Computational Engineering and Technology Innovations

www.ceti.reapress.com

Comput. Eng. Technol. Innov. Vol. 2, No. 2 (2025) 62–80.

Paper Type: Original Article

BTC Illiquidity Prediction Using High-Dimensional Features and Hybrid CNNRNN

Ehsan Karimi Shahmarvandi[1], Reza Shoukhcheshm[2], Mohammadreza Gheisari Najafabadi[2], Sasan Mazaheri[3], Mohammad Ghadiri[4], Mahboobeh Shafiei[5], Ramin Mousa[6,*]

[1] Department of Computer Engineering, University of Amsterdam, Amsterdam, Netherlands.
[2] Department of Computer Engineering, Amir Kabir University, Tehran, Iran.
[3] Department of Industrial Management, Management and Accounting Faculty, Shahid Beheshti University, Tehran, Iran.
[4] Department of Computer Science, Lakehead University, Thunder Bay, Canada.
[5] Department of Computer Engineering, Shiraz University, Shiraz, Iran.
[6] Department of Computer Engineering, Zanjan University, Zanjan, Iran; raminmousa@znu.ac.ir.

Citation:

Received: 10 September 2024 *Revised: 25 November 2024* *Accepted: 27 February 2025*	Karimi Shahmarvandi, E., Shoukhcheshm, R., Gheisari Najafabadi, M., Mazaheri, S., Ghadiri, M., Shafiei, M., & Mousa, R. (2025). BTC illiquidity prediction using high-dimensional features and hybrid CNNRNN. *Computational engineering and technology innovations*, *2*(4), 62–80.

Abstract

The ease of converting an asset (such as stock or cryptocurrency) into cash or another asset without incurring a loss is referred to as liquidity, and the relationship between the time scale and the price scale of the investment represents this concept. The opposite of liquidity is illiquidity, which creates challenges in converting assets into cash in bearish markets. This paper examines the illiquidity of Bitcoin (BTC) and presents hybrid approaches based on deep learning. The hybrid model combines spatial features with CNN and temporal features with RNN. BTC hash rate information was collected in 3 different periods to evaluate the combined approaches. Six combined models and seven basic models were considered to evaluate the results. The combined model, based on CNN and Bi-IndRNN, achieved the best results in the MAE evaluation criterion for all three intervals. This approach achieves MAEs of 0.51, 1.18, and 4.97 for intervals I, II, and III, respectively. Additionally, the Pearson correlation analysis has shown that the two tasks of price prediction and illiquidity prediction are independent of each other, indicating that these tasks are completely separate.

Keywords: Illiquidity prediction, BTC hashrate, Hybrid model, CNNRNN, Cryptocurrency.

1 | Introduction

The Digital currency is a particular form of digital money created based on cryptography. Most digital currencies use blockchain to benefit from basic features such as decentralization, transparency, and immutability [1].

Corresponding Author: raminmousa@znu.ac.ir
https://doi.org/10.48314/ceti.v2i2.56

The decentralized nature of cryptocurrencies means that no single entity could exercise control over them. Cryptocurrencies can be sent over the Internet to another person directly without an intermediary. To carry out the transfer of digital currencies between themselves, people do not need to open a bank account or use the services of banks or any other intermediary organization [2]. They generate income with digital currencies that are created through various mechanisms. For some of these currencies, such as Bitcoin (BTC), mining is the process; for others, all coins have already been mined on the network [3].

Digital currencies rely on distributed ledger technology, one of whose key products is blockchain technology. Public blockchains, which most digital currencies employ, enable every single transaction to be viewed by both network members and outsiders [4].

Digital currencies enable users to send or receive money without disclosing their identities or visiting a bank [5]. Units of virtual currency are created through mining, wherein a vast amount of computer power is expended to perform calculations that solve incredibly complex problems, leading to coin production [6]. The currencies can be issued from exchanges and then used in encrypted wallets [7]. Cryptocurrencies and blockchain are still the babies from a financial perspective, and hence, so many applications will be developed in the future [5]. Transactions include bonds, stocks, and financial instruments exchangeable through this technology [8].

The major variation between digital money and conventional money is how they are encrypted and stored. Digital money is stored as electronic data in computers or electronic wallets, unlike bank money [3]. Cryptocurrencies are untraceable, and none of them is regulated by any bank, money-generating institution, or even state. Digital money is transferred between peer-to-peer business places; a peer-to-peer setup is a setup where two users conduct transactions without relying on any central organization.

Liquidity needs a solid and stable definition from an economic point of view; there have been numerous definitions that have considered this term at different levels and degrees.

A very reasonable definition is having the capacity to trade a vast amount of money at a transaction cost that is cheap with little impact on the market price. Authors assume liquidity as the facility of an asset to be converted into cash or any other asset without loss and exhibit the relationship between an investment's time and price scale [9]. Since liquidity is a prime market factor and may reflect the quality of market performance, financial investors have always paid significance to liquidity. A whole and normative digital currency market would be extremely liquid; market investors would able to swap a certain proportion of hares at ease and at high velocity. The market is also able to complete the matching of funds and appreciation of fixed asset value. Also, the more liquidity in the market, the more investors will be attracted, the higher investor confidence, and external shocks avoided. Thus, knowing its accurate measurement is important in estimating the market risk and ensuring the market is stable.

Currently, cryptocurrencies are the most valued assets to be traded, especially for international transactions. Cryptocurrencies have created a new bubble in the economy today, considering that they have recorded winning trends in the past. Thus, businesses and individuals have shown interest in investing in cryptocurrencies. However, the promise of making a profit is the most prominent consideration individuals make before making investments. Therefore, the investors are highly interested in observing the trend of the cryptocurrency market. BTC belongs to the initial group of cryptocurrencies that could be used in financial transactions. BTC uses a decentralized peer-to-peer network for archiving transactions in blockchain technology, and it is unstable in terms of time. It was about $0.5 in 2010, and it is currently around $28,000, with a maximum value of around $64,500 as of 04/14/2021. Thus, though BTC is used for investment, its price and illiquidity need to be predicted by traders. There are many endeavours to predict the price of BTC through machine learning, especially deep learning. However, few efforts have been made to predict its illiquidity. Accurate prediction is required to improve the security, stability and effectiveness of various technological elements on an international scale. However, even after years of study and monitoring of fluid data models, there isn't an instantaneous reaction in predicting future estimations completely. It may be seen from several types of research (literature) that aim to prove relevant information on data analysis and forecasting methods. Time series forecasting is among the usual topics of study that are highly significant for analyzing the patterns of existing and past data to predict future data. Online finance and BTC have gained increasing popularity and continue to shape the world's financial markets. Moreover, the virtual currency has attracted media interest. So many people have subscribed to this scheme. One of the most popular Google searches in the US and UK is "what is BTC?" Due to the number of users, cryptocurrencies are some of the most prohibited financial institutions in the world. Nonetheless, even though humans have the highest usage and involvement in cryptocurrencies, it is crucial to know the most appropriate methods of predicting as well as understanding the deep properties of cryptocurrencies. Therefore, literature has been presented in the follow-up of this research.

The literature on cryptocurrency market liquidity is even newer than the literature on its volatility, and it has various research:

I. Description of market structure: to which reference can be made to the study [10]. In the paper, the authors investigate investment in BTC by estimating daily trading behaviour and the transaction costs of the BTC-USD exchange rate. They found that implicit transaction costs are extremely low, and the investment size is smaller than in major global markets. Furthermore, the depth is sufficient for midterm trades. BTC has a distinct intraday trend with high trade levels throughout the day. Transaction volume is positively related to volatility and inversely related to capital appreciation. Overall, their results show that BTC is most suited for consumer-level trade.

II. The dynamics of volatility and liquidity [11], [12]: in [11], the liquidity of 456 different digital currencies was examined, showing that the predictability of digital currencies with high market liquidity returns was reduced. It was also shown that while BTC returns show evidence of efficiency, cryptocurrencies are autocorrelated and not independent. Their findings also illustrate a significant cross-sectional relation between panic intensity and liquidity. Therefore, they established the fact that liquidity is an important driver of market efficiency and return predictability of new digital currencies. Będowska-Sójka et al. [12] also checked whether digital currencies volatility and liquidity are related to each other or not. They considered 12 digital currencies having highly traded capital as their sample data. They considered daily and weekly liquidity. To investigate the dependence between digital currencies, they employed the causality approach. They used the asymmetric causality test to separate the effect of growth and volatility reduction from changes in liquidity and vice versa. In total, the empirical results show that high volatility is a Granger cause of high liquidity, meaning that high volatility attracts investors and leads to further interest in new financial instruments. The Granger causality test, a statistical hypothesis test for answering whether a time series is useful in predicting another, was first proposed in 1969. Regression tends to capture "pure" correlations. However, Clive Granger argued that economic causality could be tested by the degree to which it is possible to predict future values of one-time series based on past values of another time series.

III. Liquidity forecasting: None of the above research has attempted to forecast or explain market liquidity for digital money. Moreover, as a result of the nature of the microstructure of such liquidity, writers [13] assert that it is more suitable to use non-parametric models to forecast it. They learned that the K-Nearest Neighbour (KNN) algorithm is better suited for the prediction of cryptocurrency market liquidity than a linear model such as the Autoregressive Moving Average (ARMA). They have different units in this research, such as the Canadian dollar, British pound, Ethereum, Australian dollar, Euro, Japanese yen, Danish krone, Mexican peso, South African rand, Swedish krona, Norwegian krone, Swiss franc, New Zealand dollar, BTC, Taiwanese dollar, Brazilian real, Ripple, Singaporean dollar, South Korean won. They compared short-term market liquidity forecasts of significant cryptocurrencies and fiat currencies using traditional time series models such as ARMA and GARCH and a non-parametric machine learning model known as the KNN method. They found that the KNN algorithm worked better with market liquidity's nonlinearity and complexity. Its market microstructure prediction of cryptocurrency and fiat prices was better than using ARMA and GARCH models.

In this study, the three main goals are:

I. Collecting BTC data at different intervals.

II. Collection of different features and illiquidity labeling for different intervals.

III. Predicting the illiquidity of BTC in different intervals.

For this purpose, different approaches based on the RNNs has been proposed, where the RNN network is responsible for extracting temporal features.The basic steps of the proposed model include the following steps:

I. Collection of hash rate and historical BTC data.

II. Data pre-processing (dividing data into different intervals, applying different indicators, imputing missing values, removing outliers and labelling data).

III. Applying the RNN network to predicting illiquidity.

2| Data collection

This section will discuss data collection, preprocessing, feature selection, and the preparation of labels. Therefore, this section comprises the following subsections. An overview of the data collection is shown in *Fig. 1*.

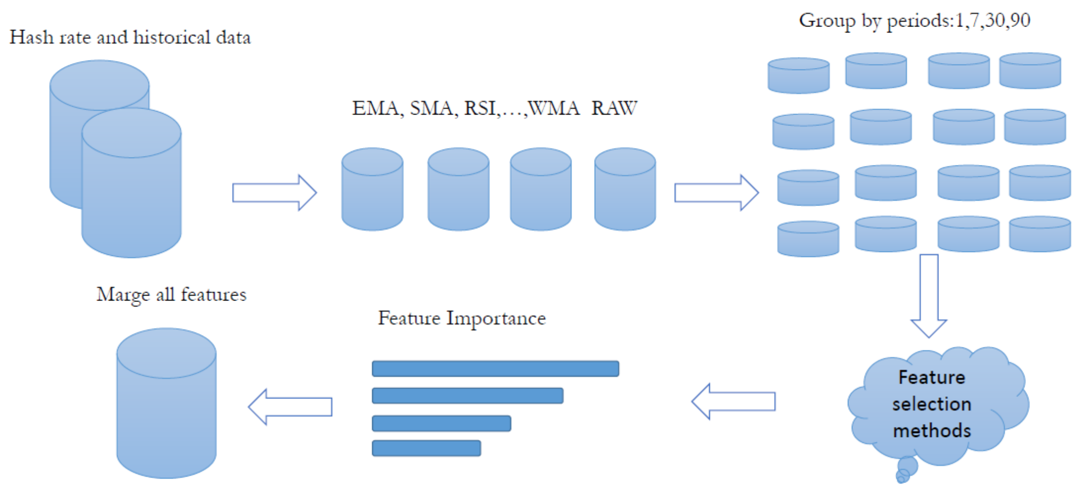

Fig. 1. Data collection and manipulation flowchart.

2.1| Hash Rate Data Collection

First, the BTC dataset, including the BTC hash rate, was collected. Features and prices of BTC data are downloadable for free online[1]. Data harvesting was performed similarly to [14]. The feature selection method picked a smaller group of features out of the enormous feature set. Technical indicators: Simple Moving Average (SMA), Exponential Moving Average (EMA), Relative Strength Index (RSI), Weighted Moving Average (WMA), Standard Deviation (STD), Variance (VAR), Triple Exponential Moving Average (TRIX) and Rate of Change (ROC) were used with these data. Subsequent to pre-processing conducted by Mudassir et al. [14], instances of missing values were estimated using the linear interpolation technique wherever possible. For all regression as well as classification models, data were shuffled and divided into two datasets, namely the training set and the validation set. Twenty percent of the data was set aside for validation, and 80% of the data was used for training. However, the Isolation Forest method, as proposed by Liu et al. [15], was applied to manage outliers. It removed approximately 14% of outlier values. Then, features applied for each interval are presented in *Table 1*.

[1] Data used in this research were collected from https://bitinfocharts.com

Table 1. Features selected for each interval.

Price Prediction Features			
III	II	I	**Intervals**
*	*		MDT fee 30: median transaction fee30trx
*		*	MDT fee 7:median transaction fee7trx
*			price 90 ema
*	*	*	size 90 trx
*		*	transactions
*		*	price 30 wma
*	*		price 3 wma
*	*	*	price 7 wma
*		*	median transaction fee 7 roc
	*	*	difficulty 30 rsi
*	*	*	mining profitability
*	*	*	price30sma USD
	*	*	sentinusd 90 ema
*	*	*	transaction value
*	*	*	top 100 cap
	*		difficulty 90 mom
*	*		hashrate 90 var
*	*	*	price 90 wma
*	*		sentinusd 90 sma
*	*	*	median transaction fee

2.2 | Illiquidity Label

In [16], they employed a new illiquidity measurement approach, on which the foundation for the summarization of data in this work relies. Considering w,d,h, and m are to be utilized for the weekly, daily, hourly and minute period, and all of them, the period is equal to w=[1,.,W] , d=[1,2,.,7], h=[00,01,.,23] and m=[00,01,.,59] and W is used for the weekly period in data collection. According to research [16], in order to measure volatility and volume patterns quantitatively, we define relative measures of volatility and volume in days, hours and minutes. We adopt the absolute return as a measure of volatility for this reason rather than the squared return. The reason is that absolute returns are less responsive than flat sections and are sufficient for measuring relative volatility. Absolute returns and strongly correlated amplitude-based measures are commonly used tools for estimating and measuring volatility. Absolute returns have the advantage of being able to be used, along with volume, to estimate some illiquidity measures below. Relative measures are more effective as mean variables are bounded between zero and one.

$$\lambda_{\sigma}^{\mathrm{day}}(\mathrm{d}) \equiv \frac{1}{\mathrm{N_d}}\sum_{\mathrm{w}} \frac{7\sum_{\mathrm{h,m}} |\mathrm{y}_\tau(\mathrm{w,d,h\ ,m})\ |}{\sum_{\mathrm{i=0}}^{6}\sum_{\mathrm{h,m}}|\mathrm{y}_\tau(\mathrm{w,d-i,h,m})|} \text{and } \lambda_{\mathrm{V}}^{\mathrm{day}}(\mathrm{d}) \equiv \frac{1}{\mathrm{N_d}}\sum_{\mathrm{w}} \frac{7\sum_{\mathrm{h,m}} \mathrm{V}_\tau(\mathrm{w,d,h,m})}{\sum_{\mathrm{i=0}}^{6}\sum_{\mathrm{h,m}}\mathrm{V}_\tau(\mathrm{w,d-i,h,m})}. \quad (1)$$

$$\lambda_{\sigma}^{\mathrm{hour}}(\mathrm{h}) \equiv \frac{1}{\mathrm{N_h}}\sum_{\mathrm{w,d}} \frac{24\sum_{\mathrm{m}} |\mathrm{y}_\tau(\mathrm{w,d,h\ ,m})\ |}{\sum_{\mathrm{j=0}}^{23}\sum_{\mathrm{m}}|\mathrm{y}_\tau(\mathrm{w,d,h-j,m})|} \text{and } \lambda_{\mathrm{V}}^{\mathrm{hour}}(\mathrm{h}) \equiv \frac{1}{\mathrm{N_h}}\sum_{\mathrm{w,d}} \frac{24\sum_{\mathrm{m}} \mathrm{V}_\tau(\mathrm{w,d,h,m})}{\sum_{\mathrm{j=0}}^{23}\sum_{\mathrm{m}}\mathrm{V}_\tau(\mathrm{w,d-i,h,m})}. \quad (2)$$

$$\lambda_{\sigma}^{\text{minute}}(m) \equiv \frac{1}{N_m}\sum_{w,d,h}\frac{60 * |y_\tau(w,d,h\,,m)|}{\sum_{k=0}^{59}|y_\tau(w,d,h-j,m)|} \text{and } \lambda_{V}^{\text{minute}}(h) \equiv \frac{1}{N_m}\sum_{w,d,h}\frac{60 * V_\tau(w,d,h\,,m)}{\sum_{k=0}^{59}V_\tau(w,d-i,h\,,m)}. \tag{3}$$

Their relative illiquidity measure was defined as follows:

$$\lambda_{\text{Illiquid}}^{\text{day}}(d) \equiv \frac{1}{N_h}\sum_{w,d}\frac{\text{Illiq}(w,d,h\,)}{\sum_{j=0}^{6}\text{Illiq}(w,d,h-j\,)} \text{ where Iliq}(w,d,h\,) \equiv \frac{\sum_m|y_\tau(w,d,h,m)|}{V_\tau(w,d,h)}. \tag{4}$$

The data in three main intervals are given in *Table 2*. it was gathered. In this table, the amount of training and test data is given.

Table 2. intervals of data and training and test size.

Dataset	Interval 1	Interval 2	Interval 3
Range	2013/04-2016/07	2013/04-2017/04	2013/04-2022/04
# Records	1206	1462	3285
# Train (80%)	964	1169	2628
# Test (20%)	242	293	657

3 | Proposed Method

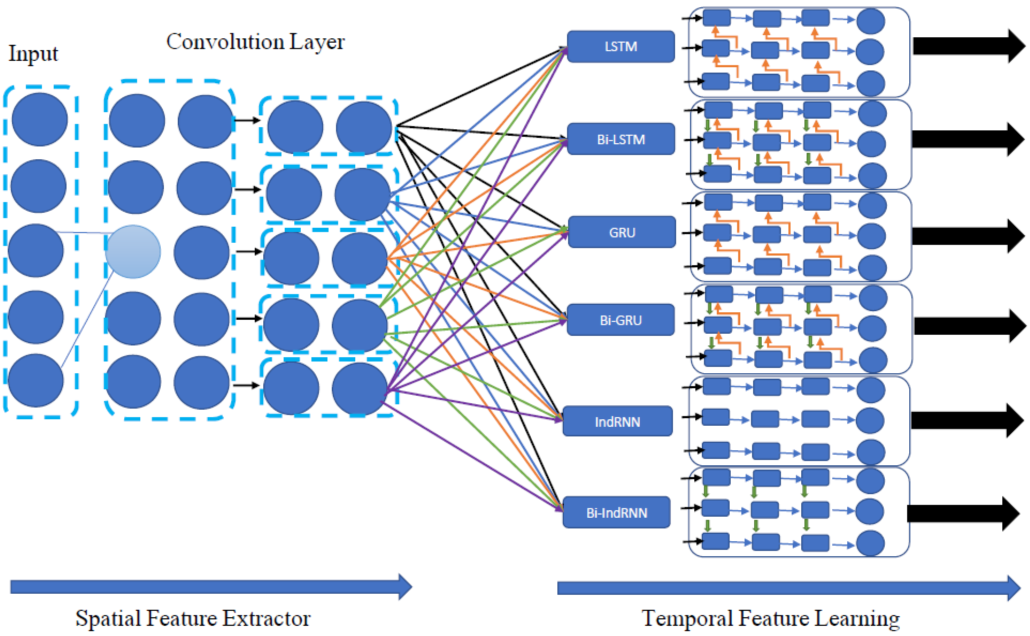

Fig. 2. An overview of proposed CNNRNN model.

Due to its advantages in combining the power of automatic feature extraction in CNN and the ability to capture long-term temporal dependencies in RNN, the CNN-RNN model has attracted the attention of many researchers in classi- fication and regression tasks. The convolution layer in the CNN-RNN separates the cross-correlation while preserving the deterministic and stochastic trends embedded among the input time series. It produces more accurate feature repre- sentations, enabling the RNN layers to detect the dependencies. Learn time more carefully. In addition to financial and textual applications, ensemble CNN-

RNN models enhanced with attention mechanisms have shown promise in hydrologic forecasting tasks, such as long-term flood prediction [17]. CNN- RNN networks for stock market prediction [18], named entity recognition [19], textual sentiment analysis [20], machine translation [21], facial expression recognition [22], and image description generation [23] have been used.

In this research, we propose a CNN-RNN-based architecture to predict non- criticism. The topology of the proposed CNN-RNN architecture is shown in *Fig. 2*. The main goal of this method is to predict illiquidity using the CNN- RNN network. We formulate the original model with the function, which extracts spatio-temporal features and predicts future illiquidity. Since a complex nonlinear mapping is expressed by accumulating multiple layers, we adopt a direct prediction strategy that avoids the accumulation of bias in the recursive strategy. The direct prediction strategy with the direct model h and its parameter Ø_h is formulated as follows:

$$\hat{y} = \emptyset_h\left(X^{\omega}_{\{t-h\}};\ \theta_h\right) + e_{t,h}, \tag{5}$$

where $[X^{\omega}_{\{t-h\}} = R^{t-h}, \ldots, R^{t-h-1}]$ with time delay ω, $\emptyset_h \ = \ [\Psi_h, \theta_h]$ are model

parameters for the forecast horizon h, where Ψ_h is a set of hyperparameters and θ_h is also a set of parameters. et, h is the prediction error of Ψ_h. The proposed model performs forecasts for a 1-day forecast horizon for different intervals. The input element R^t is a sequence extracted from the characteristics of the BTC hash rate, which is discussed in the data collection section. Convolutional neural networks use filters to extract spatial features in time series. *Fig. 2* shows the general architecture of CNN-RNN for predicting illiquidity. The proposed model for prediction includes two main steps.

I. First, the data preprocessing step defines the meta-parameters that train the CNN-RNN model. Min-max normalization for each feature and slid- ing window is one of the most common preprocessing methods before the hyperparameter definition.

II. Second, the modelling step adjusts the weights of the CNN-RNN model according to the defined meta-parameters. To extract spatial-temporal fea- tures, we develop a convolutional pooling operation that models hidden correlations between hash rate features and a gating operation applied to a recurrent memory cell that models temporal relationships from time series data.

The main configuration of the proposed approach consists of two main com- ponents, which are mentioned below:

I. CNN for extracting spatial features: the 1D-CNN shown in *Fig. 1* has three primary layers, including i) convolution layer, ii) pooling layer, and iii) output layer. The convolution layer is essential in DCNN, equipped with a kernel. The input data fed to the convolution layer are filtered to extract feature maps. A convolution kernel performs convolutions on the input data to learn multiple features. Convolution procedures create a powerful feature map in the convolution layer. By extracting local features on the input data, the number of network parameters and the complexity of the model are minimized. The convolution operation is as follows:

$$h(t) = (x * w)(t) = \sum_{\tau=-m}^{m} x(t-\tau)\omega(\tau), \tag{6}$$

where $x \in R^n$ is the input and $w \in R^{2m-1}$ is the kernel. The constructed feature map is then sent to the pooling layer (here Maxpooling was used), which is a subsection of the convolution layer. The pooling layer reduces the network parameters and dimensions by applying a downsampling tool. The following expression is used to calculate integration:

$$x_j^n = f\left(\beta_j^n \mathrm{Max}_{pooling} x_j^{n-1} b_j^n\right), \tag{7}$$

where x_j^n represents the output, x_j^{n-1} is the input of layer n, b_j^n is the bias, $\mathrm{Max}_{pooling}(.)$ is the down-sampling function, $f(.)$ is the activation function, and β represents the weights of the network. After pooling, the activator function is applied to the outputs. The activation function is a function that interacts with the neurons of the neural network and is responsible for converting inputs (output of the previous layer) into outputs. In the literature, various activation functions, including sigmoid, corrected linear unit (ReLU), and its derivatives are often used. The sigmoid and functions have significant gradient vanishing values. On the other hand, ReLU, a non-saturated activation function, partially eliminates the vanishing gradient and accelerates the convergence. ReLU maps negative values to 0 and positive values to the same positive values. This function works as follows:

$$\mathrm{ReLU} = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{if } x \leq 0. \end{cases} \quad \textbf{(8)}$$

After the pooling layer, the RNN network is located. This network can extract time series features from the convolution outputs. This relationship can be shown as follows:

$$y^k = \mathrm{RNN}\left(\omega^k x^{k-1} + b^k\right), \quad \textbf{(9)}$$

where ω^k is the weight matrix, bk and x^{k-1} is the output of ReLU application. In the proposed structure, three RNN networks were used in simple and bi- directional modes. In the following, these networks are discussed.

RNN for learning temporal features:

Initially, the input features in the form of vectors denoted by $x = [x^{<1>}, x^{<2>}, \ldots, x^{<n>}]$ are considered as inputs. This vector is already extracted by CNN in the first stage. This single-distributed vector should capture the meaning of the input. Having an input component s containing the attribute h, the output result of the encoding phase can be expressed as $\mathrm{enc}(s) = E$, where $E \subseteq R^h$ and the value of h represents a hyperparameter is shown. We used a recurrent network to encode the input layer. In this layer, the activated value in the hidden layer depends on the current input and output values of the previous step. In general, we will have the following:

$$h^{<k>} = g\left(r^{<k>}, h^{k-1}; \theta_{enc}\right), \quad \textbf{(10)}$$

Where g is the recurrent cell, $r^{<k>}$ the current input feature, $h^{<k>}$ the output of the hidden layer at time k, and h^{k-1} is the output of the hidden layer at time k-1 and θ _enc are the learnable parameters in the learning phase. Accordingly, the encoding phase is as follows:

$$\mathrm{enc}(S) = E = h^k = g\left(r^k, h^{k-1}; \theta_{enc}\right). \quad \textbf{(11)}$$

We used RNNs as a recurrent cell. Recurrent neural networks are feed-forward neural networks that are enhanced by including edges spanning adjacent time steps, adding a sense of time to the model. Edges connecting adjacent time steps, called recurrent edges, may form cycles, including cycles of length one that are self-connecting (from a node to it self over time). At time t, nodes with frequent edges receive input from the current data point $x(t)$ as well as from the hidden node values h^{t-1} in the previous state of the network. The output $\hat{y}^{(t)}$ at any time t is calculated according to the values of the hidden node $h(t)$ at time t. An input $x(t-1)$ at time $t - 1$ can produce an output $\hat{y}^{(t)}$ at time t and beyond through repeated connections. The value of $h(t)$ in each step is obtained through the following equation [24]:

$$h^t = \sigma(W^{hx} x^{(t)} + W^{hh} h^{(t-1)}) + b_h). \quad \textbf{(12)}$$

Also, the value of y^t is obtained through the following relationship:

$$\hat{y}^{(t)} = \mathrm{softmax}\left(W^{yh} h^{(t)} + b_y\right). \quad \textbf{(13)}$$

Here, W^{yh} is the matrix of standard weights between the input and the hidden layer, and W^{hh} is the matrix of recurrent weights between the hidden layer and itself in adjacent time steps. The vectors bh and by are bias parameters that allow each node to learn an offset.

The encoding layer is placed with RNN layers, which are discussed in the following three types of layers that are placed:

LSTM: the LSTM input in step *T* is the vector X ∈ RE. The hidden vector sequence is in LSTM, is calculated by the following equations:

$$
\begin{aligned}
f^{K} &= \sigma\left(W_f r^k + U_f h^{k-1} + b_f\right).\\
i^{k} &= \sigma\left(w_i r^k + U_i h^{k-1} + b_i\right).\\
g^{k} &= \tanh\left(w_g r^k + U_g h^{k-1} + b_g\right).\\
o^{k} &= \sigma\left(w_o r^k + U_o h^{k-1} + b_o\right).
\end{aligned}
\tag{14}
$$

Where i the input gate, o the output gate, f the forget gate, and g the update gate, $[W_i, W_f, W_g, W_o, b_i, b_f, b_g, b_o]$ is the set of parameters to be learned. q^k is updated through the following relationship:

$$q^k = f^k \odot q^{k-1} + i^k \odot g^k, \tag{15}$$

where the ⊙ symbol represents the element-wise product between two vec- tors. Finally, the activation of the cell is accomplished through the following relationship:

$$h^k = o^k \odot \tanh(q^k).$$

This network can also be used in bidirectional mode. The Bi-LSTM model is built on a simple LSTM network. In each cycle, the forward LSTM network (on the input order) and the inverse LSTM network (on the input inverse) are trained separately and connected to the same output layer. This model is formulated as follows [25]:

$$
\begin{aligned}
\overrightarrow{h_t} &= \overrightarrow{\mathrm{LSTM}}(h_{t-1}, \omega_t, c_{t-1})), \qquad t \in [1, T],\\
\overleftarrow{h_t} &= \overleftarrow{\mathrm{LSTM}}(h_{t+1}, \omega_t, c_{t+1})), \qquad t \in [T, 1],\\
h_t &= \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right].
\end{aligned}
\tag{16}
$$

Gated Recurrent Unit (GRU): The inputs of the GRU layer are vectors obtained through CNN layer and the output is calculated through the following equations.

$$
\begin{aligned}
z_t &= \sigma(w_z x_t + U_z h_{t-1}) + b_z.\\
r_t &= \sigma(w_r x_t + U_r h_{t-1}) + b_r.\\
h'_t &= (1 - z)h_{t-1} + z_t h'_t,
\end{aligned}
\tag{17}
$$

where z_t is update gate, r_t is rest gate, h_t is candidate gate, and h_t is output activation. $W_Z, W_R, W_h, U_Z, U_R, U_N$ are learnable matrixes, b_n, b_z, b_r are learnable biases, σ is sigmoid activation function, and is an element-wise multi-plication.

Bidirectional of this model can be defined as follows:

Right direction:

$$
\begin{aligned}
\vec{z}_t^{\,i} &= \delta\left(\overrightarrow{W}_i^{x} x_t^i + \vec{U}_i^{r} h_{t-1}^i\right).\\
\overrightarrow{r_t}^{i} &= \delta\left(\overrightarrow{W}_i^{r} x_t^i + \vec{U}_i^{r} h_{t-1}^i\right).\\
\overrightarrow{\widetilde{h_t}} &= \tanh\left(\overrightarrow{W}_i x_t + r_t^{o} \vec{U}_t h_{t-1}\right).
\end{aligned}
\tag{18}
$$

$$\vec{h}_t^i = z_t^i h_{t-1}^i + (1 - z_t^i) h_{t-1}^i.$$

Left direction:

$$\overleftarrow{z}_t^i = \delta(\overleftarrow{W}_i^z x_t^i + \overleftarrow{U}_t^i h_{t-1}^i).$$

$$\overleftarrow{r}_t^i = \delta(\overleftarrow{W}_i^r x_t^i + \overleftarrow{U}_i^r h_{t-1}^i).$$

$$\overleftarrow{\tilde{h}_t} = \tanh(\overleftarrow{W}_i x_t + r_t^o \overleftarrow{U}_t h_{t-1}). \tag{19}$$

$$\overleftarrow{h}_t^i = z_t^i h_{t-1}^i + (1 - z_t^i) h_{t-1}^i.$$

The final output from the two-mode combination is obtained as follows:

$$y_t = \mathrm{Softmax}(U[\vec{h}_t^i, \overleftarrow{h}_t^i]). \tag{20}$$

Independent Recurrent Neural Network (IndRNN): IndRNN was pro- posed in [26] as a main component of RNN, which is as follows:

$$h_t = \sigma\left(W x_t + u \odot h_{t-1} + b\right), \tag{21}$$

Where $x_t \in R^M$ and $h_t \in R^N$ are the input and hidden state in time step t, respectively, $W \in R^{N*M}$, $u \in R^N$ and $b \in R^N$ are the weight of current input, return input and bias of neurons. represents the Hadamard product (element product). σ is the essential activation function of neurons, which is Revised Linear Unit (ReLU) in this paper, and N is the number of neurons in the IndRNN layer. Bi-directional mode of this model can be defined as follows:

$$\vec{h}_t = \sigma(\vec{w} x_t + \vec{\mu} \odot \overrightarrow{h_{t-1}} + \vec{b}).$$

$$\overleftarrow{h}_t = \sigma(\vec{w} x_t + \overleftarrow{\mu} \odot \overleftarrow{h}_{t-1} + \overleftarrow{b}).$$

The final output from the two-mode combination is obtained as follows:

$$h_t = [\vec{h}_t, \overleftarrow{h}_t]. \tag{22}$$

4 | Experimental Result

First, in *Table 4*, the results of the proposed approaches and the proposed approaches in [14] article, which were used to predict the price of BTC, are given. These approaches are as follows:

I. ANN: the neural network that the authors considered includes the following hyperparameters.

II. Optimizer: adam was considered for this purpose (this value was also consid- ered in this research).

III. Hidden layers (neurons): two fully connected layers were considered, each layer having 128 neurons.

IV. Learning rate: the value of 0.08 was considered for this purpose.

V. Epoch: 5000 was considered, which was integrated with early stopping to avoid overfitting.

VI. Batch size: the value of 64 was considered for this purpose.

VII. Activation: set according to ReLU issues.

VIII. Loss function: logcosh is set.

This network is used in two modes of regression and classification in the basic article, and we used the regression mode to predict illiquidity.

I. Stacked Artificial Neural Network (SANN): stacked models are generally created from the combination of several models or the output of several models. In the basic article, 5 ANN networks were used with the settings mentioned in the ANN approach. Stacked Artificial Neural Network (SANN) consists of 5 separate ANN models that are used to train a larger ANN. Each model is trained in a separate layer. Since ANNs are stochastic, each trained model has different weights that enable them to learn their differences well. The final ANN uses the results of these models. This network is used in two modes of regression and classification in the basic article, and we used the regression mode to predict illiquidity.

II. Support Vector Machines (SVM): SVM is one of the most popular supervised learning algorithms used for classification as well as regression problems. The goal of SVM is to create the best decision line or boundary that can separate the n-dimensional space into classes so that we can easily assign a new data point to the correct category in the future. This border of the best decision is called hyperplane or margin. This algorithm selects the extreme points/vectors that help create the hyperplane. These extreme cases are called support vectors, and hence, the algorithm is called a SVM. For small data sets, SVM can provide predictions with a low error rate without requiring much training time. In the basic article, this approach is considered with the Gaussian RBF kernel, which was considered only in its regression mode (SVR) to predict the illiquidity of this approach.

III. Long short-term memory: this approach is an RNN network that was previously discussed in the proposed method section. This approach is used in both regression and classification modes, and in this research, its regression mode was used depending on the type of labels.

To evaluate the performance of regression models, the criteria Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) were used. A model with low MAE and MAPE is desirable.

Table 3. The results were obtained using the baseline approaches and the proposed approaches in two random and linear data-splitting policies.

Validation Method ↓	Metrics →	MAE ($)			MAPE (%)		
	Intervals →	I	II	III	I	II	III
Random Split (Run)	ANN	1.05	8.12	6.37	3.00	8.22	1.32
	SVM	1.23	5.37	9.47	0.96	2.09	2.21
	SANN	1.06	5.77	7.45	2.96	6.13	1.22
	LSTM	0.55	4.47	5.54	0.68	1.70	1.11
	Simple RNN	1.22	2.41	9.40	0.98	2.53	2.01
	GRU	1.45	3.65	6.66	0.69	1.91	1.12
	IndRNN	0.52	1.21	5.01	0.65	1.83	1.12
	CNN+LSTM	0.65	1.32	4.02	0.62	1.98	1.19
	CNN+BI-LSTM	0.51	1.18	5.00	0.62	1.76	1.17
	CNN+GRU	0.57	1.31	5.39	0.87	1.94	1.25
	CNN+BI-GRU	0.86	1.56	6.73	0.99	2.04	2.85
	CNN+ IndRNN	0.51	1.19	4.98	0.61	1.77	1.08
	CNN+ Bi-IndRNN	0.51	1.18	4.97	0.61	1.76	1.10

Table 3. Continued.

Validation Method ↓	Metrics →	MAE ($)			MAPE (%)		
	Intervals →	I	II	III	I	II	III
Linear Split (Run)	ANN	8.21	9.4	8.50	5.70	22.2	3.21
	SVM	2.04	6.19	9.87	0.87	14.49	7.51
	SANN	2.75	12.7	12.13	3.89	9.58	2.10
	LSTM	2.83	5.02	14.77	3.75	1.18	2.51
	Simple RNN	2.05	6.12	6.10	0.90	1.51	7.92
	GRU	2.40	5.67	4.87	3.80	1.90	3.01
	IndRNN	2.01	4.80	3.89	3.70	1.15	2.42
	CNN+LSTM	2.81	4.97	14.65	3.73	1.19	2.51
	CNN+BI-LSTM	2.82	5.05	14.65	3.74	1.17	2.51
	CNN+GRU	2.39	5.67	4.46	3.79	1.87	2.91
	CNN+BI-GRU	2.37	5.45	4.47	3.78	1.86	2.94
	CNN+ IndRNN	2.00	4.81	3.91	3.69	1.16	2.40
	CNN+ Bi-IndRNN	2.00	4.83	3.92	3.69	1.17	2.43

The Keras[1] was used to implement the ANN, SANN, LSTM, and proposed approaches. for SVM a implementation the SKlearn[2] library was used.

The results of regression models for three intervals are given in Table 4. In the first interval, from April 2013 to July 2016, BTC prices did not experience much volatility; hence, illiquidity was also low. In this interval, all the models have performed the predictions well. In Random Split (Paper) mode, the ANN model achieved MAE=1.05 and MAPE=3.00 in interval I. This model reached MAE=8.12 and 6.37 in intervals II and III, respectively. In interval I, the SVM model provided weaker results than the ANN approach regarding MAE. This approach reached MAE=1.23 and MAPE=0.96. The SVM approach in interval III also obtained weaker results than ANN. However, in interval II, the SVM approach achieved MAE=5.37 and MAPE=2.09.

The SANN approach, a combination of ANNs, achieved MAE=1.06 and MAPE=2.96 in the interval I, which has performed better than ANN and SVM approaches

in terms of MAE. In interval II, this approach achieved MAE=5.77 and MAPE=6.13. The SANN approach reached MAE=7.45 in Interval III, which provided weaker results than the ANN approach. The LSTM approach ob- tained more acceptable results than ANN, SVM and SANN. This approach reached MAE=0.55 and MAPE=0.68 in the interval I. This approach reached MAE=4.47 and 5.54 in intervals II and III, respectively. Also, this approach reached MAPE=1.70 and 1.11 in intervals II and III, respectively. The simple RNN model was able to reach MAE=1.22 and MAPE=0.98. This approach reached MAE=2.41 and MAPE=2.53 in interval II. Interval III presented weaker results and reached MAE=9.40 and MAPE=2.01. The GRU model obtained the worst results among the comparative and proposed models in interval I. This approach reached MAE=1.45 and MAPE=0.69 for inter- val I. Also, this approach obtained values of MAE=3.65 and RMSE=1.91 for interval II. This approach provided weaker results in interval III and reached MAE=6.66 and RMSE=1.12. The IndRNN model was able to reach MAE=0.52 and MAPE=0.65 in interval I. Also, this approach recorded 1.21 and 5.01 values for MAE and 5.01 and 1.12 values for MAPE in intervals II and III. The proposed combined approaches provided results comparable to other approaches. This approach, combined with LSTM, obtained MAE equal to 0.65, 1.32, and 4.02 for intervals I, II, and III, respectively. Also, this approach obtained MAPE equal to 0.62, 1.98, and 1.19 for intervals I, II, and III, respectively. This approach, combined with Bi-LSTM, reached MAE=0.51 and MAPE=0.62 in interval I. Also, the CNN+BI-LSTM model reached MAE=1.18 and MAE=5.00 in intervals II and II, respectively, and

[1] https://keras.io/

[2] https://scikit-learn.org/

this value was equal to 1.76 and 1.17 in the MAPE criterion, respectively. In the interval I, the proposed approach, in combination with GRU and Bi-GRU, obtained MAE=0.57 and 0.86 and MAPE=0.87 and 1.94, respectively. This approach also obtained MAE=1.31 and 1.56 and MAPE=1.94 and 2.04 in In- terval II combined with GRU and Bi-GRU. In interval III, this model provided weaker results. Combined with GRU and Bi-GRU approaches, this approach reached MAE=5.39 and 6.73 and MAPE=1.25 and 2.85, respectively. In the interval I, the CNN+IndRNN and CNN+Bi-IndRNN approaches could reach MAE=1.51 and MAPE=0.61, respectively. These two approaches also ob- tained close results regarding MAE and MAPE in Interval II. In interval II, the CNN+ IndRNN approach reached MAE=1.19 and MAPE=1.77, and the CNN+ Bi-IndRNN approach reached MAE=4.97 and MAPE=1.10.

Table 3 also shows the base line results and proposed approaches in the Linear Split (Run) policy. ANN model in Linear Split could reach MAE=8.21 and MAPE=5.70 in interval I. The SVM model has obtained more acceptable results in interval I than ANN. This approach was able to reach MAE=2.04 and MAPE=0.87. This approach reached MAE=6.19 and MAPE=14.49 in interval II. In interval III, the SVM approach's results were weaker than the ANN approach. The SVM approach reached MAE=9.87 and RMSE=7.51 in this interval. Compared to SVM, the SANN approach obtained weaker results; compared to ANN, it obtained better results in interval I. This approach was able to achieve MAE=2.75 and MAPE=3.89 in Interval I. Also, in interval II, this approach reached MAE=12.7 and MAPE=14.49. In interval III, the re- sults of this approach in the MAE criterion were worse than those of SVM and ANN. This approach reached MAE=12.13 and MAPE=2.10 in Interval III. The LSTM approach in the interval I left poor results; this approach reached MAE=2.83 and MAPE=3.75, which was better than the ANN approaches and worse than the SVM and SANN approaches. The best results obtained by this approach were in interval II, which reached MAE=5.2 and MAPE=1.18. The Simple RNN approach obtained better results in interval I compared to the reviewed approaches. This approach reached MAE=2.05 and MAPE=0.90 in the interval I; compared to LSTM, it has improved about 0.78 in MAE. In interval II, this approach provided weaker results than LSTM in MAE. Also, compared to LSTM, this approach reached MAE=6.10 in interval III, which has improved by about 8.67. The GRU approach obtained the lowest MAE in Interval I and the worst MAE in Interval II. This approach obtained MAE=2.40 and MAE=5.67 for intervals I and II, respectively. IndRNN- based approaches obtain the lowest MAE for intervals I, II, and III. The IndRNN approach was obtained for Interval I (MAE=2.01 and MAPE=3.07), Interval II (MAE=4.08 and MAPE=1.15), and Interval III (MAE=3.89 and MAPE=2.42). The CNN+ IndRNN approach obtained Interval I (MAE=2.00 and MAPE=3.69), Interval II (MAE=4.81 and MAPE=1.16), and Interval III (MAE=3.91 and MAPE=2.40). Also, the CNN+ Bi-IndRNN approach was able to obtain Interval I (MAE=2.00 and MAPE=3.69), Interval II (MAE=4.83 and MAPE=1.17), and Interval III (MAE=3.92 and MAPE=2.43). The com- bined approach of CNN and LSTM in two modes, Bi-directional and simple, obtained almost equal results in two intervals, I and II. In the CNN+GRU and CNN+Bi-GRU approaches, these results were almost close to each other. The bar chart of the comparative approaches and the proposed approaches are given in *Figs. 3-6*. The proposed CNN+ Bi-IndRNN approach has reached the lowest MAE values in all three intervals.

Fig. 5 shows the confusion matrix for different models in price prediction and illiquidity prediction tasks. Correlation is used to quantify the relationship between variables. The purpose of presenting confusion matrices is to show the relationship between the two tasks of price prediction and illiquidity prediction. First, *Fig. 7* shows the relationship between forecasting ap- proaches. According to this matrix, it can be concluded that price prediction and illiquidity prediction models have a negative correlation. This correlation is shown with lower intensity in *Fig. 8*. It can be concluded that these two tasks are entirely different from each other, and there is no connection between them.

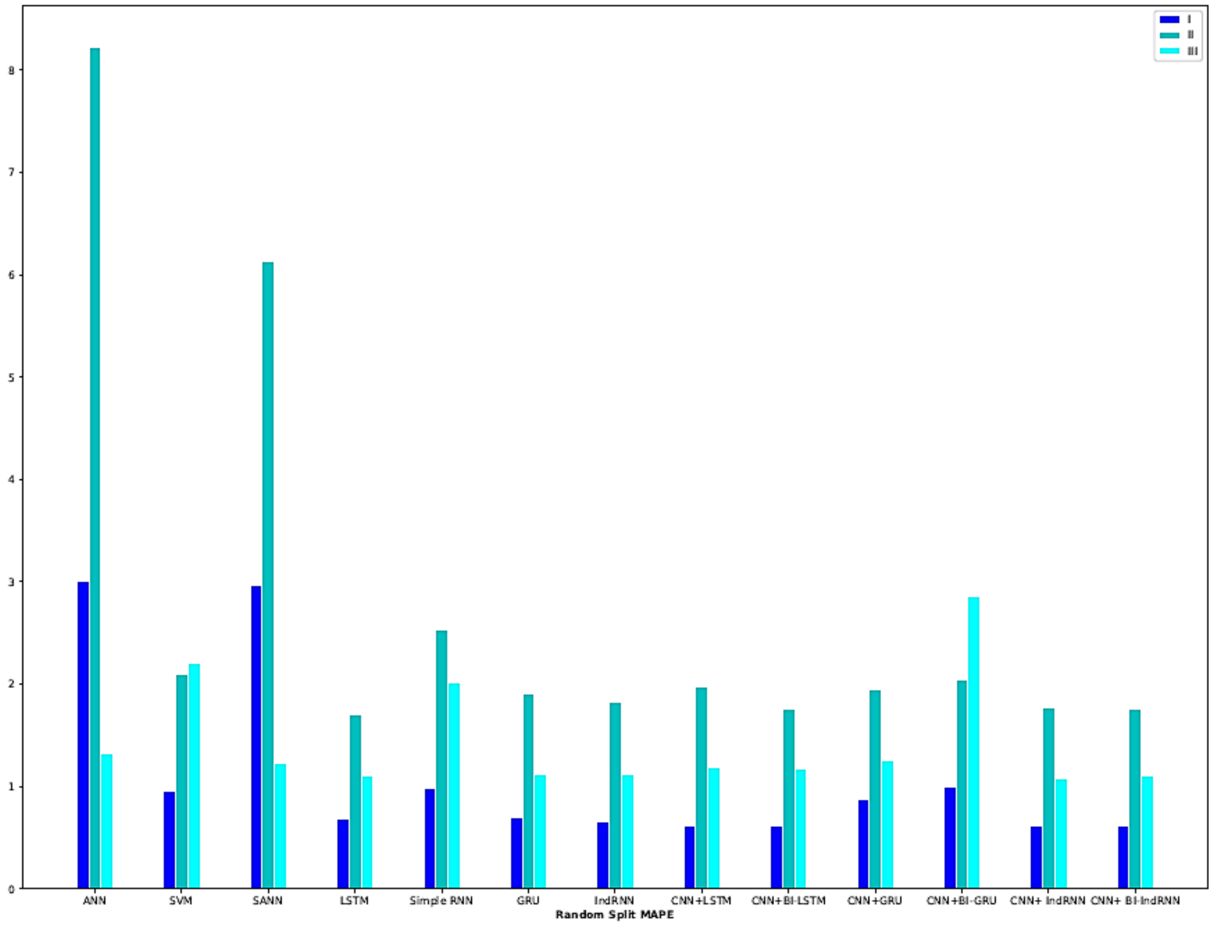

Fig. 1. Random MAPE.

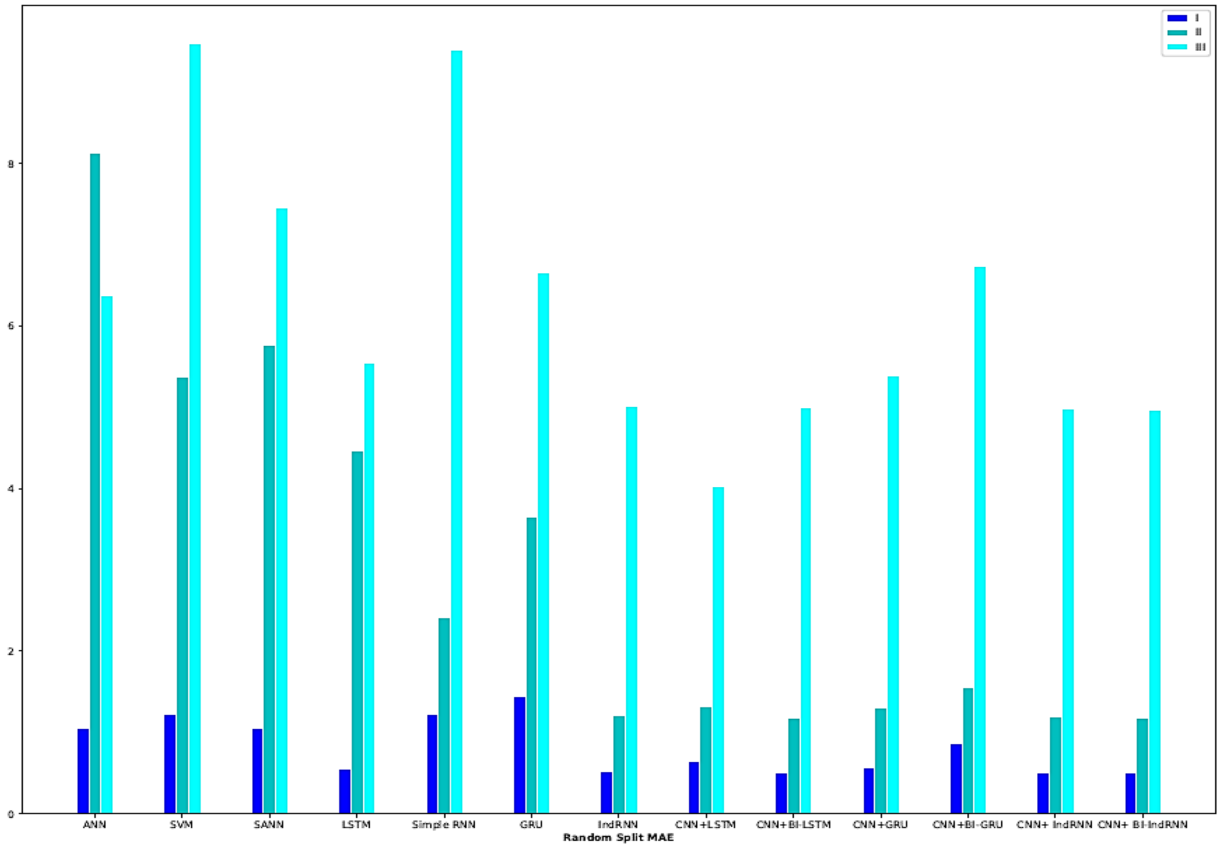

Fig. 4. Random MAE.

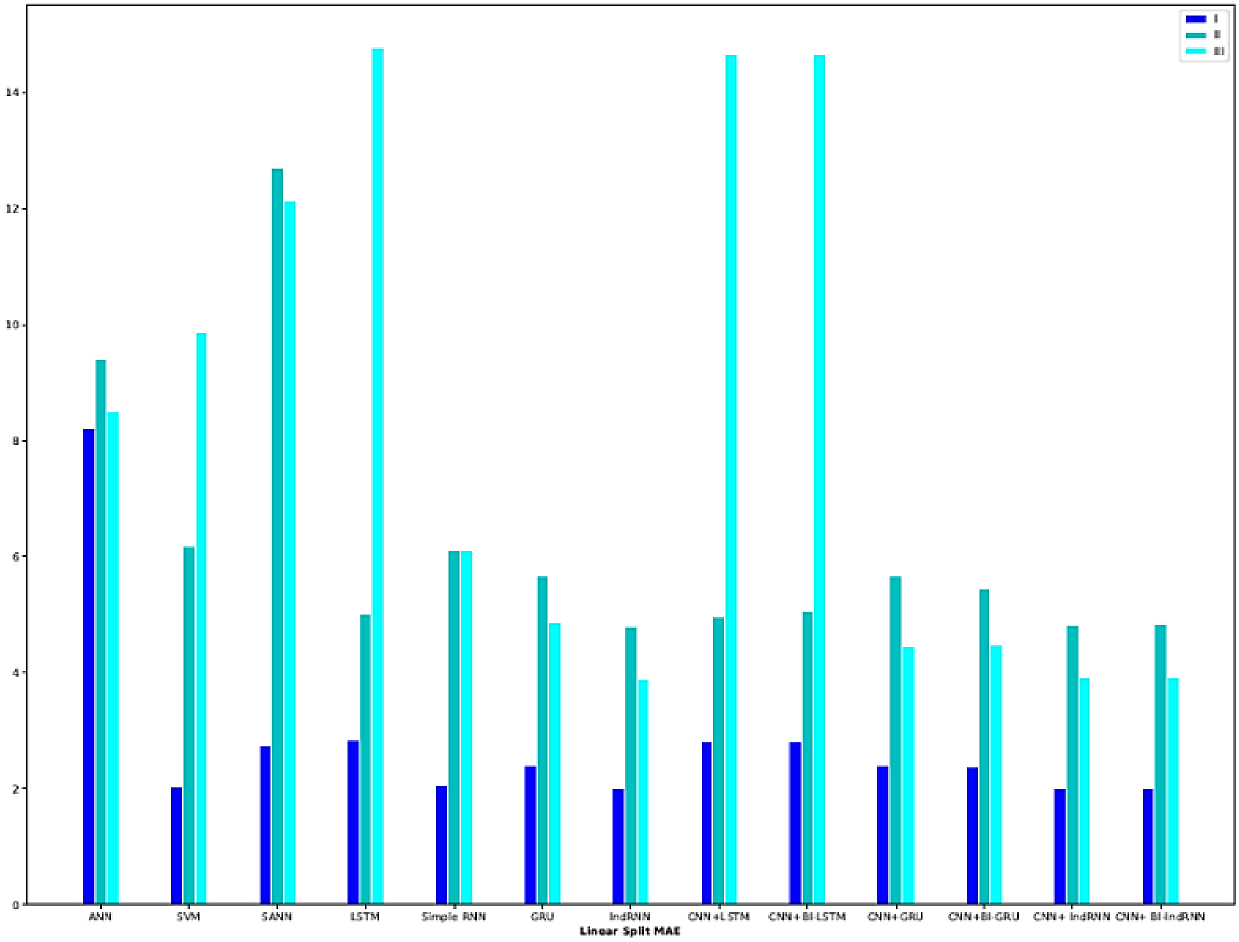

Fig. 2. Linrear split MAPE.

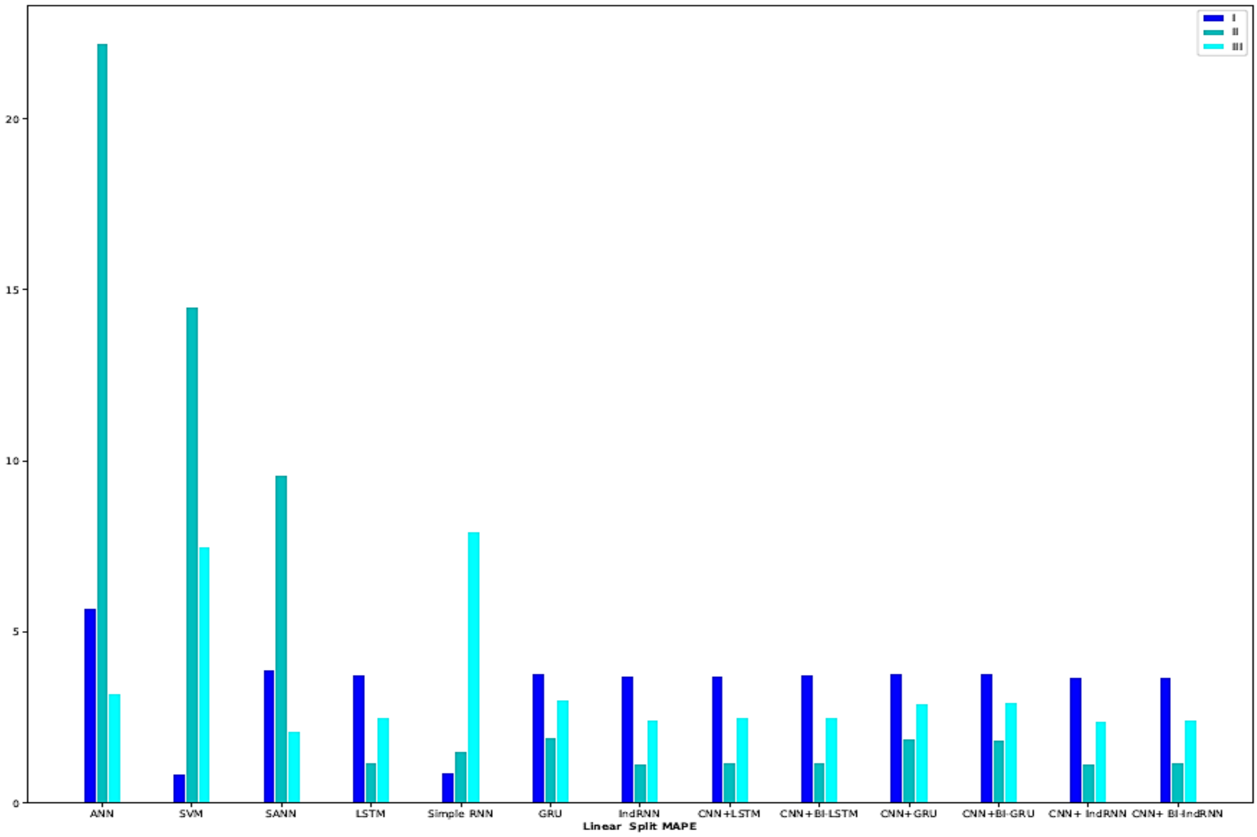

Fig. 3. Linrear split MAE.

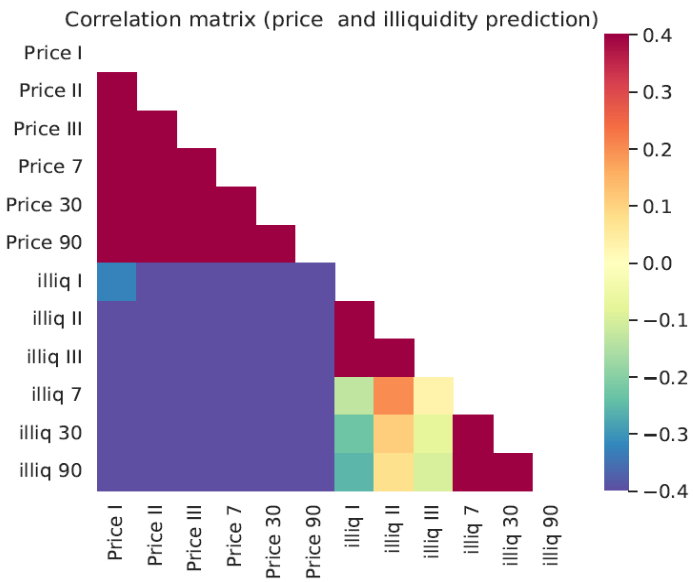

Fig. 4. Correlation Model between price prediction and illiquidity predication.

Warm colors (red) indicate a positive correlation, cool colors (blue) indicate a negative correlation (random split).

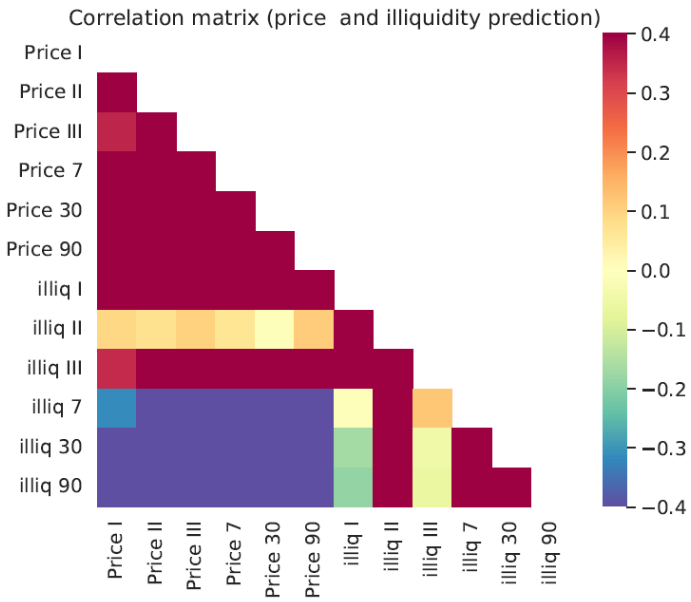

Fig. 5. Correlation model between price prediction and illiquidity predication.

Warm colors (red) indicate a positive correlation, cool colors (blue) indicate a negative correlation (linear split).

5 | Conclusion

This research aimed to predict short-term and medium-term BTC market illiquidity using time series models and machine learning algorithms. We found that the IndRNN algorithm is a better predictor than other approaches due to the nonlinearity of market liquidity and the complexity of its market mi- crostructure. Considering a time series analysis, the IndRNN model performs better for capturing short-term liquidity, while the LSTM and GRU models also provide acceptable results. Despite the above results, limitations in this study can be seen. First, the sample used in this study is small; the data history used in this research is considered in the daily interval, which reduces the number of training samples. Secondly, base crypto currencies and altcoins have much less history than BTC, which makes it challenging to present a model for them. For future research, other

GARCH-type models can be investigated in comparative analysis, such as the VAR-BEKK-GARCH model proposed by Yu et al. [27] for time series of log spreads. The challenge ahead in this issue is that some intervals rarely occur. Each of these intervals can be mapped to classes, and depending on the minority classes, minority class control methods can be applied.

Acknowledgments

The authors would like to express their sincere gratitude to the researchers and institutions that made the cryptocurrency datasets and computational resources available for this study. The authors also appreciate the valuable comments and suggestions provided by anonymous reviewers, which contributed to improving the quality of this manuscript.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request. All data generated or analyzed during this study are included in the published article. Publicly available Bitcoin-related datasets used in this research can also be obtained from their respective online repositories.

Authors' Contributions

E. K. S.: Conceptualization, Methodology, Software, Formal Analysis, Data Curation, Writing—Original Draft.

R. S.: Methodology, Software, Validation, Investigation.

M. G. N.: Data Curation, Formal Analysis, Visualization.

S. M.: Conceptualization, Investigation, Writing—Review & Editing.

M. G.: Validation, Supervision, Methodology.

M. S.: Visualization, Investigation, Writing—Review & Editing.

R. M.: Supervision, Project Administration, Funding Acquisition (if applicable), Writing—Review & Editing.

All authors have read and approved the final version of the manuscript.

Funding

This research received no external funding.

Consent for Publication

All authors have reviewed and approved the final version of the manuscript and consent to its publication.

Ethics Approval and Consent to Participate

This study did not involve human participants or animals. The research was conducted using publicly available cryptocurrency datasets and complied with all relevant ethical standards for computational and data-driven research. Therefore, formal ethical approval and informed consent were not required.

Referenses

[1] Huberman, G., Leshno, J. D., & Moallemi, C. (2019). An economist's perspective on the BTC payment system. *AEA papers and proceedings*, *109*, 93–96. https://doi.org/10.1257/pandp.20191019

[2] Sensoy, A. (2019). The inefficiency of BTC revisited: A high-frequency analysis with alternative currencies. *Finance research letters*, *28*, 68–73. https://doi.org/10.1016/j.frl.2018.04.002

[3] Saito, T. (2015). A microeconomic analysis of BTC and illegal activities. In *Handbook of digital currency* (pp. 231–248). San Diego: Academic Press. https://doi.org/10.1016/B978-0-12-802117-0.00012-6

[4] Huang, J., Mensi, M., Oveisi, E., Mantella, V., & Buonsanti, R. (2019). Structural sensitivities in bimetallic catalysts for electrochemical CO_2 reduction revealed by Ag–Cu nanodimers. *Journal of the American chemical society*. https://pubs.acs.org/doi/abs/10.1021/jacs.8b12381

[5] Kajtazi, A., & Moro, A. (2019). The role of BTC in well diversified portfolios: A comparative global study. *International review of financial analysis*, *61*, 143–157. https://doi.org/10.1016/j.irfa.2018.10.003

[6] Conti, M., Kumar, E. S., Lal, C., & Ruj, S. (2018). A survey on security and privacy issues of BTC. *IEEE communications surveys & tutorials*, *20*(4), 3416–3452. https://doi.org/10.1109/COMST.2018.2842460

[7] Lee, J. H., & Pilkington, M. (2017). How the blockchain revolution will reshape the consumer electronics industry [Future directions]. *IEEE consumer electronics magazine*, *6*(3), 19–23. https://doi.org/10.1109/MCE.2017.2684916

[8] Baur, D. G., Hong, K., & Lee, A. D. (2018). BTC: Medium of exchange or speculative assets? *Journal of international financial markets, institutions and money*, *54*, 177–189. https://doi.org/10.1016/j.intfin.2017.12.004

[9] Chen, B., Zeng, X., Liu, Y., Xiao, F., Huang, M., Bing Tan, K., … & Zhan, G. (2022). Thermal decomposition kinetics of M–BTC (M = Cu, Co, Zn, and Ce) and M–BTC/Pt composites under oxidative and reductive environments. *Chemical engineering journal*, *450*, 138470. https://doi.org/10.1016/j.cej.2022.138470

[10] Dyhrberg, A. H., Foley, S., & Svec, J. (2018). How investible is BTC? Analyzing the liquidity and transaction costs of BTC markets. *Economics letters*, *171*, 140–143. https://doi.org/10.1016/j.econlet.2018.07.032

[11] Wang, B., Long, W., & Wei, X. (2018). Investor attention, market liquidity and stock return: A new perspective. *Asian economic and financial review*, *8*(3), 341. https://doi.org/10.18488/journal.aefr.2018.83.341.352

[12] Będowska-Sójka, B., Hinc, T., & Kliber, A. (2020). Volatility and liquidity in cryptocurrency markets—the causality approach. *Contemporary trends and challenges in finance* (pp. 31–43). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-43078-8_3

[13] Cortez, K., Rodríguez-García, M. D., & Mongrut, S. (2021). Exchange market liquidity prediction with the k-nearest neighbor approach: Crypto vs. fiat currencies. *Mathematics*, *9*(1). https://doi.org/10.3390/math9010056

[14] Mudassir, M., Bennbaia, S., Unal, D., & Hammoudeh, M. (2025). Time-series forecasting of BTC prices using high-dimensional features: A machine learning approach. *Neural computing and applications*, *37*(28), 22979–22993. https://doi.org/10.1007/s00521-020-05129-6

[15] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2012). Isolation-based anomaly detection. *ACM transactions on knowledge discovery*, *6*(1). https://doi.org/10.1145/2133360.2133363

[16] Hansen, P. R., Kim, C., & Kimbrough, W. (2024). Periodicity in cryptocurrency volatility and liquidity*. *Journal of financial econometrics*, *22*(1), 224–251. https://doi.org/10.1093/jjfinec/nbac034

[17] Marjan, K., Reza, N. M., Mahmood, F., Iman, A., Rouzbeh, N., & H., G. A. (2024). Improving long-term flood forecasting accuracy using ensemble deep learning models and an attention mechanism. *Journal of hydrologic engineering*, *29*(6), 4024042. https://doi.org/10.1061/JHYEFF.HEENG-6262

[18] You, Y. (2025). Research on precision optimization of deep learning method based on vat-lstm and transformer encoder in financial data prediction. *2025 international conference on intelligent systems and computational networks (ICISCN)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICISCN64258.2025.10934410

[19] Sharma, A., & Kansal, V. (2024). Comprehensive evaluation of cnn, rnn, and crnn for document retrieval in healthcare systems. *2024 5th international conference on recent trends in computer science and technology (ICRTCST)* (pp. 491–496). IEEE. https://doi.org/10.1109/ICRTCST61793.2024.10578433

[20] Kumar, M., Khan, L., & Chang, H. T. (2025). Evolving techniques in sentiment analysis: A comprehensive review. *PeerJ computer science*, *11*, e2592. https://doi.org/10.7717/peerj-cs.2592

[21] Biswas, D., & Gil, J. M. (2024). Design and implementation for research paper classification based on CNN and RNN models. *Journal of internet technology*, *25*(4), 637–645. https://jit.ndhu.edu.tw/article/view/3081/0

[22] Manalu, H. V., & Rifai, A. P. (2024). Detection of human emotions through facial expressions using hybrid convolutional neural network-recurrent neural network algorithm. *Intelligent systems with applications*, *21*, 200339. https://doi.org/10.1016/j.iswa.2024.200339

[23] Yoga, M., Ramyasri, M. M., Bhavatharini, N., Harini, M., & NivethaSri, A. (2024). Unravelling visual narratives with deep learning for image caption generation. *2024 international conference on cognitive robotics and intelligent systems (ICC-ROBINS)* (pp. 387-393). IEEE. https://doi.org/10.1109/ICC-ROBINS60238.2024.10533953

[24] Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *ArXiv preprint arxiv:1506.00019*, *abs/1506.0*. https://doi.org/10.48550/arXiv.1506.00019

[25] Li, X., Zhang, W., Ding, Y., Cai, J., & Yan, X. (2025). Rolling bearing degradation stage division and RUL prediction based on recursive exponential slow feature analysis and Bi-LSTM model. *Reliability engineering & system safety*, *259*, 110923. https://doi.org/10.1016/j.ress.2025.110923

[26] Li, S., Li, W., Cook, C., Zhu, C., & Gao, Y. (2018). Independently recurrent neural network (INDRNN): building a longer and deeper {RNN}. *2018 conference on computer vision and pattern recognition, {CVPR} 2018, Salt Lake City, UT, USA* (pp. 5457–5466). IEEE. 10.1109/CVPR.2018.00572

[27] Yu, L., Zha, R., Stafylas, D., He, K., & Liu, J. (2020). Dependences and volatility spillovers between the oil and stock markets: New evidence from the copula and VAR-BEKK-GARCH models. *International review of financial analysis*, *68*, 101280. https://doi.org/10.1016/j.irfa.2018.11.007