they are limited by factors such as the volatility and unpredictability of cryptocurrency markets, potential overfitting due to limited historical data, and the models' reliance on past trends, which may not fully capture future market anomalies or external events. Some works explore individual models (e.g., LSTM or RNN), but there is a scarcity of rigorous head-to-head comparisons across multiple models, like NeuralProphet, RNN, and LSTM, under the same experimental conditions. Additionally, deep learning models are often treated as black boxes in existing research, with minimal effort to interpret predictions or explain the decision-making processes, reducing trust and usability in real-world trading systems. Furthermore, many models ignore or underutilize influential external variables such as news sentiment, regulatory events, macroeconomic indicators, and social media trends, which are critical in crypto markets. Many studies also focus on predictive accuracy without considering practical trading constraints, such as transaction fees, slippage, liquidity, or execution delays, which can significantly affect profitability [4, 7].

Authors in [8] designed the cryptocurrency Price Prediction Model to forecast the prices of Bitcoin (BTC), Litecoin (LTC), and Ethereum (ETH) by leveraging historical data and deep learning techniques. Utilizing algorithms such as LSTM, Gated Recurrent Unit (GRU), and Bidirectional LSTM (bi-LSTM), the GRU algorithm demonstrated the lowest error rate, indicating superior predictive performance. Consequently, the GRU model emerges as an excellent choice for comprehending cryptocurrency prices, potentially assisting investors and traders in decision-making. Future research could delve into additional factors, such as social media activity and trading volume to further enhance predictions.

Authors in [9] employed Linear Regression (LR) and LSTM models to predict Bitcoin prices using three years of historical data. The LSTM model exhibited superior prediction performance compared to LR. Additionally, a tkinter-based graphical user interface was developed, allowing users to input feature values and acquire Bitcoin price predictions directly from the website. Authors in [10] report that LSTM also outperforms Auto Regressive Integrated Moving Average (ARIMA) models when using historical data and sentiment scores, suggesting that incorporating sentiment analysis enhances price prediction accuracy. Moreover, authors in [11] applied LSTM and GRU to Bitcoin price datasets, with GRU demonstrating superior efficiency over LSTM in most scenarios and requiring less compilation time. The data were trained using sliding windows, where the window size and the number of days positively impacted the performance of each model.

Bitcoin prices experienced a sharp decline in 2018, dropping approximately 80% from their peak, creating negative sentiment in the market and resulting in a long adjustment period for this cryptocurrency. Bitcoin prices began to recover gradually in the second half of 2019. Authors in [12] used historical data spanning from September 17, 2014, to May 25, 2023 to train and assess the performance of the NeuralProphet model, employing metrics such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The authors report that NeuralProphet reliably forecasts Bitcoin prices by effectively capturing intricate patterns and trends within its price movements.

In [13], various models such as RNN, LSTM, and LR were employed to forecast Bitcoin prices using minute-by-minute data from 2012 to 2019. These models were applied to predict future prices, and metrics like RMSE, MAE, and coefficient of determination ($R^2$) were utilized to assess prediction accuracy. The study revealed that RNN with LSTM outperforms LR primarily due to its ability to capture longer-term dependencies within the data. Nevertheless, improvements remain necessary, achievable through feature extensions, parameter tuning, and experimentation with alternative neural network models.

Authors in [14] conducted an analysis using Theil-Sen, Huber, LSTM, and GRU models applied on 1-minute trading data from the Bitstamp website, spanning from 2012 to 2018. The objective was to learn and compare the actual and predicted prices within the initial 20 days of the test data. The findings revealed that the GRU model exhibited the highest performance, followed by the LSTM, Huber, and Theil-Sen models, in that sequence. Although the Huber model demonstrated the fastest computation time among all models, the results remained unsatisfactory, emphasizing the necessity for further improvements and analysis of additional features. Authors in [15] utilized RNN and LSTM methods for Bitcoin price forecasting. To address challenges posed by Bitcoin's unpredictable nature, Bayesian-optimized RNN and LSTM networks were employed. Wavelet coherence analysis explored correlations between Bitcoin price fluctuations and factors such as hash rate, difficulty, and Google Trends views. The LSTM network attained 52% accuracy with an RMSE of 8%, whereas the RNN achieved 50% accuracy with an RMSE of 9%. Both outperformed the ARIMA model, which achieved 46% accuracy and an RMSE of 11%. GPU training was 67.7% faster than CPU training, demonstrating deep learning's potential for capturing complex, time-dependent patterns. Authors in [16] utilized ARIMA, Random Forest (RF), Support Vector Machine (SVM), LSTM, and WaveNet to forecast Bitcoin prices and reported that these models display relatively low performance in long-term predictions, reflecting the influence of random components.

Furthermore, authors in [17] employed Bi-LSTM and GRU to predict cryptocurrency prices using three datasets from Yahoo Finance, which contain daily data on three cryptocurrencies. The parameters of both models were optimized through grid search, and their performance was evaluated using metrics such as Mean Squared Error (MSE), RMSE, MAE, MAPE, and $R^2$ score. GRU outperformed Bi-LSTM for all three cryptocurrencies. Authors in [18] used LSTM to predict cryptocurrency prices for five coins (BTC, ETH, BNB, ADA, DOGE) using data from January 2018 to September 2021. They reported that the best prediction performance was obtained when predicting the DOGE coin with 20 epochs, with an RMSE value of 0.0630. They also concluded that LSTM is not suitable for predicting coins with high prices and high volatility, such as BNB, ETH, and BTC. Authors in [19] used an LSTM model to predict future stock market prices for two assets (GOOGL and NKE) while studying the effect of different training epochs on the model accuracy. They used MSE as the metric to measure model loss and varied the number of training epochs (12, 25, 50, and 100). The study inferred that increasing the number of epochs enhances forecasting precision and that training with less data but more epochs can also improve testing results. Authors in [20] proposed a method to predict cryptocurrency prices using a deep RNN with LSTM units. They reported that their approach effectively captures time-dependent and nonlinear features in cryptocurrency market data. Historical data of Bitcoin, Ethereum, and Ripple from https://www.cryptodatadownload.com were used as input. They compared their RNN-LSTM model with other machine learning methods, such as Naive Bayes, RF, SVM, and Multilayer Perceptron, using MSE as the evaluation metric. The results indicated that the RNN-LSTM model outperformed all other methods in terms of MSE for both 1-hour and 1-day prediction windows. Finally, authors in [21] developed a deep learning model that combines LSTM with sentiment analysis to predict the future prices of three cryptocurrencies: Bitcoin, Ethereum, and Binance. They collected two types of data: historical cryptocurrency price data from Yahoo Finance and social media posts from Twitter and Reddit. The study demonstrated that their model achieved higher accuracy in forecasting cryptocurrency prices than other models, including ARIMA, Convolutional Neural Network (CNN), and RNN.

Table I presents a comparison between resent related works and this comparative study, highlighting specific shortcomings and limitations in previous research that create gaps for new investigations. The table outlines various aspects such as dataset used, models and methodologies applied, and performance metrics. Through this comparison, we aim to clearly emphasize the novelty and contributions of our work. Specifically, our approach distinguishes itself by utilizing an extended time series dataset combined with a multi-architecture approach involving LSTM, RNN, and NeuralProphet, with extensive evaluation across multiple forecasting horizons.

TABLE I. COMPARISON AND LIMITATIONS OF RELATED STUDIES ON CRYPTOCURRENCY PRICE PREDICTION

| **Ref. / year** | **Categories / criteria** | | | |
|---|---|---|---|---|
| | **Comparison of three prediction models: LSTM, RNN, and NeuralProphet** | **Comprehensive review of related work** | **Use of RMSE and MAPE metrics** | **Use of a dataset from the most recent 5 years** |
| [21] / 2023 | | ✓ | ✓ | ✓ |
| [20] / 2023 | | ✓ | ✓ | |
| [18] / 2022 | | | ✓ | |
| [17] / 2022 | | ✓ | ✓ | ✓ |
| [15] / 2018 | | | ✓ | |
| [13] / 2020 | | ✓ | ✓ | |
| [12] / 2023 | | | ✓ | ✓ |
| [11] / 2020 | | ✓ | | |
| This study | ✓ | ✓ | ✓ | ✓ |

Based on a rigorous review of related existing studies, this work focuses on improving Bitcoin price forecasting accuracy using three deep learning models. The paper mainly aims to compare the accuracy of established prediction models with the recently proposed NeuralProphet model to determine the most precise method for predicting Bitcoin's future price movements.

## II. METHODOLOGY

This study aims to forecast the daily price of Bitcoin by leveraging advanced deep learning techniques that select and utilize relevant features. The performance of the models is evaluated using error metrics, including MAPE and RMSE, to identify the most reliable approach for Bitcoin price prediction. Specifically, the study employs RNN, LSTM, and the NeuralProphet model developed by Facebook in 2021, which are among the most effective methods for modeling the highly volatile market of Bitcoin. The process of developing these models is illustrated in Figure 1.

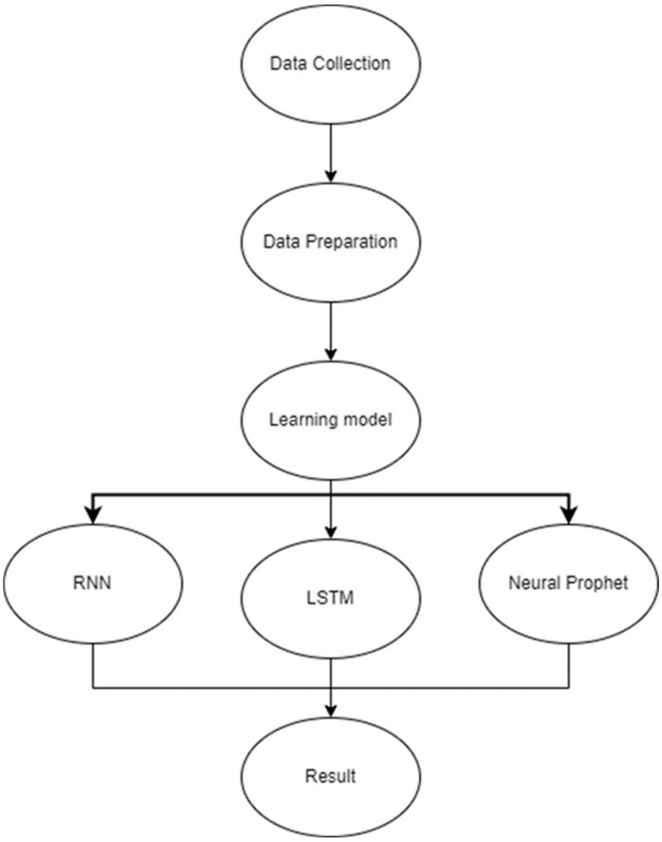

Fig. 1. Process for developing the prediction models.

### A. Recurrent Neural Networks

RNNs are specialized for handling sequential data such as time series and natural language [7]. They differ from feedforward neural networks by allowing information flow between layers, making them effective for tasks such as recognizing unsegmented handwriting or speech [22]. One key feature of RNNs is parameter sharing across layers, which reduces the number of parameters the model needs to learn. This parameter sharing is crucial for efficiently processing sequential data of varying lengths. RNNs use Backpropagation Through Time (BPTT) to calculate gradients and learn from sequence data. BPTT is a variant of the traditional backpropagation algorithm, accumulating errors at each time step. However, RNNs encounter challenges like exploding and vanishing gradients during learning, which depend on the magnitude of the gradient. To address these issues, one approach is to use fewer hidden layers, simplifying the RNN model.

The RNN architecture consists of an input layer for input sequences, a hidden layer for processing and storing information, and an output layer for generating output sequences. An activation function is employed to make the hidden layer outputs more nonlinear. BPTT, designed for sequential data, is used to calculate gradients in RNNs [23-25].

### B. Long Short-Term Memory

LSTM networks are a specialized variant of RNNs that excel in managing long-term dependencies in sequential data. LSTM networks were created primarily for addressing sequential prediction tasks [26]. These networks can learn the order of data in sequences, making them widely applied in areas such as language modeling, speech recognition, and text generation. LSTM networks employ a chain-like structure that allows them to retain information for a long time [27].

In LSTM, the neurons that perform the computations are called memory cells or cells, similar to standard neural networks. Each memory cell contains weights and gates, which distinguish LSTM from traditional RNNs. A standard LSTM unit consists of three gates, the input gate, forget gate, and output gate, which control the information that goes in and out of the cell and help the cell remember the data [28]:

- Forget gate: The forget gate decides which information to keep or discard from the previous steps. It uses a sigmoid function that takes both the previous hidden state and the current input as inputs, making the network more efficient. The forget gate selects the relevant information from the past that is needed.
- Input gate: The input gate brings data into the cells. It uses a sigmoid function that combines the previous hidden state and the current input, selecting relevant past information.
- Output gate: The output gate determines the information to be passed to the next hidden state, combining data from the memory cell and previous inputs, helping in making predictions.

### C. NeuralProphet

NeuralProphet represents the next iteration of the Prophet algorithm initially developed by Facebook in 2021, serving as a powerful tool for forecasting time series data [29]. This advanced algorithm has been designed with a focus on simplicity, making it an accessible framework for interpretable time series prediction. NeuralProphet uses a fusion of classic components and neural networks to produce highly accurate time series forecasts quickly. It encompasses all the core components found in the original Prophet model, including trend, seasonality, recurring events, and regressors, and adds enhanced capabilities such auto-regression and handling of lagged covariates. These features are particularly useful in scenarios where the current state affects the immediate future, which is common in applications such as energy consumption, traffic flow, and environmental monitoring [30]. For example, if a server's load suddenly increases, it may result from recent events with effects that persist over time, a detail that NeuralProphet can capture in its short-term predictions. In short, NeuralProphet is well-suited for smaller datasets and complex data structures, offering flexibility, GPU support, and improved forecasting accuracy.

The fundamental elements of NeuralProphet include two primary components: the trend component and the seasonal component. The trend component characterizes general trends or long-term patterns in the data, whereas the seasonal component captures recurring patterns over time. Beyond these, NeuralProphet can also incorporate the regression component, accommodating external factors influencing the time series data. For instance, it can include the price of rice as a regressor in rice sales data. Furthermore, NeuralProphet's strength lies in its ability to automatically tune many hyperparameters associated with the model and training process, such as the number of seasonal components, confidence intervals, and optimization methods, adapting to the provided dataset to enhance forecasting performance.

## III. IMPLEMENTATION

This study develops a predictive model for Bitcoin price forecasting by comparing three deep learning approaches: RNN, LSTM and NeuralProphet. Each model is trained using the same number of epochs and batch size to ensure a fair comparison. However, accurately predicting Bitcoin prices remains challenging due to its high volatility.

### *A. Data Acquisition*

The initial steps in model development involve data acquisition and preprocessing, which include gathering, combining, organizing, and structuring data for further analysis. These steps are essential for preparing the data for activities such as data visualization, analytics, and machine learning applications [31]. Ensuring high data quality is essential for reliable model training and evaluation.

The dataset used in this study spans from 15 May 2018 to 14 May 2023, comprising 1,827 rows of data. The data were sourced from Yahoo Finance, a widely used and reliable platform for financial market information, including cryptocurrencies. The dataset includes key features such as opening price, closing price, high, low, adjusted closing price, and trading volume, all of which are essential for time series forecasting and volatility analysis. Yahoo Finance has been a reputable source for financial data in academic and industrial research due to its accessibility and historical depth [17, 18, 32]. The use of daily-level granularity over a 5-year period ensures sufficient variability and temporal patterns, which are crucial for training and validating deep learning models aimed at price prediction. Table II presents a sample of the Bitcoin dataset collected from Yahoo Finance (https://finance.yahoo.com/).

TABLE II. SAMPLE OF BITCOIN DATASET COLLECTED FROM YAHOO FINANCE

| **Date** | **15/5/2018** | **16/5/2018** | **17/5/2018** | **18/5/2018** |
|---|---|---|---|---|
| **Open** | 8,705.20 | 8,504.40 | 8,370.10 | 8,091.80 |
| **High** | 8,836.20 | 8,508.40 | 8,445.50 | 8,274.10 |
| **Low** | 8,456.50 | 8,175.50 | 8,054.10 | 7,974.80 |
| **Close** | 8,510.40 | 8,368.80 | 8,094.30 | 8,251 |
| **Adj Close** | 8,510.38 | 8,368.83 | 8,094.32 | 8,251 |
| **Volume** | 6,705,710,080 | 6,760,220,160 | 5,862,530,048 | 5,764,190,208 |

### *B. Data Preparation and Feature Selection*

The dataset, stored in .csv format, is imported using the Pandas library function read_csv. Not relevant fields are removed, and missing values are filled to make the data ready for modeling. The dataset includes columns such as date, open, high, low, close, adjusted close, and volume. Finally, in this study, the date column is used as the index, and the closing price is selected as the target variable for prediction.

### *C. Data Normalization*

For the LSTM and RNN models, data normalization is applied to ensure that all features have a similar range. The Min-Max scaler is used to transform each feature into values between 0 and 1, based on the minimum and maximum values of the dataset. The transformation is defined as follows:

$$x_{sc} = \frac{x-\min(x)}{\max(x)-\min(x)} \tag{1}$$

### *D. Data Splitting*

This study focuses on time series prediction rather than classification. The methodology utilizes a sliding window approach with two distinct time windows: a long-term window of one year for training and a short-term window of one month for testing. Both windows are shifted forward by one month in each iteration. The training consists of 1,416 data points (approximately 4 years), whereas the testing dataset contains 365 data points (1 year).

The dataset is divided into three example windows as follows:

- First window: Training period from 2021-01-01 to 2021-12-31, with testing from 2022-01-01 to 2022-01-31.
- Second window: Training from 2021-02-01 to 2022-01-31, with testing from 2022-02-01 to 2022-02-28.
- Third window: Training from 2021-03-01 to 2022-02-28, with testing from 2022-03-01 to 2022-03-31.

Figure 2 illustrates the division of these windows. This sliding window technique enables continuous training and testing on updated data, improving the accuracy of Bitcoin price predictions.

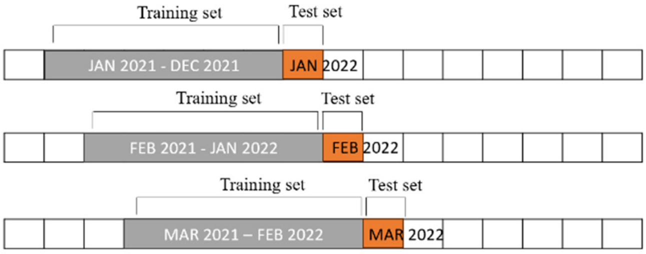

Fig. 2. Example of sliding windows for training and testing.

### *E. Data Sequence Creation*

Data sequence creation is responsible for generating input sequences and corresponding target values that will be used to train the RNN and LSTM models. In this study on time series prediction, creating sequences in this manner allows the RNN to learn patterns and relationships between past data points and the next data point. These sequences are essential for training the model to make predictions based on historical data. Figure 3 presents an example of Python code illustrating the preprocessing stages implemented by the proposed models.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM
from keras.layers import Dropout
import tensorflow as tf
# Load the dataset
data = pd.read_csv('BTCprice.csv')
# Convert 'Date' column to datetime format
data['Date'] = pd.to_datetime(data['Date'],
                format='%Y/%m/%d')
# Extract the 'Close' prices as the target variable
target = data['Close'].values
# Normalize the data to scale between 0 and 1
scaler = MinMaxScaler(feature_range=(0, 1))
target = scaler.fit_transform(target.reshape(-1,
         1))
# Split data into training and testing sets
train_data = target[(data['Date'] >= '2021-01-01')
             & (data['Date'] < '2021-12-31')]
test_data  = target[(data['Date'] >= '2022-01-01')
             & (data['Date'] < '2022-1-31')]
# Prepare the data for LSTM
def create_sequences(dataset, seq_length):
    dataX, dataY = [], []
    for i in range(len(dataset) - seq_length):
        dataX.append(dataset[i:(i + seq_length)])
        dataY.append(dataset[i + seq_length])
    return np.array(dataX), np.array(dataY)
seq_length = 1
trainX, trainY = create_sequences(train_data,
                 seq_length)
testX, testY = create_sequences(test_data,
                seq_length)
```

Fig. 3. Example code for data preprocessing.

## F. Modeling

In this research, a deep learning-based regression model is selected due to the continuous nature of Bitcoin price values. The Keras library is used to create the LSTM and RNN models, whereas the NeuralProphet library is used to create the NeuralProphet models. The parameters for epochs and batch size are kept consistent across all models, with the number of epochs set to 100 and the batch size set to 1. This configuration enables the model to learn from each individual data point, which is especially important given the high volatility of Bitcoin prices.

Using a batch size of 1 allows the model to update its weights after each data point, enabling it to respond quickly to changes in the data and capture subtle fluctuations in Bitcoin prices more effectively. However, this also means that training takes longer because the model performs frequent updates, which can be computationally intensive.

This approach is crucial in time series forecasting, where the ability to adapt to new information rapidly is more valuable than computational efficiency gained from larger batch sizes. Figures 4, 5, and 6 provide example code for RNN, LSTM, and NeuralProphet modeling, respectively.

### 1) Recurrent Neural Network

A sequential model is constructed to define the neural network layer by layer. A SimpleRNN layer with 50 units serves as the input layer to process sequences of data, whereas a dense layer with 1 unit acts as the output layer to make the final prediction. The model uses the "mean_squared_error" function for loss calculation and the "adam" optimizer to enhance model performance. Figure 4 illustrates the main coding stages of the RNN model.

```
# Build the RNN model
model = Sequential()
model.add(SimpleRNN(50, input_shape=(seq_length,
          1)))
model.add(Dense(1))
model.compile(loss='mean_squared_error',
              optimizer='adam')
# Train the model
model.fit(trainX, trainY, epochs=100,
          batch_size=1, verbose=2)
# Make predictions on the test set (January 2023
to May 2023)
test_predictions = model.predict(testX)
# Inverse transform the predictions to get the
original scale
test_predictions =
        scaler.inverse_transform(test_predictions)
```

Fig. 4. Example code for RNN modeling.

### 2) Long Short-Term Memory

A sequential model is created using Keras, comprising two LSTM layers with 50 units each to process sequential data. To mitigate overfitting, dropout layers are included after each LSTM layer. The final layer is a dense layer with one unit to produce the prediction. The model employs the "mean_squared_error" function to measure loss and the "adam" optimizer for optimization. Figure 5 illustrates the main coding stages of the LSTM model.

```
#build the LSTM model
model = Sequential()
model.add(LSTM(50, return_sequences=True,
      input_shape=(seq_length, target.shape[1])))
model.add(Dropout(0.2))
model.add(LSTM(50, return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error',
              optimizer='adam')
# Train the model
model.fit(trainX, trainY, epochs=100,
          batch_size=1, verbose=2)
# Make predictions on the test set (January 2022)
test_predictions = model.predict(testX)
# Inverse transform the predictions to get the
original scale
test_predictions =
        scaler.inverse_transform(test_predictions)
# Keep track of the dates for your test data
test_dates = data['Date'][(data['Date'] >= '2022-
             01-01') & (data['Date'] < '2022-1-
             31')][seq_length:]
```

Fig. 5. Example code for LSTM modeling.

### 3) NeuralProphet

A NeuralProphet model is created with specific configuration parameters, where the batch size is set to 1, and the model is trained for 100 epochs. Figure 6 presents the main coding steps for this model. The freq='D' parameter in NeuralProphet specifies the frequency of the time series data, where 'D' stands for daily data. It tells the model that the data points are recorded daily, influencing how it models trends, seasonality, and other time-based components.

```
# Build the NeuralProphet model
model = NeuralProphet(batch_size=1, epochs=100)
# Train the NeuralProphet model
start_time = time.time()
model.fit(train_data, freq='D')
end_time = time.time()
# Testing the NeuralProphet model
future = model.make_future_dataframe(test_data,
         periods=test_size)
forecast = model.predict(future)
```

Fig. 6. Example code for NeuralProphet modeling.

## G. *Evaluation Metrics*

Evaluation metrics are a tool that helps us evaluate the performance of a classification model on a test dataset [33]. They indicate how well the model correctly labels the data and where it makes mistakes. We use two metrics to assess how accurate the model is: RMSE and MAPE. The RMSE is a metric that measures the average distance between the actual values and the model's predictions in the dataset, as shown in (2):

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - y_i)^2}{N}} \qquad (2)$$

where $N$ is the total number of data points (patterns), $x_i$ is the predicted value for pattern $i$, and $y_i$ is the actual value for pattern $i$. In addition, MAPE is a metric that indicates how much the model's predictions deviate from the observed values in percentage terms. The equation of MAPE is given in (3):

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{x_i - y_i}{y_i}\right| \times 100 \qquad (3)$$

where $N$ is the number of patterns, $|\cdot|$indicates the absolute error, $x_i$ is the predicted value for pattern $i$, and $y_i$ is the actual value for pattern $i$. Figure 7 shows an example of the performance calculation code for the three models.

```
from sklearn.metrics import
mean_absolute_percentage_error, mean_squared_error
# Inverse transform the true prices to their
original scale
true_prices = scaler.inverse_transform(testY)
# Calculate the MAPE
mape = mean_absolute_percentage_error(true_prices,
        test_predictions)
# Calculate the RMSE
rmse = np.sqrt(mean_squared_error(true_prices,
        test_predictions))
print(f"Root Mean Squared Error (RMSE): {rmse}")
print("Mean Absolute Percentage Error (MAPE):
      {:.2f}%".format(mape))
```

Fig. 7. Example code for the performance evaluation part.

# IV. RESULTS AND DISCUSSION

To compare all the results, the model with the lowest RMSE and MAPE is considered the most accurate. According to these metrics, the LSTM model outperforms the other two models in predicting Bitcoin prices. It achieves the lowest RMSE and MAPE due to its superior ability to capture long-term dependencies in the data compared to RNN and NeuralProphet, as it relies more effectively on past price data. Figures 8,9, and 10 illustrate the comparison between actual and predicted values for the RNN, LSTM, and NeuralProphet models, respectively, showing only a few instances where forecasts deviate from actual outcomes. In this study, predictions are evaluated using both one-year and one-month intervals.

## A. *One-Year Interval*

Table III shows the RNN model's Bitcoin price prediction results, with a MAPE of 0.03 and RMSE of 825.6, making it the second-best among the models. The predicted prices average 22,799.82 USD, with a maximum of 32,719.97 USD and a minimum of 16,173.67 USD, whereas actual prices average 22,204.62 USD, with a maximum of 31,792.31 USD and a minimum of 15,787.28 USD. The difference in average predicted and actual prices is 595.20 USD.

TABLE III. RNN MODEL PREDICTION RESULTS

| Model | RMSE | MAPE |
|---|---|---|
| RNN | 825.61 | 0.03 |

Figure 8 shows the difference between actual and predicted values for the RNN model, demonstrating close alignment with the actual prices and only a few deviations.

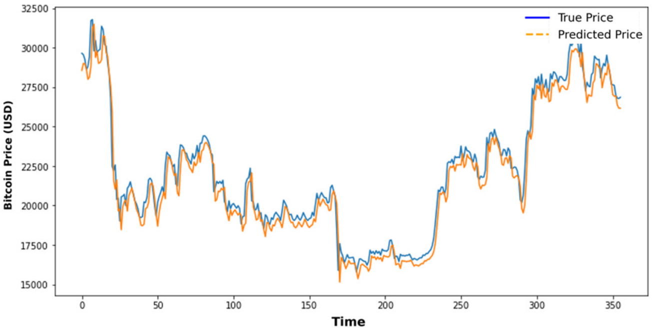

Fig. 8. Comparison of predicted and actual Bitcoin prices using the RNN model.

Table IV presents the LSTM model's Bitcoin price predictions, with a MAPE of 0.02 and RMSE of 751.17, making it the most accurate model. Predicted prices average 22,304.15 USD, with a maximum of 31,205.36 USD and a minimum of 16,351.06 USD, whereas actual prices average 22,204.62 USD, with a maximum of 31,792.31 USD and a minimum of 15,787.28 USD. The difference in average values is 99.53 USD.

TABLE IV. LSTM MODEL PREDICTION RESULTS

| Model | RMSE | MAPE |
|---|---|---|
| LSTM | 751.17 | 0.02 |

Figure 9 shows the LSTM model's predictions compared to actual prices. The close alignment between the two lines indicates the model's high accuracy with minimal deviations.

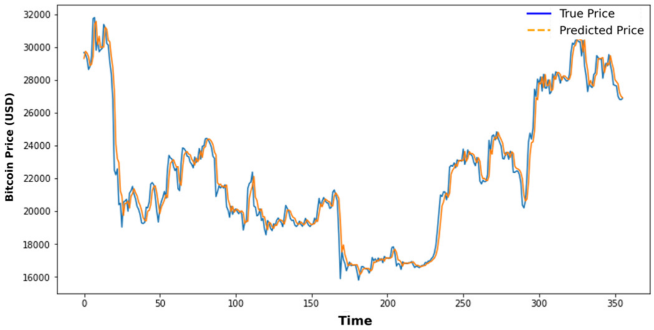

Fig. 9. Comparison of predicted and actual Bitcoin prices using the LSTM model.

Table V shows the NeuralProphet model's Bitcoin price predictions, with a MAPE of 0.32 and RMSE of 10,090.53, making it the least accurate model. Predicted prices average 15,113.65 USD, with a maximum of 21,637.08 USD and a minimum of 2,698.45 USD, whereas actual prices average 22,204.62 USD, with a maximum of 31,792.31 USD and a minimum of 15,787.28 USD. The difference in average values is 7,090.97 USD.

TABLE V. NEURALPROPHET MODEL PREDICTION RESULTS

| Model | RMSE | MAPE |
|---|---|---|
| Neuralrophet | 10,090.53 | 0.32 |

Figure 10 shows a comparison of actual and predicted Bitcoin prices using the NeuralProphet model. The noticeable gaps between the two lines indicate significant deviations, highlighting the model's poor performance.

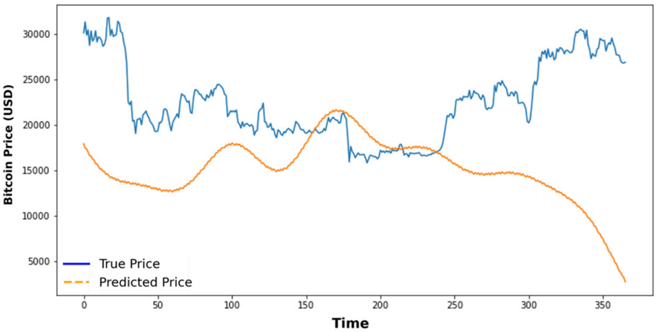

Fig. 10. Comparison of predicted and actual Bitcoin prices using the NeuralProphet model.

Figures 11 and 12 compare the prediction accuracy of the LSTM, RNN, and NeuralProphet models using RMSE and MAPE. LSTM consistently outperforms the others with the lowest RMSE (751.17) and MAPE (0.02). RNN follows with slightly higher errors (RMSE of 825.61 and MAPE of 0.03), whereas NeuralProphet shows poor performance, with a significantly higher RMSE (10,090.53) and MAPE (0.32). These results demonstrate that LSTM is the most accurate model for predicting Bitcoin prices, whereas NeuralProphet struggles with large deviations.

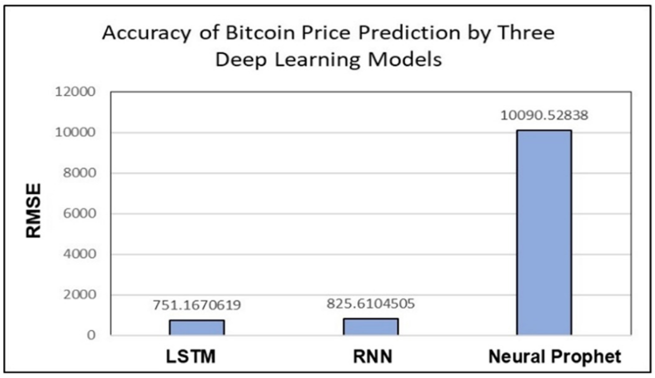

Fig. 11. Comparison of prediction performance among the tested models by RMSE metric.

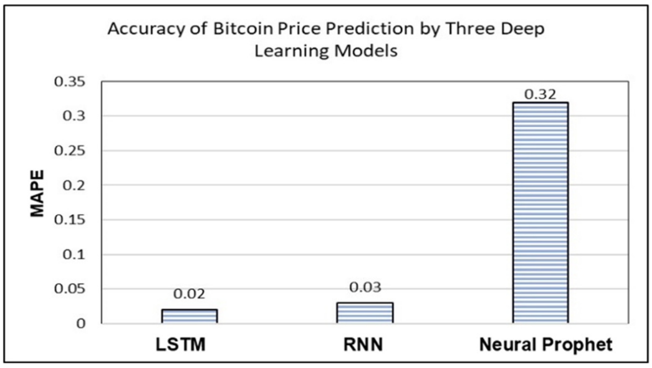

Fig. 12. Comparison of prediction performance among the tested models by MAPE metric.

Several factors may explain why NeuralProphet performs worse. First, RNN and LSTM models are specifically designed to handle sequential data effectively, as they can retain past information, which may provide an advantage over the NeuralProphet model. Second, LSTM and RNN models can capture long-term dependencies within the data, which is likely crucial for accurate Bitcoin price prediction. In contrast, while the NeuralProphet model is effective in many scenarios, it may not capture these dependencies as effectively.

## *B. One-Month Interval*

For the one-month prediction windows, the RNN model achieved the following performance: in the first window (2021-01-01 to 2021-12-31 for training, 2022-01-01 to 2022-01-31 for testing), the MAPE was 0.06 and RMSE was 2,597.00. In the second window (2021-02-01 to 2022-01-31 for training, 2022-02-01 to 2022-02-28 for testing), the MAPE was 0.03 and RMSE was 1,478.78. In the third time window (2021-03-01 to 2022-02-28 for training, 2022-03-01 to 2022-03-28 for testing), the MAPE was 0.03 and RMSE was 1,438.81. On average, across the three-time windows, the MAPE was 0.04 and the RMSE was 1,838.20. The RNN is considered the second-best model for one-month predictions. Table VI summarizes the results, and Figures 13 compares predicted and actual Bitcoin prices for January, February, and March 2022.

TABLE VI. RNN MODEL PREDICTION RESULTS (ONE-MONTH INTERVAL)

| Time window | RMSE | MAPE |
|---|---|---|
| Jan 2022 | 2,597.00 | 0.06 |
| Feb 2022 | 1,478.78 | 0.03 |
| Mar 2022 | 1,438.81 | 0.03 |
| Average | 1,838.20 | 0.04 |

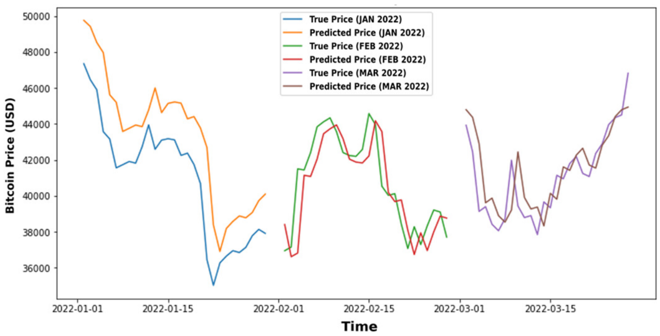

Fig. 13. Comparison of predicted and actual Bitcoin prices using the RNN model (Jan/Feb/Mar 2022).

For the same windows, the LSTM model performed as follows: in the first-time window, the MAPE was 0.02 and RMSE was 1,278.57. In the second window, MAPE was 0.03 and RMSE was 1,767.14. In the third window, MAPE was 0.03 and RMSE was 1,441.30. The average MAPE and RMSE across the three windows were 0.03 and 1495.67, respectively. LSTM was the best model for one-month predictions. Table VII summarizes these results, and Figure 14 compares predicted and actual Bitcoin prices for January, February, and March 2022.

TABLE VII. LSTM MODEL PREDICTION RESULTS (ONE-MONTH INTERVAL)

| Time window | RMSE | MAPE |
|---|---|---|
| Jan 2022 | 1,278.57 | 0.02 |
| Feb 2022 | 1,767.14 | 0.03 |
| Mar 2022 | 1,441.30 | 0.03 |
| Average | 1,495.67 | 0.03 |

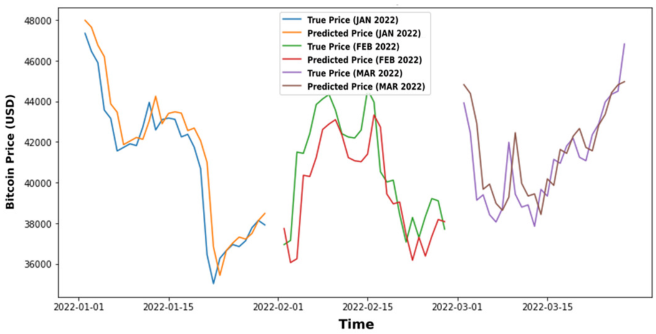

Fig. 14. Comparison of predicted and actual Bitcoin prices using the LSTM model (Jan/Feb/Mar 2022).

The NeuralProphet model achieved the following: in the first window, MAPE was 0.04 and RMSE was 1,756.90. In the second window, MAPE was 0.20 and RMSE was 8,559.15. In the third window, MAPE was 0.21 and RMSE was 9,556.85. The average MAPE and RMSE across the three windows were 0.15 and 6,624.30, respectively. NeuralProphet performed the worst for one-month predictions. Table VIII summarizes these results, and Figure 15 compares predicted and actual Bitcoin prices for January, February, and March 2022.

TABLE VIII. NEURALPROPHET MODEL PREDICTION RESULTS (ONE-MONTH INTERVAL)

| Time window | RMSE | MAPE |
|---|---|---|
| Jan 2022 | 1,756.90 | 0.04 |
| Feb 2022 | 8,559.15 | 0.20 |
| Mar 2022 | 9,556.85 | 0.21 |
| Average | 6,624.30 | 0.15 |

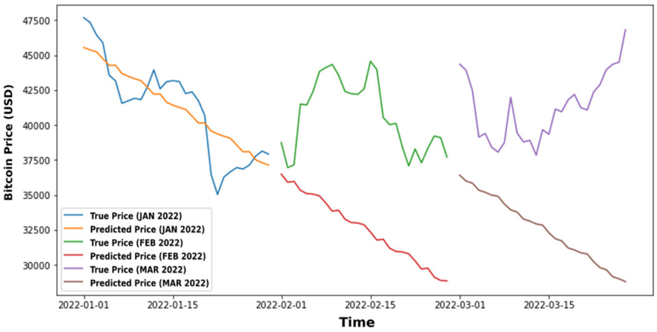

Fig. 15. Comparison of predicted and actual Bitcoin prices using the NeuralProphet model (Jan/Feb/Mar 2022).

### *C. Study Limitations*

This study has several limitations. First, the predictions were based on a single column of data, which may not capture the full complexity of cryptocurrency price movements. Second, the study focused mainly on Bitcoin, a cryptocurrency with high market capitalization. This means that the findings may not be applicable to newly launched Initial Coin Offering (ICO) coins which have lower market capitalizations and potentially different market behaviors. Finally, training deep learning models, especially LSTM, can be time-consuming. This may make it challenging to keep up with the rapid fluctuations in cryptocurrency prices, as the models may not adapt quickly enough to sudden market shifts.

## V. CONCLUSION

This study demonstrates that the three deep learning models, including NeuralProphet, Long Short-Term Memory (LSTM), and Recurrent Neural Network (RNN) models, are capable of predicting Bitcoin prices and were effectively applied in this study. All models perform well in capturing the overall trends and patterns of Bitcoin price movements and can assist traders in identifying potential buying or selling opportunities.

However, the high volatility of Bitcoin makes accurate forecasting inherently challenging. Among the models tested using the same parameters and various time windows, the LSTM model achieved the highest prediction accuracy. This improved performance comes at the cost of longer execution times due to the model's structural complexity. In contrast, the NeuralProphet model achieved the lowest accuracy. Although it is designed for business time series forecasting, NeuralProphet might not capture the complex patterns in the data as effectively as LSTM and RNN. Nevertheless, NeuralProphet, as a newly launched model by Facebook in 2021, has potential for improved performance through further development and optimization.

We can conclude that deep learning models are suitable for Bitcoin price prediction and can provide valuable insights for investors and traders. Future work could focus on developing an intuitive user interface to provide easy access to predictions and datasets, facilitating the exploration of additional deep learning models to identify the most effective approach for forecasting cryptocurrency prices.

## ACKNOWLEDGMENT

The authors express their gratitude to the Research Management Center (RMC), Multimedia University, and the Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM), for their valuable support in this research.

## REFERENCES

[1] N. Kshetri, "Editorial: Economic, social and political impacts of blockchain," *Telecommunications Policy*, vol. 48, no. 3, Apr. 2024, Art. no. 102718, https://doi.org/10.1016/j.telpol.2024.102718.

[2] A. Hayes. "What Happens to Bitcoin After All 21 Million Are Mined?." Investopedia. https://www.investopedia.com/tech/what-happens-bitcoin-after-21-million-mined/.

[3] T. Antamis, A. Drosou, T. Vafeiadis, A. Nizamis, D. Ioannidis, and D. Tzovaras, "Interpretability of deep neural networks: A review of methods, classification and hardware," *Neurocomputing*, vol. 601, Oct. 2024, Art. no. 128204, https://doi.org/10.1016/j.neucom.2024.128204.

[4] M. H. Ahmadilivani, M. Taheri, J. Raik, M. Daneshtalab, and M. Jenihhin, "A Systematic Literature Review on Hardware Reliability Assessment Methods for Deep Neural Networks," *ACM Computing Surveys*, vol. 56, no. 6, Jan. 2024, Art. no. 141, https://doi.org/10.1145/3638242.

[5] C. Atheeq, R. Sultana, S. A. Sabahath, and M. A. K. Mohammed, "Advancing IoT Cybersecurity: Adaptive Threat Identification with Deep Learning in Cyber-Physical Systems," *Engineering, Technology & Applied Science Research*, vol. 14, no. 2, pp. 13559–13566, Apr. 2024, https://doi.org/10.48084/etasr.6969.

[6] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, https://doi.org/10.1016/j.neunet.2014.09.003.

[7] P. B. Weerakody, K. W. Wong, G. Wang, and W. Ela, "A review of irregular time series data handling with gated recurrent neural networks," *Neurocomputing*, vol. 441, pp. 161–178, June 2021, https://doi.org/10.1016/j.neucom.2021.02.046.

[8] M. J. Hamayel and A. Y. Owda, "A Novel Cryptocurrency Price Prediction Model Using GRU, LSTM and bi-LSTM Machine Learning Algorithms," *AI*, vol. 2, no. 4, pp. 477–496, Dec. 2021, https://doi.org/10.3390/ai2040030.

[9] A. Ho, R. Vatambeti, and S. K. Ravichandran, "Bitcoin Price Prediction Using Machine Learning and Artificial Neural Network Model," *Indian Journal of Science and Technology*, vol. 14, no. 27, pp. 2300–2308, Aug. 2021, https://doi.org/10.17485/IJST/v14i27.878.

[10] S. M. Raju and A. M. Tarif, "Real-Time Prediction of BITCOIN Price using Machine Learning Techniques and Public Sentiment Analysis." arXiv, June 18, 2020, https://doi.org/10.48550/arXiv.2006.14473.

[11] T. Awoke, M. Rout, L. Mohanty, and S. C. Satapathy, "Bitcoin Price Prediction and Analysis Using Deep Learning Models," in *Sixth International Conference on Information System Design and Intelligent Applications (INDIA 2019)*, Vizianagaram, Andhra Pradesh, India, 2019, pp. 631–640, https://doi.org/10.1007/978-981-15-5397-4_63.

[12] T. R. Noviandy *et al.*, "Deep Learning-Based Bitcoin Price Forecasting Using Neural Prophet | Ekonomikalia Journal of Economics," *Ekonomikalia Journal of Economics*, vol. 1, no. 1, pp. 19–25, July 2023, https://doi.org/10.60084/eje.v1i1.51.

[13] H. Kavitha, U. K. Sinha, and S. S. Jain, "Performance Evaluation of Machine Learning Algorithms for Bitcoin Price Prediction," in *2020 Fourth International Conference on Inventive Systems and Control*, Coimbatore, India, 2020, pp. 110–114, https://doi.org/10.1109/ICISC47916.2020.9171147.

[14] T. Phaladisailoed and T. Numnonda, "Machine Learning Models Comparison for Bitcoin Price Prediction," in *2018 10th International Conference on Information Technology and Electrical Engineering*, Bali, Indonesia, 2018, pp. 506–511, https://doi.org/10.1109/ICITEED.2018.8534911.

[15] S. McNally, J. Roche, and S. Caton, "Predicting the Price of Bitcoin Using Machine Learning," in *2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing*, Cambridge, UK, 2018, pp. 339–343, https://doi.org/10.1109/PDP2018.2018.00060.

[16] L. Felizardo, R. Oliveira, E. Del-Moral-Hernandez, and F. Cozman, "Comparative study of Bitcoin price prediction using WaveNets, Recurrent Neural Networks and other Machine Learning Methods," in *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing*, Beijing, China, 2019, pp. 1–6, https://doi.org/10.1109/BESC48373.2019.8963009.

[17] A. Aljadani, "DLCP2F: a DL-based cryptocurrency price prediction framework," *Discover Artificial Intelligence*, vol. 2, no. 1, Oct. 2022, Art. no. 20, https://doi.org/10.1007/s44163-022-00036-2.

[18] M. F. Rizkilloh and S. Widiyanesti, "Prediksi Harga Cryptocurrency Menggunakan Algoritma Long Short Term Memory (LSTM)," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 1, pp. 25–31, Feb. 2022, https://doi.org/10.29207/resti.v6i1.3630.

[19] A. Moghar and M. Hamiche, "Stock Market Prediction Using LSTM Recurrent Neural Network," *Procedia Computer Science*, vol. 170, pp. 1168–1173, Jan. 2020, https://doi.org/10.1016/j.procs.2020.03.049.

[20] V. Kumar T, S. Santhi, K. G. Shanthi, and G. M, "Cryptocurrency Price Prediction using LSTM and Recurrent Neural Networks," in *2023 2nd International Conference on Applied Artificial Intelligence and Computing*, Salem, India, 2023, pp. 1–5, https://doi.org/10.1109/ICAAIC56838.2023.10141048.

[21] P. Negi, R. Dhawad, N. C. Morris, R. Agrawal, and C. Dhule, "Cryptocurrency Price Analysis using Deep Learning," in *2023 International Conference on Sustainable Computing and Smart Systems*, Coimbatore, India, 2023, pp. 112–118, https://doi.org/10.1109/ICSCSS57650.2023.10169178.

[22] S. A. Ahmed, E. H. Khalifa, M. Nawaz, F. A. Abdalla, and A. F. A. Mahmoud, "Enhancing Cloud Data Center Security through Deep Learning: A Comparative Analysis of RNN, CNN, and LSTM Models for Anomaly and Intrusion Detection," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 20071–20076, Feb. 2025, https://doi.org/10.48084/etasr.9445.

[23] S. Sangameswaran. "Recurrent Neural Network (RNN)." MLearning.ai. https://sudarshan-s.medium.com/recurrent-neural-network-rnn-20a619190586.

[24] F. Aljuaydi, M. Zidan, and A. M. Elshewey, "A Deep Learning CNN-GRU-RNN Model for Sustainable Development Prediction in Al-Kharj City," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 20321–20327, Feb. 2025, https://doi.org/10.48084/etasr.9247.

[25] S. A. Marhon, C. J. F. Cameron, and S. C. Kremer, "Recurrent Neural Networks," in *Handbook on Neural Information Processing*, M. Bianchini, M. Maggini, and L. C. Jain, Eds. Berlin, Heidelberg, Germany: Springer, 2013, pp. 29–65, https://doi.org/10.1007/978-3-642-36657-4_2.

[26] Q. Wang, R.-Q. Peng, J.-Q. Wang, Z. Li, and H.-B. Qu, "NEWLSTM: An Optimized Long Short-Term Memory Language Model for Sequence Prediction," *IEEE Access*, vol. 8, pp. 65395–65401, 2020, https://doi.org/10.1109/ACCESS.2020.2985418.

[27] X. Zhu, P. Sobihani, and H. Guo, "Long Short-Term Memory Over Recursive Structures," in *Proceedings of the 32nd International Conference on Machine Learning*, Lille, France, 2015, pp. 1604–1612.

[28] N. Hettige and W. Basnayake, "Horizontal Divide and Conquer Methods for Non-Linear Time Series Forecasting Using LSTM-ANN; a Comparative Study Based on Simulated Data," *American Journal of Applied Mathematics and Statistics*, vol. 11, no. 2, pp. 63–69, Aug. 2023, https://doi.org/10.12691/ajams-11-2-4.

[29] C. Samal. "Time Series Tutorial using NeuralProphet." Analytics Vidhya. https://medium.com/analytics-vidhya/time-series-tutorial-using-neuralprophet-e918a1b437ed.

[30] O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, "NeuralProphet: Explainable Forecasting at Scale." arXiv, Nov. 29, 2021, https://doi.org/10.48550/arXiv.2111.15397.

[31] S. Robinson, C. Stedman, and E. McLaughlin, "data collection." Search CIO. https://www.techtarget.com/searchcio/definition/data-collection.

[32] "Bitcoin USD (BTC-USD) Price History & Historical Data." Yahoo Finance. https://finance.yahoo.com/quote/BTC-USD/history/.

[33] J. Brownlee. "What is a Confusion Matrix in Machine Learning." MachineLearningMastery.com. https://www.machinelearningmastery.com/confusion-matrix-machine-learning/.

## AUTHORS PROFILE

**Tan Khai Lian** holds a Bachelor of Computer Science (Hons.) in Data Science, awarded in 2024 by the Institute of Computer Science and Digital Innovation, UCSI University, Kuala Lumpur, Malaysia. He is a fresh graduate and currently employed at HONOR Technology (Malaysia) Sdn. Bhd., where he serves as a Data/IT Support Specialist. His areas of expertise include machine learning and data analysis, with practical experience in applying these techniques to real-world challenges.

**Ismail Ahmed Al-Qasem Al-Hadi** is an Assistant Professor at the Institute of Computer Science & Digital Innovation, UCSI University, Malaysia. He earned his Ph.D. in Intelligent Computing from Universiti Putra Malaysia (UPM) in 2017. His research interests include recommendation systems, machine learning, and deep learning approaches. Dr. Ismail has authored and co-authored various national and international publications in high-ranking journals as well as international conferences.

**Mohammad Ahmed Alomari** is affiliated with the Faculty of Technology and Electronic and Computer Engineering (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia, where he is currently working as a Senior Lecturer. He received his M.Sc. and PhD in Computer Systems Engineering from Universiti Putra Malaysia (UPM), Malaysia. This author became a Member (M) of the IEEE in 2010 and a Senior Member (SM) in 2020. Dr. Mohammad's major research interests include data storage encryption in mobiles, the security of smart renewable energy and smart grids, embedded systems, and cryptography. He is currently extending his research to the Internet of Things (IoT), blockchain technology, and cloud computing. Dr. Mohammad has authored and co-authored several national and international publications in journals and international conferences while working as a reviewer for worldwide reputable journals in Springer and the SAI organization. Dr. Mohammad is a member of organizing and technical committees of many international conferences and workshops in Malaysia and Japan. He is also a member of the International Association of Engineers (IAENG) and the Malaysian Society for Engineering & Technology (MySET).

**Mohammed Nasser Al-Andoli** received his B.Sc. degree in Computer Information Systems from Mutah University, Jordan, in 2011, his M.Sc. degree in Computer Science from the Jordan University of Science and Technology, Jordan, in 2016, and his Ph.D. degree in Information Technology from Multimedia University, Malaysia, in 2022. From 2022 to 2023, he was a Postdoctoral Researcher at the Faculty of Engineering and Technology, Multimedia University, Malaysia. In 2023, he was a Senior Lecturer at the Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka (UTeM), Malaysia. He is currently a Senior Lecturer at the Faculty of Computing and Informatics, Multimedia University, Malaysia. His research interests include malware analysis, complex network analysis, interactive media, machine learning, high-performance computing, deep learning, and parallel computing.

**Muhammed Basheer Jasser** (Member, IEEE) received the master's and Ph.D. degrees in software engineering from University Putra Malaysia (UPM). He is currently a Senior Lecturer and the Program Leader of B.Sc. degree (Hons.) in information technology with the Department of Computing and Information Systems, School of Engineering and Technology, Sunway University. He was granted the Malaysian Technical Cooperation Program Scholarship (MTCP) from the Ministry of Higher Education (Malaysia) for his postgraduate studies. He is also working on several fundamental and industrial research projects in the area of artificial intelligence and software engineering funded by several companies and universities. Several postgraduate students are working under his supervision on these projects. His major research interests include optimization algorithms, evolutionary computation, model-driven software engineering, formal specification, verification and theorem proving, artificial intelligence, and machine learning. He is also a member of several professional academic bodies, including IEEE, the Institute of Electronics, Information and Communication Engineers (IEICE), and Formal Methods Europe Organization.

**Abdulguddoos S. A. Gaid** obtained a B.Eng in Electronics Engineering from Sudan University for Science and Technology (SUST), Khartoum, Sudan, in 2000. He pursued an M.Eng and a PhD in Electrical, Electronic, and Systems Engineering from Universiti Kebangsaan Malaysia (UKM) in 2004 and 2010, respectively. After joining Taylor's University in September 2010, he served as a Lecturer until August 2012. From September 2012 onward, Dr. Gaid has been a Faculty Member at the Communication and Computer Department, the Faculty of Engineering, Taiz University. He teaches various courses on electrical, electronics, and communications. His primary research areas include wireless communications and microstrip patch antennas for wireless communication networks. Dr. Gaid has a publication record of over 30 conference and journal papers and has actively contributed to organizing committees for conferences such as the International Conference on Technology, Science, and Administration (ICTSA 2021) and the 3rd International Conference on Emerging Smart Technologies and Applications (eSmarTA 2023) held at Taiz University.