Received: 15 January 2023 | Revised: 22 June 2023 | Accepted: 5 August 2023

DOI: 10.1111/exsy.13428

ORIGINAL ARTICLE

Expert Systems WILEY

Investigating the effectiveness of Twitter sentiment in cryptocurrency close price prediction by using deep learning

Bahareh Amirshahi | **Salim Lahmiri**

Department of Supply Chain and Business Technology Management, John Molson School of Business, Concordia University, Montreal, Quebec, Canada

Correspondence
Salim Lahmiri, Department of Supply Chain and Business Technology Management, John Molson School of Business, Concordia University, Montreal, QC, Canada.
Email: salim.lahmiri@concordia.ca

Abstract

In recent years, cryptocurrencies' price prediction has attracted the interest of many people including investors, researchers and practitioners. In this study, we proposed a hybrid model for predicting the daily close price of cryptocurrencies based on different neural networks such as long short-term memory, convolutional neural network and attention mechanism. Using an ensemble of three pre-trained language models, we extracted sentiment of cryptocurrency-related tweets posted between 1 January 2021 and 31 December 2021. We constructed 20 different versions of our model and evaluated their performance on data of 27 most traded cryptocurrencies using a history of previous days' sentiment data along with close prices as input data. The flexible input layer of our model enables different ways of feeding data into the model to adjust it for different cryptocurrencies to obtain better predictions. Our analysis revealed several important findings. We showed that longer sequences of input data achieve most accurate predictions on average. More specifically, using a history of 14- and 21-days' data results in lowest RMSE values on average compared to using a history of 7 days. However, there is no significant difference between the results related to the input sequences with lengths of 14 and 21. In addition, our findings suggest that sentiment data can be useful in predicting prices for more than 70% of the studied cryptocurrencies. Thus, peoples' emotions, opinions, and sentiment that are expressed through their posts on Twitter platform play a significant role in prediction of cryptocurrencies' prices.

KEYWORDS
cryptocurrencies, deep learning, hybrid model, price prediction, sentiment analysis

1 | INTRODUCTION

Since the introduction of blockchain by Nakamoto (2008) as an underlying technology for the first cryptocurrency, that is, Bitcoin, the application of this technology has been raised in many sectors, in particular, in the financial sector. According to Chang et al. (2020), blockchain has significant impacts on financial services by introducing decentralization, transparency, security, and innovation. 'Finance is the natural scenario of the blockchain, and cryptocurrencies are also by far one of its most successful applications', Chang et al. (2020). Cryptocurrencies operate on decentralized networks of blockchain which eliminates the central control of banks over financial transactions. The transparency of blockchain increases the trust among participants in the cryptocurrency ecosystem which consequently positioning the cryptocurrencies as a favoured investment target of investors all over the world. Cryptocurrencies offer diversification and hedging benefits for investors by considerably reducing portfolio risk (Sun et al., 2020). Due to cryptocurrencies' decentralization, immutability, and security, they have become a global phenomenon attracting a significant number of users (Oyedele et al., 2023). Accurate price forecasts are important for asset allocation and gaining profits in

cryptocurrency markets. In this regard, a wide variety of techniques have been attempted and utilized by researchers of this domain, specifically by using artificial intelligence (Chang et al., 2021). However, developing a reliable and accurate model that can perform well in forecasting of all cryptocurrencies' prices is a challenging task due to the different characteristics of each market. That is why researchers evaluate the effectiveness of their proposed models on a limited number of cryptocurrencies. In this study, we aim to address such difficulty by proposing a prediction model with a flexible architecture that can be adaptable for different cryptocurrencies.

Our study has been significantly inspired by two streams of studies in the literature: (1) studies that utilize neural networks especially the deep learning (DL) including long short-term memory (LSTM) and convolutional neural networks (CNNs) (Chen et al., 2021; Ji et al., 2019; Lahmiri & Bekiros, 2019a; Lahmiri & Bekiros, 2019b; Lahmiri & Bekiros, 2020; Oyedele et al., 2023; Patel et al., 2020; Zhang et al., 2021; Zoumpekas et al., 2020), (2) studies that investigate the effectiveness of social media data in predicting cryptocurrencies' prices (Anbaee Farimani et al., 2022; Kraaijeveld & de Smedt, 2020; Ortu et al., 2022; Zou & Herremans, 2022).

The novelty of our work can be seen from both experimental and methodological aspects:

1. From the experimental point of view:

- Most previous studies have focused on Bitcoin as it has the largest market capitalization (Chen et al., 2021; Ji et al., 2019; Lahmiri & Bekiros, 2020; Zou & Herremans, 2022). We evaluate the performance of our price prediction models on a large set consisting of 27 cryptocurrencies. Although other cryptocurrencies have not attracted much attention, it is worth investigating the robustness of the models on less famous cryptocurrencies to offer a suitable strategy for their price prediction and understand their overall price dynamics (Oyedele et al., 2023).
- We perform sentiment analysis on a large text dataset comprising of 16+ million tweets related to cryptocurrencies that were posted between 1 January 2021 and 31 December 2021. We show the effectiveness of using sentiment data in predicting the close price of more than 70% of the cryptocurrencies studied in this work. The reason that we use Twitter data is that this social media platform is the most popular one among traders and provides a combination of both news and investor sentiment at the same time (Kraaijeveld & de Smedt, 2020).

2. From the methodological point of view:

- We propose a hybrid prediction model based on two popular DL models namely LSTM and CNN with a flexible input layer that can be customized for different cryptocurrencies. Due to the highly volatile nature of cryptocurrencies, developing a single model that can perform well on multiple cryptocurrency datasets is not possible. The flexibility of our model allows the choice of the best architecture for each cryptocurrency. It is worth mentioning that the reason for using both CNN and LSTM models is that the former performs well in finding local dependencies that are independent from time and the latter can catch long-term dependencies in data.
- The novelty of our sentiment analysis compared to the other similar studies such as Kraaijeveld and de Smedt (2020) is that they use a lexicon-based approach to calculate polarity scores for each tweet while we employ three pre-trained language models in an ensemble way to calculate sentiment scores of the tweets. While lexicon-based sentiment analysis relies on limited pre-defined sentiment dictionaries, the pre-trained language models have been trained on vast amount of text data and can capture the meaning of the words within their surrounding context. Moreover, using pre-trained language models that have been fine-tuned specifically for the finance and social media data, will increase the accuracy of predicted sentiments. Furthermore, compared to the work of Zou and Herremans (2022) in which only one pre-trained language model was employed to perform sentiment analysis, we use three of them in an ensemble way and this reduces the risk of relying on single model's predictions.

To the best of our knowledge, the combination of ensemble sentiment classifiers and a hybrid price prediction model with a flexible input layer has not been proposed by other researchers, making our framework novel and innovative.

Afterward, the paper proceeds as follows. Section 2 overviews the literature on the predictability of cryptocurrency prices by using artificial neural networks (ANNs) and social media. Section 3 presents the models and Section 4 present data and results. A discussion of the results is provided in Section 5. Finally, Section 6 concludes the work and suggests future research directions.

2 | LITERATURE REVIEW

2.1 | Neural networks and cryptocurrency price prediction

The high performance of both LSTM and gated recurrent unit (GRU) models in time series forecasting have been proved in many studies as they can handle the problem of gradient vanishing (Bengio et al., 1994) in long sequences. Patel et al. (2020) developed a hybrid model that

concatenates the results of the GRU and LSTM networks and predicts the price of Litecoin and Monero using a dense layer. To prepare the training data, the authors created multiple input–output samples where the inputs are the average price and its 30 previous values, and the output is the next day's average price. After training the model, the last 30 observations in the training set were used to predict the next day's price, that is, the first price in the test set. The predicted price was used in the next input sequence along with its last 30 values to predict the next value. This process continues for k times ($k = 1$, 3, and 7 days) resulting in k forecasts that can be compared with their real values. The results showed that the proposed hybrid model is superior to a LSTM baseline model for all k values. However, they found that their hybrid model is more effective in short term predictions, that is, for $k = 1$.

Chen et al. (2021) collected a various range of technical and economic features over four different periods for Bitcoin and by using two feature selection approaches, they determined the most important factors affecting Bitcoin's price in each period. The selected features were used as inputs for a LSTM network. To find out whether these factors play the major role in price forecasting, the results were compared to the results of four baseline models that were trained with only historical price data. The baseline models were the autoregressive integrated moving average (ARIMA), support vector regression (SVR), adaptive network fuzzy inference system (ANFIS) and another LSTM, all trained using only historical price data. Their results showed that not only the economic and technology features used in the LSTM model provided higher prediction performances, but also their effects changed over time. In other words, factors that determine the price of Bitcoin in different periods are not the same.

Zoumpekas et al. (2020) compared both the prediction performance and complexity of six machine learning models in the forecasting of Ethereum closing price in each half-hour. The CNN models of this study were a two and three-layer network both performing one-dimensional convolutions as the authors were dealing with one-dimensional time series data. In addition to using the regular LSTM and GRU networks, the stacked LSTM and Bi-Directional LSTM (BiLSTM) were also investigated in this work. After validating all six networks, the best one for each case was selected to be evaluated on six randomly selected test sets. Indeed, for each model, the authors used the confidential intervals of obtained performance measures on six different test sets, and this was done for the purpose of robustness. They observed that the LSTM model was superior to the other networks and the second-best model was found to be the GRU. In terms of complexity, the recurrent neural networks (RNNs) seemed to be more complex than the CNNs as they are more memory intensive and take more time to be trained. One interesting work of the authors was developing a web-based application where in the back-end system, the data of the previous 30 min (6 observations in 5-min intervals) were collected and used by the pre-trained LSTM network to predict the next 5-min closing price of Ethereum. The results were displayed in the front end as live graphs.

Ji et al. (2019) addressed both classification (i.e., predicting whether the next period's price goes up or down) and regression problems by comparing various state-of-the-art DL methods. Using a sequence of previous m days ($m = 5$, 10, 20, 50, and 100) for each of the 18 Bitcoin blockchain features, they predicted whether the price would go up or down in the next day for the classification problem, while the regression problem was defined as predicting the value of the price for the next day. Training all models with different values of m, the authors found that for the regression, shorter sequences give better performance, while for the classification, longer sequences are required. Comparing the results of six different models (deep neural network [DNN], LSTM, CNN, a combination of CNN and RNN, etc.) with a few baseline models, the LSTM was turned out to have the best regression performance and the DNN was selected as the best classifier.

Another study that covers both trend and price predictions is the study of (Zhang et al., 2021) in which they proposed a new model named Weighted and attentive memory channels (WAMCs) consisting of three modules. Apart from proposing a new model, the novelty of this work is in the use of different cryptocurrencies' data to predict one cryptocurrency's closing price. In fact, four cryptocurrencies with a high degree of correlation and similar trends were selected for this purpose. Using a moving window of length 7 (i.e., historical prices) and four cryptocurrencies closing prices as channels, the input data samples were constructed as 3-dimensional tensors (7, 1, 4) and one of the four cryptocurrencies was selected as the target. In the attentive memory module of the WAMC, the authors used a self-attention mechanism over the outputs of a GRU network to capture important information in different time steps. The outputs of this module are fed into the channel-wise weighting module where the importance of each channel is determined using a GRU model. The last module performs convolution and pooling operations on the outputs from the previous module and it predicts either the price or its movement direction for the target cryptocurrency. The results revealed that the WAMC outperforms all baseline models considered in this study. Moreover, the WAMC was trained and tested three times, each time without one of the three modules, and the results showed that its performance is worse than the performance of the full WAMC model consisting of all modules.

In an extensive investigation by Lahmiri and Bekiros (2020), three different sets of models were analysed for prediction of the next 5-min Bitcoin price using its five previous observations. The authors utilized several complexity measures to quantify the nonlinear dynamics of the Bitcoin series data. For instance, the sample entropy was used to measure the randomness of data. The results of this complexity analysis required the authors to employ advanced machine learning methods that can capture noisy and nonlinear behaviours in data. Out of seven different models, the Bayesian regularization neural network (BRNN) was found to have the best performance in prediction of Bitcoin's high-frequency price data. The reason mentioned by the authors is that the objective function of the Bayesian optimization in this network penalizes large weights and hence, the network is less likely to overfit.

Kim et al., 2021 utilized various blockchain information of the Ethereum such as transaction volume, mining difficulty, and network activity as features for prediction of Ethereum price. Conducting a stepwise analysis for two machine learning models, namely, the ANN and support vector

machine (SVM), they showed that the ANN model that includes Ethereum- and Bitcoin-specific blockchain information along with macro-economic factors has the lowest prediction error among all other combination of features. The authors of this study mentioned that social media data such as user sentiment should be used in their future studies to further improve the accuracy of the Ethereum price predictions.

In a recent study, Oyedele et al. (2023) evaluated the prediction performance of DL and boosted tree-based techniques on six cryptocurrency datasets collected from three different data sources. The reason for using multiple data sources is to investigate the robustness of prediction models in terms of how they respond to patterns in multiple data sources. Another aspect of their work is that they designed two training sets where one has peaks and lows of prices and the other one where higher spikes are not part of the training set. The authors showed that their models performed well and produced more accurate results on datasets with more training data that include peaks and drops in prices. Another conclusion was that CNN model is more reliable with limited training data compared to other DL and tree-based techniques and it is easily generalizable for predicting several cryptocurrencies' daily closing prices.

2.2 | Social media and cryptocurrency price prediction

One of the first researchers that studied the prediction power of Twitter sentiment for a large set of cryptocurrencies was Kraaijeveld and de Smedt (2020). The authors studied to what extent public Twitter sentiment can be used to predict price returns for the nine largest cryptocurrencies. They collected more than 24 million tweets and by defining a few bot detection measures, they removed tweets that were generated by bots. They used the valence aware dictionary and sentiment reasoner (VADER) algorithm (Hutto & Gilbert, 2014) which is a lexicon-based approach to calculate polarity scores for each tweet and then aggregated the polarity scores into daily and hourly intervals. Using the bivariate Granger-causality test, they concluded that Twitter sentiment has predictive power for three out of nine cryptocurrencies in their study.

Ortu et al. (2022) used three sets of input data in their study to predict the price movements of Bitcoin and Ethereum. They trained the pre-trained BERT-base-case model (Devlin et al., 2018) on some labelled benchmark datasets and then used the trained model to extract emotions and sentiments of GitHub and Reddit comments. They built hourly and daily time series data by aggregating sentiments and emotions of multiple comments for each hour and day, respectively. These social media indicators along with other kinds of data were fed into four different models. They showed that for the daily frequency, the models that are fed with social media indicators outperform the other models.

Anbaee Farimani et al. (2022) evaluated the effectiveness of using both news sentiment and informative indicators for price regression on Forex and Cryptocurrency markets. They built news sentiment time series via the probability distribution of news title embedding generated through the FinBERT (Araci, 2019) classifier layer. They showed that considering both market data and news sentiment can significantly reduce the error for price regression.

Zou and Herremans (2022) used FinBERT (Araci, 2019) to generate actual word embeddings. They stacked embeddings of tweets from the same day into 20-dimensional vectors. These stacked embeddings were fed into a CNN model while the technical indicators and other data were passed into an SVM model. They put together these two models in both sequential and parallel formats to predict extreme price movement. Their results indicate that the sequential model outperforms the SVM which uses only price and technical analysis data. This shows that adding prediction based on Twitter content improves the overall performance of the model. Table 1 lists the recent studies in cryptocurrency prediction.

3 | METHODS

In this section, we present LSTM, CNN, sentiment classifiers and various architectures of our hybrid model. In addition, we explain how we selected our sentiment classifiers from the available pre-trained language models. Then, we describe hyper-parameter tuning procedure. Finally, we present the performance measure used to assess the performance of each model.

3.1 | Long short-term memory

Long short-term memory networks are a variant of RNN introduced by Hochreiter and Schmidhuber (1997). As it is explained in (Bengio et al., 1994), a task displays long-term dependencies if prediction of the desired output at time t depends on input presented at an earlier time T. Gradient-based learning algorithms such as RNNs face difficulties in performing such tasks and their parameters settle in sub-optimal solutions that take into account short-term dependencies but not long-term dependencies. LSTM networks allow the information to persist in the network and could learn both short and long-term dependencies in the input sequence. Figure 1 shows the different parts of a typical LSTM unit which includes cell state (denoted by C), hidden state (denoted by h), and three gates namely forget, input, and output gates. These gates control the flow of the information in the LSTM.

TABLE 1 Recent studies in cryptocurrency price prediction.

References	Market/data	Data period/ frequency	Input features	Methodology
Patel et al. (2020)	Litecoin and Monero	Litecoin: 24 August 2016 to 23 February 2020 Monero: 30 January 2015 to 23 February 2020/ daily	Average price of cryptocurrency for the day	LSTM-GRU hybrid
Chen et al. (2021)	Bitcoin	Four important periods for Bitcoin: 1 August 2011 to 31 December 2013 1 August 2013 to 31 December 2014 1 July 2014 to 31 December 2017 1 July 2015 to 31 July 2018/daily	1. Technology category: Blockchain information, public attention 2. Economic category: Macroeconomic indicators, Global currency ratios	ANN and RF for feature selection LSTM for price prediction
Zoumpekas et al. (2020)	Ethereum	Training data: 8 August 2015 to 28 May 2018 Test data were collected in six randomly selected timeframes/5-min	Historical price data, transaction and blockchain data, user comments, economic factors and other community data. Google Trends and Wikipedia	2-layer CNN, 3-layer CNN, LSTM, BiLSTM, Stacked LSTM, GRU
Ji et al. (2019)	Bitcoin	29 November 2011 to 31 December 2018/daily	A sequence of 18 daily value of Bitcoin blockchain features such as block size, the total number of transactions, and so forth.	DNN, LSTM, CNN, a combination of CNNs and RNNs (CRNN), residual network (ResNet), Ensemble of LSTM, DNN and CNN
Zhang et al. (2021)	BTC, Bitcoin Cash, Litecoin, ETH, EOS, and XRP	23 July 2017 to 15 July 2020/daily	A window length (7, 11, 15) of different cryptocurrencies closing prices	A weighted and attentive memory channels (WAMC) model consisting of three modules
Lahmiri & Bekiros (2020)	Bitcoin	1 January 2016 to 16 March 2018/5-min	Previous five observations of prices	SVR, Gaussian Poisson regression (GRP), regression tree (RT), kNN, Feedforward neural network (FFNN), Bayesian regularization neural network (BRNN), and radial basis function neural network (RBFNN)
Kim et al. (2021)	Ethereum	11 August 2015 to 28 November 2018/daily	Ethereum and other cryptocurrencies (Bitcoin, Litecoin, Dashcoin) blockchain information, global currency ratio, and macro-economic development index	ANN, SVM
Oyedele et al. (2023)	Six cryptocurrencies (BTC, ETH, BNBUSD, LTC, XLM and DOGE)	1 January 2018 to 31 December 2021	OHLCV and technical indicators	Boosted tree based (AdaBoost, GBM, XGB) and deep learning (DFNN, CNN, GRU) techniques
Kraaijeveld & de Smedt (2020)	Nine cryptocurrencies	Between 4 June 2018, and 4 August 2018/daily and hourly	Tweets	Lexicon based sentiment analysis
Ortu et al. (2022)	Bitcoin, Ethereum	January 2017 to January 2021/daily and hourly	Technical and trading indicators, GitHub, and Reddit comments	BERT for extracting the sentiment, emotion, VAD, multivariate attention long

(Continues)

TABLE 1 (Continued)

References	Market/data	Data period/ frequency	Input features	Methodology
				short-term memory fully convolutional network (MALSTM-FCN) for the classification task
Anbaee Farimani et al. (2022)	Forex and Cryptocurrency markets	From September 2018 to May 2021/hourly	Technical indicators and news sentiment	FinBERT for extracting news sentiment, LSTM for price prediction part
Zou & Herremans (2022)	Bitcoin	From 2015 to 2021/ daily	Tweets, OHLCV data and 13 technical indicators, with correlated asset prices such as Ethereum and Gold	FinBERT for extracting tweets sentiment, combination of SVM and CNN for price movement prediction

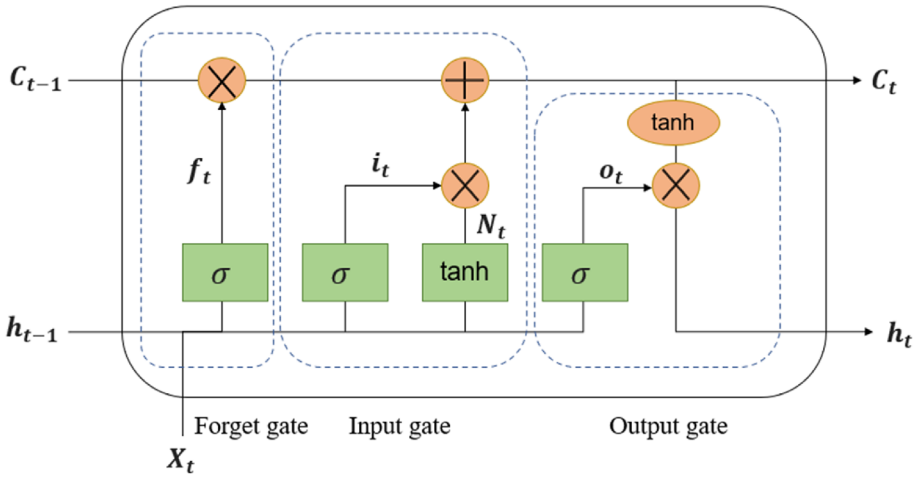

FIGURE 1 Long short-term memory unit structure.

3.1.1 | Forget gate

LSTM first decides whether information from the previous timestamp should be kept or not. This can be shown by Equation (1):

$$f_t = \sigma(X_t.U_f + H_{t-1}.W_f) \tag{1}$$

where X_t is input to the current timestamp, t, U_f is weight associated with the input, $H_{t-1}$is the hidden state of the previous timestamp, t-1, and W_f is the weight matrix associated with hidden state. A sigmoid function is applied to f_t and its value becomes 0 or 1. Value of 0 means that the network forgets everything and value of 1 indicates it will keep every information from t-1.

3.1.2 | Input gate

This gate quantifies the importance of the new information that are entered the unit.

$$i_t = \sigma(X_t.U_i + H_{t-1}.W_i) \tag{2}$$

U_i and W_i are the weight matrices associated with input and hidden states. Another sigmoid function is applied to i_t and its value will be between 0 and 1. The value that is close to 1 dedicates more importance to the input information. The new information that will be added or subtracted from the cell state of the LSTM is shown by Equation (3). This time a tan h activation function is used to make the input information between −1

for subtraction of information from cell state and 1 for adding information to the cell state. However, the new information, N_t, will be updated by input and forget gates before entering the cell state (Equation (4)).

$$N_t = \tan h(X_t.U_c + H_{t-1}.W_c) \tag{3}$$

$$C_t = f_t.C_{t-1} + i_t.N_t \tag{4}$$

C_{t-1} is the cell state in the previous timestamp, $t-1$.

3.1.3 | Output gate

This gate decides what information will be outputted as a hidden state at time t. This is determined by Equations (5) and (6).

$$O_t = \sigma(X_t.U_o + H_{t-1}.W_o) \tag{5}$$

$$h_t = O_t.\tan h(C_t) \tag{6}$$

3.2 | Convolutional neural networks

Convolutional neural networks introduced by Lecun et al. (1998) have many applications in image analysis and classification problems. Recently, it has been shown to be also effective for sequence data analysis (Ji et al., 2019). Figure 2 shows a simple 1-dimensional CNN architecture.

The input layer passes a fixed-length sequence from the full time series to the convolution layer. In our case, the input layer is fed with a sequence of h time steps from the closing price and/or sentiment scores. The convolution layer slides one or multiple filters with the size of k over the input sequence one step at a time where $k < h$. From the mathematical point of view, a 1-dimensional (1-d) convolution operation is the inner product of two vectors: a subsequence from the input sequence (X) with length of k, and the filter, $w \epsilon R^k$. Equation (7) shows the 1-d convolution operation:

$$C_j = \sum_{i=1}^{k} w_i.X_i \tag{7}$$

where i and j are indices of the filter and output sequence, respectively. X_i is the ith element in the input sequence X, w_i is the ith element in the filter (or kernel) of length k and C_j is the jth element of the new one-dimensional output sequence (i.e., the convolution). Consequently, the length of the output vectors (shown by n in Figure 2) is not equal to the length of the input sequence. The resulting vectors are somehow the smoothed version of the input sequence that include the most important features and information from that sequence. It is worth mentioning that the 1-d

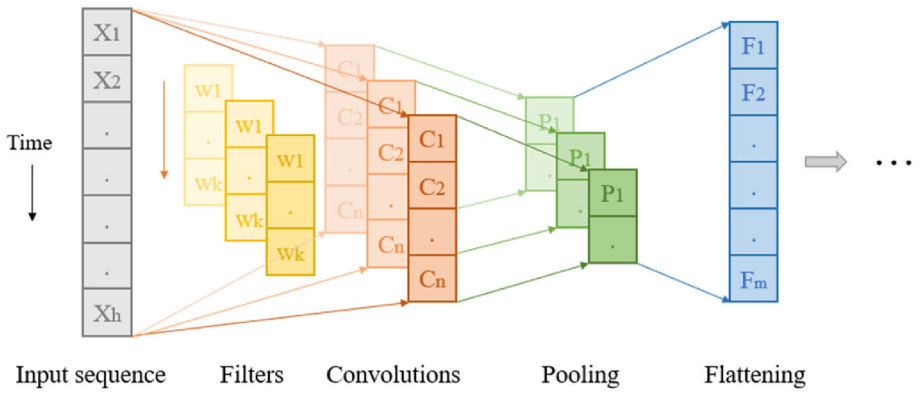

FIGURE 2 One-dimensional convolutional neural network architecture.

convolution can be applied to more than one input sequence as the filter(s) only move in one dimension along the axis of time and the convolution operations are done independently for each input sequence.

An activation function is applied after the convolution to produce the output (not shown in Figure 2). The pooling layer comes after that and it combines the output of the previous layer at certain locations into a single neuron by taking the mean, maximum, or minimum of all values of those positions (Pérez-Enciso & Zingaretti, 2019).

Finally, the outputs from the pooling operation are flattened to create a unique vector that can be followed by other layers such as a Dense layer to produce predictions. In our case, we will concatenate this flattened layer vector with the output from the LSTM model as part of our hybrid model (see Section 3.3). That is why we showed the last layer of the CNN with three dots in Figure 2.

For more details on the architecture of CNNs, please refer to (Kiranyaz et al., 2021) and (Pérez-Enciso & Zingaretti, 2019). As it has been mentioned by Kiranyaz et al. (2021), one advantage of 1-d CNNs compared to the 2-d CNNs is that 1-d CNNs with relatively shallow architectures (i.e., small number of hidden layers and neurons) are able to learn challenging tasks involving 1-d signals. Therefore, our 1-d CNN part of our proposed model will be a shallow one and consists of only one layer. On the other hand, with more layers, the important information from the input sequence might be lost as a consequence of using more convolution operations which will decrease the prediction performance of the proposed model.

3.3 | Proposed hybrid model

Our proposed model is a hybrid model, and it consists of different layers including LSTM, CNN, Attention, and Dense layers. The overall diagram of the model is exhibited in Figure 3. In the input layer, a history of previous h days of close prices $(C_{t-h},\ldots,C_{t-1})$ and sentiment data $(S_{t-h},\ldots,S_{t-1})$ is fed into the LSTM and CNN layers, then, the output layer generates the close price at day t, that is, C_t. The reason that we used both CNN and LSTM layers in parallel is that the former performs well in finding local dependencies and the latter can catch long-term dependencies in data. This way, more informative representations will be generated from input features and having these layers in parallel will give us two distinctive representations from these two layers. It is important to remark that while there is only one CNN layer in the model, the number of LSTM layers could vary from one cryptocurrency to another based on hyperparameter tuning results. The reason for including only one CNN layer is that the convolution operation of the CNN reduces the dimension of data and we do not want to lose information because of using more convolutions.

The problem with LSTM is that in the case of long sequences, there is a high probability that the initial context has been lost by the end of the sequence. 'Attention' technique (Vaswani et al., 2017) allows the network to focus on different parts of the input sequence at every stage of the output sequence allowing the context to be preserved from beginning to end. In our hybrid model, there is an optional attention layer after the LSTM layer, and it allows the model to learn where to pay attention in the input sequence by relating it to the items in the output sequence. Our input and output sequences from the LSTM layer have a length of h and the attention layer makes the LSTM layer focuses on the most informative items in the input sequence for better representing and relating them to the output sequence.

The output of the attention layer (or LSTM layer if there was no attention layer) and the CNN layer are concatenated, and the resulting vector is passed through a dense layer followed by a dropout layer to mitigate any possible overfitting problem. In the output layer, a dense layer with a linear activation function generates the predicted value for the close price at day t.

We constructed various architectures of this hybrid model, and our goal was not only to find the best architecture for each cryptocurrency, but also to find the best way of passing input data into the model. Table 2 presents 20 different models that we explored in this study. For example, CNN(C)_LSTM_At(S) refers to an architecture where there is an attention layer (At) in the model, the close price (C) and sentiment (S) vectors are fed into the CNN and LSTM layers, respectively.

3.4 | Hyper-parameters tuning

We decided to perform hyperparameter tuning for only one version of our hybrid model and we selected the CNN(C)_LSTM_At(C) model because it has all components and is fed with only close price data. We aimed to find the best version of a reference model and then by removing components and adding sentiment data in different ways, we obtained the other 19 versions of our model. Therefore, for each cryptocurrency, the close price data was divided into 80% and 20% as training-validation and testing sets, respectively. The last 20% of the training-validation set was considered as the validation set for hyperparameter tuning. The CNN(C)_LSTM_At(C) model was trained on the training set and the trained model was used to predict close prices in the validation set. More precisely, the last h values of the close price from the training set were used to predict the first close price in the validation set, the last h-1 values of the close price from the training set along with the first value of the close price in the validation set were used to predict the second close price in the validation set, and so on. This process generated 58 predicted values, equal to the size of the validation set. To find more robust results and mitigate the randomness effect of neural networks, we repeated each training

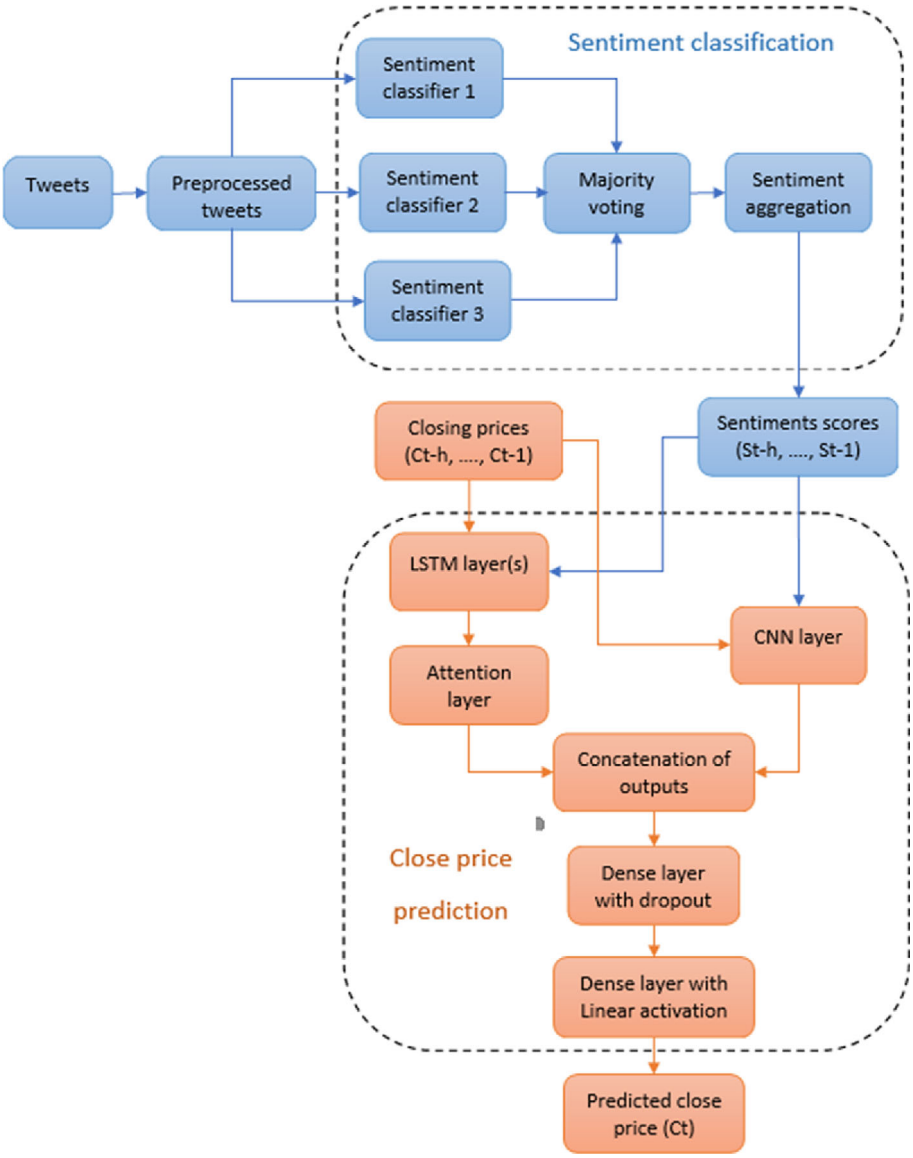

FIGURE 3 Overview of the hybrid model with different ways of feeding input data.

and validation process five times, and the optimal set of hyperparameters for each cryptocurrency was found by comparing the average RMSE value calculated over five runs.

The list of hyper-parameter and their respective search interval are presented in Table 3. Using a grid search method and fixing the value of 21 for *h* (history of input data), 'mse' for the loss function, 'adam' as the optimizer, and training for 25 epochs, we tuned the CNN(C)_LSTM_At (C) model 27 times. This is because there is no one unique set of hyperparameters that is optimal for all cryptocurrencies and the optimization results vary from one cryptocurrency to another one. Similar approaches can be seen in the work of (Oyedele et al., 2023). Once the optimal hyperparameters are found, the whole training-validation set is used to train the models and generate predicted values for the testing set.

3.5 | Sentiment classifiers

During the last few years, transformer-based language models have shown significant improvements in Natural Language Processing (NLP) tasks and have become the state-of-the-art in this field. Among those models, Bidirectional Encoder Representations from Transformers so-called BERT (Devlin et al., 2018) is the most well-known one. Due to space limitation in this paper, the details of the BERT architecture are not included here. Please refer to the original paper for more details and explanation. BERT was pre-trained on BooksCorpus (800 M words) (Zhu et al., 2015) and English Wikipedia (2500 M words) and the resulting parameters can be further optimized for a specific down-stream task such as sentiment

TABLE 2 Various architectures of the model.

1. LSTM(C)
2. LSTM_At(C)
3. LSTM(C,S)
4. LSTM_At(C,S)
5. CNN(C)_LSTM(C)
6. CNN(C)_LSTM_At(C)
7. CNN(C)_LSTM(S)
8. CNN(C)_LSTM_At(S)
9. CNN(S)_LSTM(C)
10. CNN(S)_LSTM_At(C)
11. CNN(C)_LSTM(C,S)
12. CNN(C)_LSTM_At(C,S)
13. CNN(C,S)_LSTM(C)
14. CNN(C,S)_LSTM_At(C)
15. CNN(S)_LSTM(C,S)
16. CNN(S)_LSTM_At(C,S)
17. CNN(C,S)_LSTM(S)
18. CNN(C,S)_LSTM_At(S)
19. CNN(C,S)_LSTM(C,S)
20. CNN(C,S)_LSTM_At(C,S)

TABLE 3 Hyperparameters and parameters in the tuning process.

Hyperparameter	Searching interval
Number of LSTM layers	[1, 2, 3]
Number of units in LSTM hidden layers	[50, 100]
Number of units in dense layers	[64, 128]
Number of filters in the CNN layer	[32, 64]
Filter size in the CNN layer (k)	[2, 3, 4]
Activation functions	['tanh', 'relu']
Batch size	[16, 32]
Fixed parameters	
$h = 21$	Loss = 'mse'
Epoch = 25	Optimizer = 'adam' with the default learning rate of 0.001

classification. This process is called fine-tuning, and it consists of adding one additional output layer to the BERT model for the classification task and then training the model on a sentiment labelled text dataset. Devlin et al. (2018) fine-tuned the pre-trained BERT model on the benchmark dataset of Stanford Sentiment Treebank (Socher et al., 2013) which is a binary single-sentence classification task consisting of sentences extracted from movie reviews with human annotations of their sentiment. Their results outperformed the other state-of-the-art results at the time of BERT model introduction.

The problem with general-purpose language models such as BERT is that they have not been trained on domain-specific corpus and consequently, they might not be perfect for down-stream tasks in especial domains. Recently, researchers have created domain-specific BERT models in which they pre-trained BERT model from scratch using a large corpus in a specific domain. Some examples are SciBERT (Beltagy et al., 2019) trained on a large multi-domain corpus of scientific publications and has shown statistically significant improvements over BERT for several NLP tasks, ClinicalBERT (Huang et al., 2019) pre-trained on clinical notes corpus for the purpose of predicting hospital readmission.

FinBERT (Araci, 2019) is the first finance domain specific BERT model, and it has been pretrained on a large financial corpus called TRC2-financial by the author. It is a subset of Reuters' TRC2 that consists of 1.8 M news articles published by Reuters between 2008 and 2010.

The pre-trained FinBERT model was also fine-tuned on Financial PhraseBank from (Malo et al., 2013) for the sentiment classification tasks in financial domain. As it was reported in (Araci, 2019), the results of financial sentiment analysis tasks by FinBERT show 15% improvement over the generic BERT models. This finding along with the use of FinBERT in several similar studies to ours such as (Zou & Herremans, 2022) and (Anbaee Farimani et al., 2022) motivated us to use the fine-tuned version of this model as one of our sentiment classifiers. It should be noted that we do not fine-tune this model further, and we apply the original fine-tuned model to our Twitter data directly. This is mainly because our tweets are unlabelled and fine-tuning the FinBERT model on other labelled datasets will be redundant as it has been fine-tuned on Financial PhraseBank dataset by the authors for the purpose of sentiment classification. Since we do not have true labels of sentiments for the tweets, we could not measure the performance of FinBERT model. Consequently, we can not rely on the sentiment labels generated by only one model, that is, the FinBERT. To overcome this issue, we decided to employ two more fine-tuned language models as our second and third sentiment classifiers. Then, by taking a majority vote approach, we can find the final sentiment label for each tweet. This way, we have an ensemble of three state-of-the-art sentiment classifiers where we could rely on our final predicted sentiments (see sentiment classification part in Figure 3).

Our measure to select the other two sentiment classifiers was to use language models that have been fine-tuned for social media data, particularly, Twitter data. This way, we addressed our sentiment classification problem not just from the financial domain aspect, which was covered by utilizing the FinBERT model, but also from the social media aspect. In this regard, one of the well-known language models is a RoBERTa-base model that has been trained on 124 M tweets from January 2018 to December 2021 (Loureiro et al., 2022). This model was also fine-tuned for sentiment analysis with the TweetEval benchmark (Barbieri et al., 2020) which is commonly used to evaluate the performance of language models on Twitter data. According to the findings of (Loureiro et al., 2022), their model achieves the highest accuracy (73.7%) in sentiment classification task on TweetEval dataset compared to other existing state-of-the-art methods. We employ this fine-tuned model as our second sentiment classifier and in this paper, we will use the term Twitter_RoBERTa for it. It is worth mentioning that there is another state-of-the-art language model called BERTweet model (Quoc Nguyen et al., 2020), which has been trained on over 900 M tweets (posted between 2013 and 2019), but we decided to go with RoBERTa_tweet since it was pre-trained with most recent tweets (posted between 2018 and 2021).

The third sentiment classifier that we utilized in our study is called RoBERTuito (Pérez et al., 2021) which is a pre-trained language model for user-generated text in Spanish, trained on over 500 million tweets. English evaluation results of RoBERTuito on SemEval 2017 Task-4 dataset (Rosenthal et al., 2017) indicate competitive performance against monolingual models such as BERT, BERTweet, and RoBERTa. Therefore, we decided to use RoBERTuito as our third sentiment classifier in this study. Table 4 lists the sentiment classifiers used in our work.

3.6 | Performance evaluation metric

Finally, to evaluate prediction performance of the models, we used the root mean squared error (RMSE) measure specified in Equation (8).

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(C_i - \hat{C}_i\right)^2} \tag{8}$$

where C_i and $\hat{C}_i$ are the real and predicted values of close price at time i, respectively, $n = 73$ is the number of predictions which is equal to the size of testing sample for each cryptocurrency (20% of 1 year data). Lower values of the RMSE indicate better prediction performances.

4 | DATA AND RESULTS

Previous studies mostly focus on Bitcoin, Ethereum and a few other cryptocurrencies since they are the most traded cryptocurrencies in the market. Aiming to do experiments on a large set of cryptocurrencies, we collected Twitter and price data for 27 cryptocurrencies. The details of data collection come below.

TABLE 4 Sentiment classifiers used in this study.

Sentiment classifier	Pre-trained dataset	Sentiment analysis dataset
FinBERT (Araci, 2019)	TRC2-financial (a subset of Reuters' TRC2)	Financial PhraseBank (Malo et al., 2013)
Twitter_RoBERTa (Loureiro et al., 2022)	123.86 M tweets until the end of 2021	TweetEval benchmark (Barbieri et al., 2020)
RoBERTuito (Pérez et al., 2021)	Over 500 million Spanish tweets	SemEval 2017 Task-4 dataset (Rosenthal et al., 2017)

4.1 | Price data

In the first step, we prepared a list of 100 most traded cryptocurrencies that have available data on the website 'www.investing.com'. Then, cryptocurrencies were sorted based on their entrance date into the market. The 27 oldest ones that were selected to be used in this study are as below:

Bitcoin, XRP, Monero, Ethereum, Ethereum Classic, Litecoin, Zcash, Stellar, Dash, Tether, IOTA, EOS, OMG Network, Bitcoin Cash, Waves, Neo, Binance Coin, TRON, Cardano, Maker, Filecoin, Decentraland, Chainlink, Tezos, Dai, Enjin Coin, Theta.

We collected daily close price, high price, low price, open price and volume of trades for each cryptocurrency for a period of 1 year from 1 January 2021 to 31 December 2021. The 'close price' was selected as the target variable to be predicted by our prediction models following the most similar studies (Chen et al., 2021; Ji et al., 2019; Patel et al., 2020). Summary statistics of the close price data are available in Table 5. Figure 4 presents the close price of six top cryptocurrencies over the entire data period. As it can be seen in this figure, cryptocurrencies are highly volatile and predicting their future behaviour with a high accuracy is a difficult task. After close look at all 27 cryptocurrencies' close price data, we did not find any sign of missing values. The only data pre-processing was to scale the data so that the close prices are between 0 and 1. The details will be explained later in Section 4.3.

4.2 | Twitter data

An open-source python library named *snscrape* (https://github.com/JustAnotherArchivist/snscrape) was used to collect cryptocurrency-related tweets through the web-scrapping method. As a search key parameter, for each cryptocurrency, we considered a few hashtags based on their

TABLE 5 Summary statistics of close price for all 27 cryptocurrencies.

Cryptocurrency	*N*	Min	Max	Mean	Standard deviation
BinanceCoin	365	37.72	676.6	378.3	169.0
Bitcoin	365	29359.9	67,528	47,411	9761.8
BitcoinCash	365	341.53	1547	605.7	187.4
Cardano	365	0.175	2.965	1.499	0.6151
Chainlink	365	11.84	52.26	26.63	7.0149
Dai	365	0.9946	1.005	1.000	0.0013
Dash	365	86.92	443.6	194.0	65.49
Decentraland	365	0.0788	5.195	1.212	1.152
EnjinCoin	365	0.1320	4.490	1.758	0.9088
EOS	365	2.503	14.47	4.664	1.598
Ethereum	365	729.1	4808	2777	1024
EthereumClassic	365	5.692	133.8	42.15	23.81
Filecoin	365	21.05	190.9	69.63	39.24
IOTA	365	0.2856	2.515	1.205	0.4462
Litecoin	365	107.3	386.8	185.7	48.09
Maker	365	581.0	5985	2757	856
Monero	365	125.5	483.7	247.7	64.50
Neo	365	14.42	122.8	46.24	20.40
OMG_Network	365	2.369	19.10	7.247	3.490
Stellar	365	0.1275	0.7317	0.3631	0.1028
Tether	365	0.9990	1.003	1.000	0.0005
Tezos	365	1.997	8.699	4.535	1.487
Theta	365	1.743	14.19	6.563	2.795
Tron	365	0.02686	0.1639	0.0802	0.0293
Waves	365	5.395	36.49	17.85	7.539
XRP	365	0.2210	1.836	0.8670	0.3485
Zcash	365	56.78	319.6	150.8	50.48

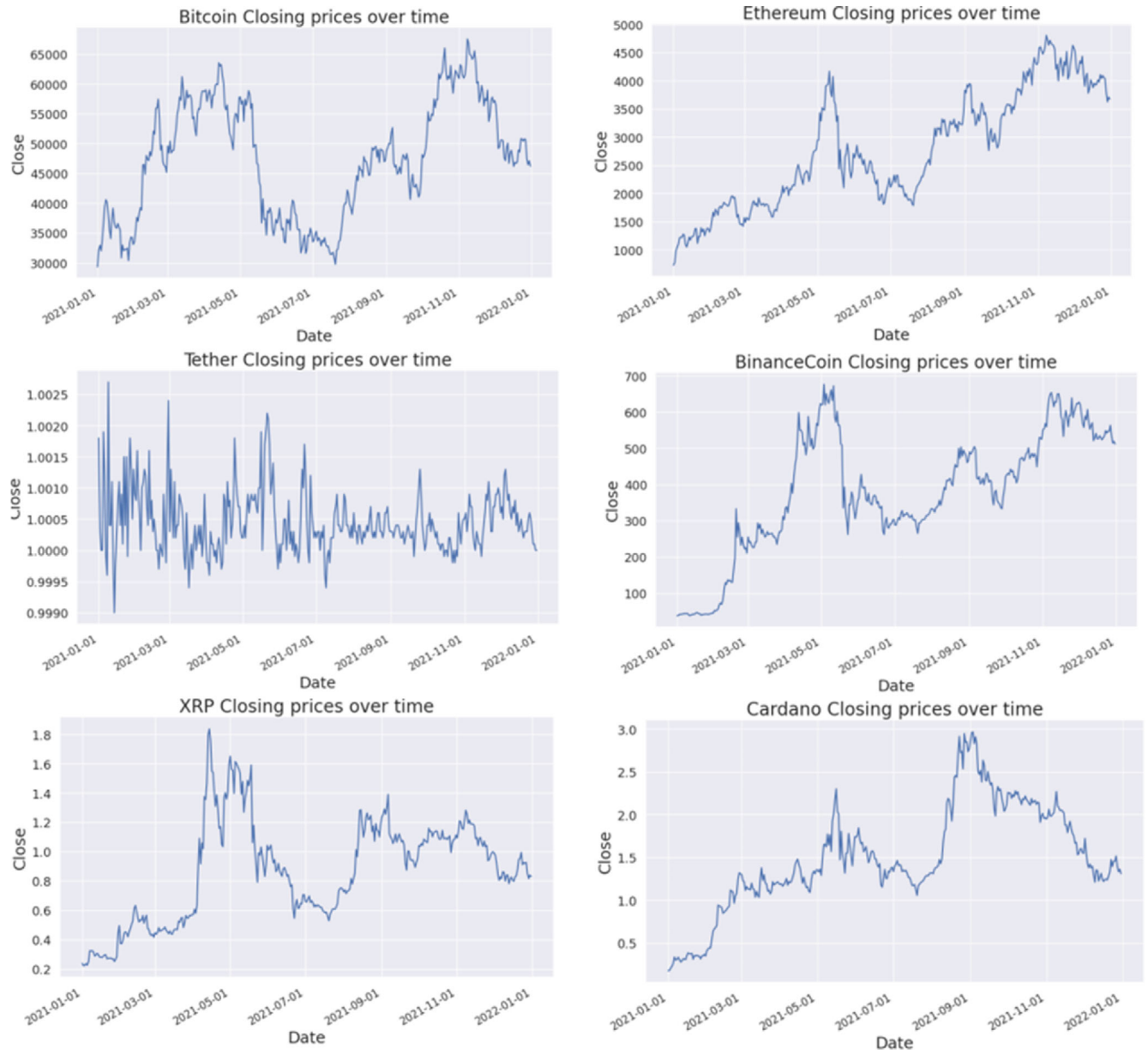

FIGURE 4 Closing price of cryptocurrencies over the entire period of study (1 January 2021 to 31 December 2021).

names and symbols. For example, in the case of Bitcoin, the search key was '*#BTC OR #btc OR #BITCOIN OR #Bitcoin*'. Since the data interval for our price dataset was from 1 January 2021 to 31 December 2021, we set the same interval for Twitter data. We had to set a limit on the number of tweets that we collect for each cryptocurrency. Otherwise, due to the limitation of our computational power, we had to run the models for months to perform sentiment analysis on our tweets, especially for some popular cryptocurrencies such as Bitcoin and Ethereum which are more discussed on Twitter. Therefore, we collected a maximum of 5000 tweets per day for each cryptocurrency, and for each tweet, we collected these features: date and time (datetime) of the posting, the text of the tweet, tweet ID, username, URL of the tweet and a few other attributes. Although in this study, we only used the text of the tweet along with its posting datetime, the other features will be utilized in our future works. To feed clean and less noisy text data into our models, we performed a few pre-processing (cleaning) operations which are available in Table 6.

4.3 | Results

In this section, we will investigate the prediction performance of 20 different models that we introduced in the previous section. The first model in Table 2, that is, LSTM(C) will be our base model and we are interested in comparing and ranking the prediction performance of other 19 models with respect to this model.

Our research questions are defined as follows:

- What is the effect of using sentiment data on the prediction performance of the models?
- Is having a CNN layer helpful in improving the prediction performance of the models?
- Does the attention layer improve the prediction performance of the models?
- What should be the length of the input sequence (i.e., the history parameter, *h*)?
- What should be the configuration of the input layer for each cryptocurrency?

As explained in Section 3.4., three fine-tuned language models were employed to classify the sentiment of each tweet into one of the three categories as negative, neutral and positive. This was because our tweets were unlabelled, and we could not use our own dataset to fine-tune the existing pre-trained language models. We relied on the available models that have been fine-tuned on similar datasets to ours and we used them directly to predict sentiment labels for our tweets. However, by applying a majority voting technique, we only kept tweets that were given the same label by at least two of the classifiers. This way, we obtained more robust and reliable sentiment labels. Figure 5 exhibits the results for Bitcoin tweets. For each cryptocurrency, the data loss due to majority voting was less than 1.5% (see Table 7).

Our model accepts time series data as its input, and we need to convert sentiment data into numbers that can be understood by our DL model. Following the work of (Hiew et al., 2019), we used Equation (9) to aggregate sentiments of multiple tweets on each day and calculated a sentiment score at day *t*. Accordingly, our price and sentiment data have the same granularity.

$$\text{Sentiment score}_t = \frac{\text{Pos}_t - \text{Neg}_t}{\text{Pos}_t + \text{Neu}_t + \text{Neg}_t} \tag{9}$$

Pos_t, Neu_t and Neg_t indicate the number of tweets with sentiment class of positive, neutral and negative at day *t*, respectively.

As presented in Table 2, we have 20 different versions of our model where the simplest and most complex ones are LSTM(C) and CNN(C,S)_LSTM_At(C,S), respectively. For each cryptocurrency, we used its optimal set of hyperparameters and input data (i.e., close prices and sentiment scores) to evaluate the prediction performance of all 20 models on their testing set. Since sentiment and price data are in different scales, we used 'min-max normalization' method to scale them into a range of 0 and 1. This also assists the optimization algorithm of our models to converge faster. Thus, the following equations were used:

$$\tilde{C}_t = \frac{C_t - C_{min}}{C_{max} - C_{min}} \tag{10}$$

$$\tilde{S}_t = \frac{S_t - S_{min}}{S_{max} - S_{min}} \tag{11}$$

TABLE 6 Results of tweets pre-processing operations with an example.

Tweet: '@Rookie b Yaaaaay you're lucky. This will totally explode June 3rd https://t.co/0l via @YouTube #Bitcoin $BTC'	
Operation	**Pre-processed tweet**
Convert all English alphabet characters to lower case	'@rookie b yaaaaay you're lucky. this will totally explode June 30th https://t.co/0lp via @youtube #bitcoin $btc'
Remove links and mentions	'b yaaaaay you're lucky. this will totally explode June 30th via #bitcoin $btc'
Remove ticker symbols ($)	'b yaaaaay you're lucky. this will totally explode June 30th via #bitcoin'
Remove hashtag symbols (#)	'b yaaaaay you're lucky. this will totally explode June 30th via bitcoin'
Replace any emojis with the text they represent	'b yaaaaay you're lucky. this will totally explode: exploding_head: June 30th via bitcoin'
Expand contraction words	'b yaaaaay you are lucky. this will totally explode: exploding_head: June 30th via bitcoin'
Reduce three or more repetitions of any character to two characters	'b yaay you are lucky. this will totally explode: exploding_head: June 30th via bitcoin'
Remove words that contain numbers	'b yaay you are lucky. this will totally explode: exploding_head: June via bitcoin'
Remove words with one character	'yaay you are lucky. this will totally explode: exploding_head: June via bitcoin'
Remove multiple spaces	'yaay you are lucky. this will totally explode: exploding_head: June via bitcoin'

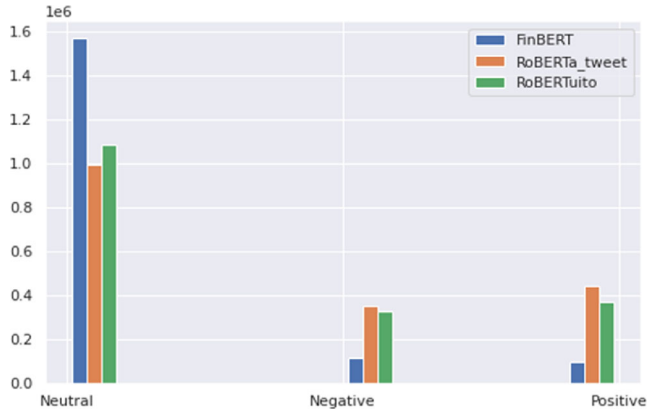

FIGURE 5 Frequency plot of predicted sentiments by each sentiment classifier – Bitcoin Twitter data as an example.

TABLE 7 Number of tweets before cleaning, after cleaning and after majority voting in sentiment analysis.

Cryptocurrency	Number of tweets before cleaning	Number of tweets after cleaning	Data loss due to cleaning	Number of tweets after majority voting	Data loss due to majority voting
BinanceCoin	1,759,614	1,745,908	0.78%	1,737,513	0.48%
Bitcoin	1,783,500	1,781,118	0.13%	1,766,382	0.83%
BitcoinCash	453,194	452,454	0.16%	449,936	0.56%
Cardano	1,328,099	1,327,521	0.04%	1,316,923	0.80%
Chainlink	1,005,240	970,112	3.49%	960,764	0.96%
Dai	28,048	27,857	0.68%	27,687	0.61%
Dash	74,455	73,763	0.93%	73,118	0.87%
Decentraland	297,174	295,351	0.61%	292,541	0.95%
EOS	109,743	109,100	0.59%	108,122	0.90%
EnjinCoin	100,388	98,553	1.83%	97,586	0.98%
Ethereum	1,813,200	1,808,472	0.26%	1,792,384	0.89%
EthereumClassic	1,808,520	1,808,202	0.02%	1,793,220	0.83%
Filecoin	123,793	122,986	0.65%	122,082	0.74%
IOTA	432,607	432,236	0.09%	428,595	0.84%
Litecoin	1,194,347	1,191,249	0.26%	1,181,627	0.81%
Maker	45,930	45,482	0.98%	45,110	0.82%
Monero	238,864	237,856	0.42%	236,445	0.59%
Neo	72,337	71,437	1.24%	70,897	0.76%
OMG_Network	67,426	66,556	1.29%	65,600	1.44%
Stellar	188,307	186,560	0.93%	184,843	0.92%
Tether	1,642,775	1,641,160	0.10%	1,635,580	0.34%
Tezos	10,493	10,475	0.17%	10,418	0.54%
Theta	163,307	160,940	1.45%	159,336	1.00%
Tron	17,907	17,903	0.02%	17,810	0.52%
Waves	78,217	76,308	2.44%	75,906	0.53%
XRP	1,819,418	1,817,155	0.12%	1,800,384	0.92%
Zcash	130,429	130,115	0.24%	129,239	0.67%
Total (average)	16,787,332	16,706,829	(0.74%)	16,580,048	(0.78%)

where:

- $\tilde{C}_t \in [0, 1]$, C_{min}, and C_{max} are the standardized, minimum, and maximum values of close price, respectively.
- $\tilde{S}_t \in [0, 1]$, S_{min}, and S_{max} are the standardized, minimum, and maximum values of sentiment scores, respectively.

To find the effect of input data length, we trained and evaluated models under three different values for the history parameter, that is, h. More specifically, we set $h = 7$, $h = 14$ and $h = 21$ as using the data from 1, 2 and 3 weeks before to predict today's close price. We fixed the parameters of training as Epoch = 25, Loss = 'mse', and Optimizer = 'adam'. To get more robust RMSE values and reduce the randomness effects in neural networks training, we repeated the training and evaluation of each model 30 times and the average of RMSE values over 30 runs was recorded as the RMSE value for each model. After obtaining RMSE values for each cryptocurrency, the average RMSE values over all cryptocurrencies were calculated. The results were ranked and presented as bar plots in Figure 6. In addition, the standard deviations of the RMSE values were shown in Figure 7.

By comparing the average RMSE values for three different history values in Figure 6, it can be seen that using data from 14 previous days for predicting today's close price has the lowest RMSE values on average. Moreover, by comparing the standard deviation of the RMSE values in Figure 7, we can see that the results of models under the setting of $h = 14$ are more stable because they show lowest values for the standard deviation on average. Therefore, from both aspects of accuracy and stability, using a history of 14 days of data is the best compared to using shorter or longer histories, that is, $h = 7$ and $h = 21$. Looking at the bar plot of the average RMSE values for the case of $h = 7$, we see that the LSTM(C) as the base model of this study outperforms all other 19 models. This means that our models do not perform well with short sequences of input data. However, in the case of $h = 14$, 17 out of 19 models outperform the LSTM(C) model.

Looking at the best five models in the case of $h = 14$, the effectiveness of adding the CNN layer in reducing the RMSE values is detectable. The effectiveness of having the attention layer in our models can be seen in the case of $h = 21$ where 4 out of the 5 best models have the attention layer in their structure. This indicates that the attention layer improves the performance of models when it is applied to longer sequences of input data. We could not make any general conclusions about the effectiveness of using sentiment scores in improvement of the RMSE values. We will discuss this effect in Section 5 where we look at the results on the cryptocurrency-level rather than considering the results that were achieved for all cryptocurrencies on average.

Although using $h = 14$ results in having more accurate and stable prediction values on average, the lowest RMSE value was achieved in the case of $h = 21$, related to the CNN(C,S)_LSTM_At(S) model with an average RMSE value of 268.5. While this model is the most stable one with a standard deviation of 1176.5 (See Figure 5, for $h = 21$), it does not perform well with shorter sequences of input data, especially for the case of $h = 14$, achieves the highest RMSE value on average.

It is worth mentioning that the goal of this study is not to find one unique best model that performs well for all cryptocurrencies and different lengths of input sequences. Due to the highly volatile nature of these markets, finding a model with high performance for all cryptocurrencies is almost impossible. We aim to investigate the different architectures of our proposed models and find the best architecture and input layer configuration for each cryptocurrency. Therefore, the above conclusions were made for the average RMSE values calculated over 27 cryptocurrencies. This means that the CNN(C,S)_LSTM_At(S) model is not the best model for all 27 cryptocurrencies. The next section discusses the results in more detail.

5 | DISCUSSION

Tables 8, 9 and 10 present the best model and its respective RMSE value that has been obtained for each cryptocurrency under different settings of h. In addition, considering the LSTM(C) as the base model, the RMSE reduction value from the base model to the best model has been reported in these tables. The results were sorted based on the RMSE reduction values in each table. Comparing the RMSE reduction values for $h = 7$, $h = 14$ and $h = 21$, we can see that in cases of $h = 21$ and $h = 14$, our models could improve the RMSE value of the base model more significantly compared to the case of $h = 7$. This was expected since in Section 4.3, we found out that our models underperform the base model when they are fed with short sequences of input data.

In each table, the models that are fed by sentiment scores (denoted by 'S') have been highlighted with green. Between 70% and 78% of the best models have the sentiment data as one of their input data either fed into the CNN or LSTM layer of the input layer. Therefore, by extracting people's emotions and sentiments from their posts on the Twitter platform and using them in prediction models, we can generate more accurate forecasts for cryptocurrencies' prices.

To make more robust comparisons, we employed a paired sample t-test for the results that are reported in Tables 8, 9 and 10. In a paired sample t-test, each row of the two samples is related to the same subject; in our case, two RMSE values correspond to the same cryptocurrency. Moreover, to increase the power of the test, a one-tailed version of the test was used. Before interpreting any test results, it should be noted that wherever the RMSE term is mentioned, it refers to the average of RMSE values that is calculated over the RMSE values of all 27 cryptocurrencies. We define the following hypotheses:

1. $H_o : \overline{RMSE}_{best_{model}} - \overline{RMSE}_{LSTM(C)} > 0$

   $H_a : \overline{RMSE}_{best_{model}} - \overline{RMSE}_{LSTM(C)} \leq 0$

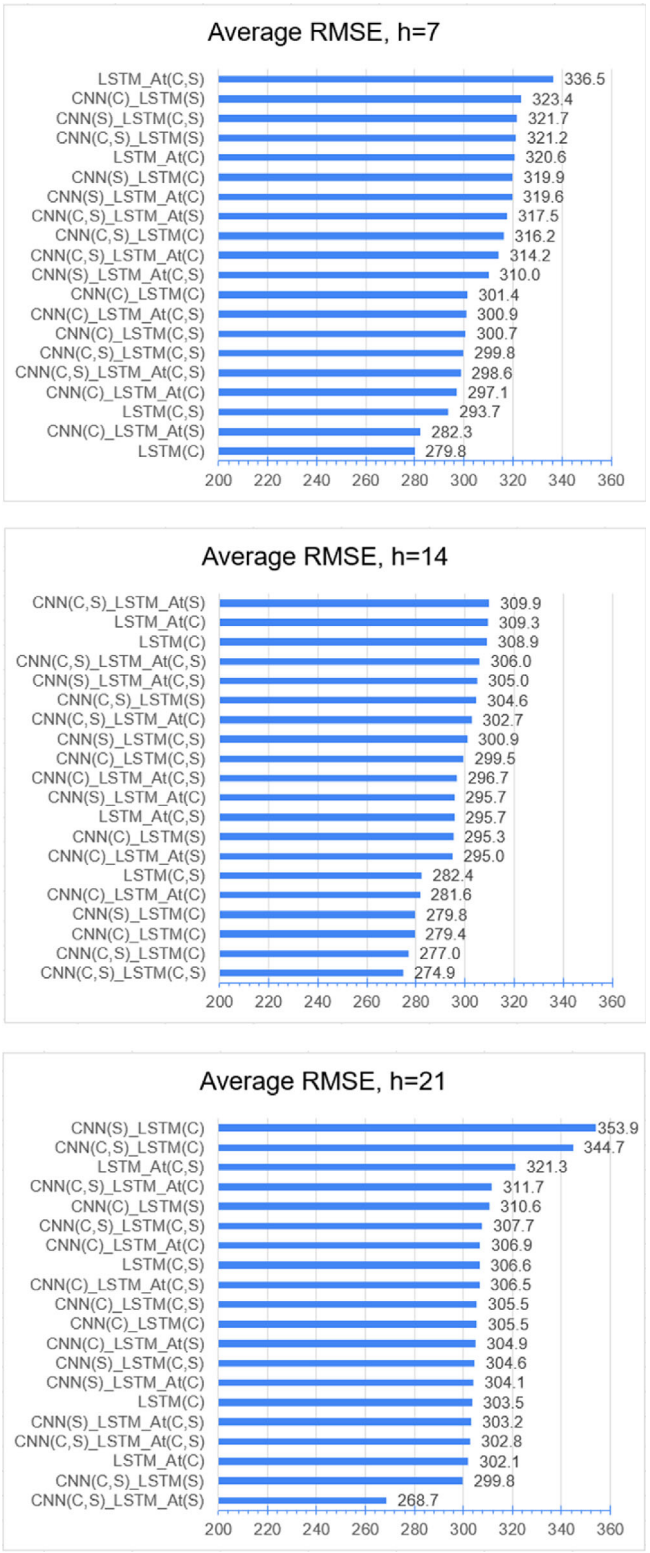

FIGURE 6 Average root mean squared error values for each model under different settings of $h = 7$, $h = 14$ and $h = 21$.

where $\overline{RMSE}_{best_{model}}$ is the average of RMSE value calculated over the RMSE values of the best models, and $\overline{RMSE}_{LSTM(C)}$ is the average RMSE value calculated over the RMSE values of the LSTM(C) model obtained for 27 cryptocurrencies.

The *p*-values reported in the second column of Table 11 are greater than 5% and consequently, the null hypothesis is not rejected. However, after taking a careful look at the RMSE values, we noticed that the RMSE values of Bitcoin and Ethereum are like

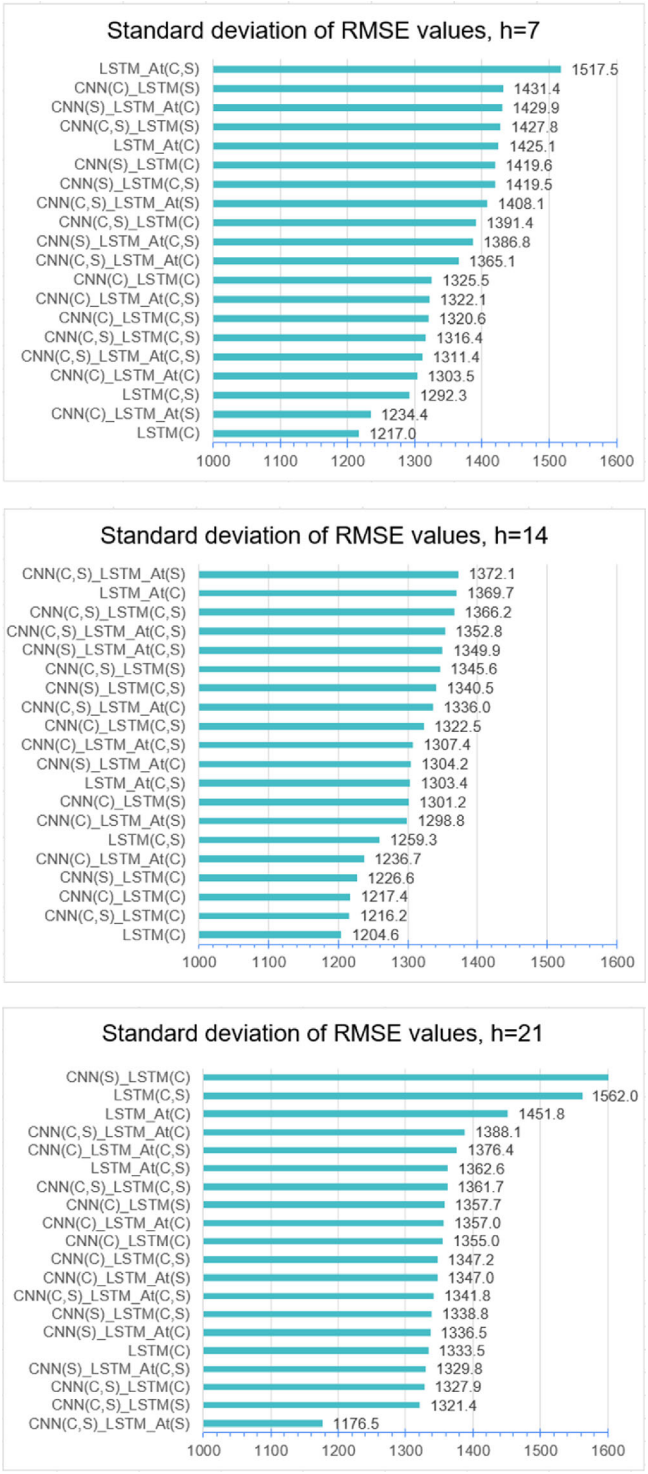

FIGURE 7 Standard deviation of root mean squared error values for each model under different settings of $h = 7$, $h = 14$ and $h = 21$.

outliers and hence, they affect the results of our *t*-test. Therefore, we removed the RMSE values of these cryptocurrencies from our *t*-test. As can be seen in the third and last columns of Table 11, the *p*-values got improved, which means that we can reject the null hypothesis on the level of 5% for all cases of *h*. Consequently, the best models have reduced the RMSE value of the base model (LSTM(C)) significantly.

We also defined the following hypotheses to compare the average RMSE values of the best models that we obtained for three different lengths of input sequences:

TABLE 8 The best models of each cryptocurrency with the respective RMSE ($h = 7$).

Cryptocurrency	Best model	RMSE	RMSE of base model	RMSE reduction
OMG_Network	CNN(S)_LSTM(C,S)	2.821	3.657	22.84%
Decentraland	CNN(S)_LSTM(C,S)	4.104	5.251	21.84%
Litecoin	CNN(C,S)_LSTM_At(C)	20.25	23.29	13.05%
Tron	CNN(S)_LSTM(C)	0.0092	0.0102	10.09%
EnjinCoin	CNN(S)_LSTM_At(C,S)	0.9544	1.058	9.76%
BinanceCoin	LSTM_At(C,S)	56.79	62.65	9.35%
Theta	CNN(S)_LSTM(C)	1.031	1.134	9.04%
Cardano	CNN(S)_LSTM(C)	0.2217	0.2436	8.99%
Dash	CNN(C,S)_LSTM(C,S)	29.54	32.37	8.75%
Stellar	CNN(S)_LSTM(C)	0.0399	0.04359	8.53%
Chainlink	CNN(S)_LSTM(C)	4.198	4.586	8.47%
Waves	CNN(C)_LSTM(S)	2.682	2.879	6.86%
Tether	CNN(C)_LSTM_At(C)	0.00030	0.00032	6.62%
Bitcoin	LSTM(C,S)	6337	6731	5.86%
EthereumClassic	CNN(C,S)_LSTM(C)	7.365	7.790	5.45%
Maker	CNN(C)_LSTM(S)	257.6	272.4	5.45%
XRP	CNN(C,S)_LSTM(S)	0.1023	0.1078	5.16%
Ethereum	LSTM_At(C)	561.89	591.4	4.99%
IOTA	CNN(C)_LSTM(C)	0.0997	0.1046	4.67%
Neo	LSTM_At(C)	8.584	8.908	3.65%
Dai	CNN(S)_LSTM_At(C)	0.0014	0.0014	3.14%
BitcoinCash	CNN(S)_LSTM(C,S)	93.69	96.22	2.63%
Zcash	CNN(C)_LSTM_At(S)	37.83	38.81	2.50%
Filecoin	CNN(S)_LSTM(C)	13.75	13.99	1.75%
Monero	LSTM_At(C)	30.90	31.36	1.49%
Tezos	CNN(C)_LSTM(C)	0.5710	0.5777	1.16%
EOS	LSTM(C)	0.8688	0.8688	0.00%
Average		276.8	293.7	

Note: The models that are fed by sentiment scores (denoted by 'S') have been highlighted with green.

2. $H_o : \overline{RMSE}_{h=14} - \overline{RMSE}_{h=7} > 0$

   $H_a : \overline{RMSE}_{h=14} - \overline{RMSE}_{h=7} \leq 0$

3. $H_o : \overline{RMSE}_{h=21} - \overline{RMSE}_{h=7} > 0$

   $H_a : \overline{RMSE}_{h=21} - \overline{RMSE}_{h=7} \leq 0$

4. $H_o : \overline{RMSE}_{h=21} - \overline{RMSE}_{h=14} > 0$

   $H_a : \overline{RMSE}_{h=21} - \overline{RMSE}_{h=14} \leq 0$

The *p*-values reported in Table 12 indicate that the null hypotheses in (2) and (3) are rejected on the significance level of 10%. Therefore, using input sequences with lengths of $h = 14$ and $h = 21$ reduces the RMSE values of the best models on average compared to input data with length of $h = 7$. The null hypothesis in (4) is not rejected and hence, there is no significant difference between the average value of RMSE achieved for the best models in cases of $h = 14$ and $h = 21$.

TABLE 9 The best models of each cryptocurrency with the respective RMSE ($h = 14$).

Cryptocurrency	Best model	RMSE	RMSE of base model	RMSE reduction
Decentraland	CNN(S)_LSTM(C,S)	3.275	5.413	39.49%
OMG_Network	LSTM(C,S)	2.594	3.379	23.24%
EnjinCoin	LSTM_At(C,S)	0.8526	1.054	19.11%
Cardano	CNN(S)_LSTM(C)	0.2132	0.2487	14.25%
Ethereum	LSTM(C,S)	469.3	546.9	14.19%
BinanceCoin	CNN(S)_LSTM_At(C)	54.42	62.86	13.43%
Bitcoin	CNN(S)_LSTM(C,S)	6275	6985	10.16%
Zcash	CNN(C,S)_LSTM(C)	34.80	38.59	9.81%
EthereumClassic	CNN(C)_LSTM(C)	7.270	7.934	8.37%
XRP	CNN(S)_LSTM(C)	0.0988	0.1064	7.15%
Theta	CNN(C,S)_LSTM(C,S)	1.090	1.1734	7.08%
Tron	CNN(S)_LSTM(C)	0.0085	0.0091	6.57%
Dai	CNN(S)_LSTM_At(C)	0.0015	0.0016	5.60%
Chainlink	CNN(S)_LSTM(C)	4.155	4.327	3.97%
Maker	CNN(S)_LSTM(C)	248.2	257.8	3.72%
BitcoinCash	LSTM_At(C,S)	89.46	92.22	2.99%
Neo	CNN(S)_LSTM_At(C,S)	8.660	8.911	2.82%
Stellar	LSTM_At(C)	0.0412	0.0422	2.28%
EOS	LSTM_At(C)	0.8197	0.8384	2.23%
Filecoin	CNN(C)_LSTM_At(S)	14.13	14.41	1.96%
Tether	CNN(C)_LSTM_At(C,S)	0.00031	0.00032	1.71%
Dash	LSTM(C,S)	31.37	31.84	1.46%
Waves	CNN(S)_LSTM(C)	2.835	2.873	1.34%
Monero	LSTM_At(C,S)	33.27	33.63	1.07%
IOTA	LSTM(C)	0.1064	0.1064	0.00%
Litecoin	LSTM(C)	23.96	23.96	0.00%
Tezos	LSTM(C)	0.5771	0.5771	0.00%
Average		270.6	300.9	

Note: The models that are fed by sentiment scores (denoted by 'S') have been highlighted with green.

6 | CONCLUSION

We fill a significant knowledge gap and make two main contributions to the literature of cryptocurrency price prediction: (1) from the methodological point of view, we developed a new hybrid model with a flexible input layer that can be customized for different cryptocurrencies, (2) from the experimental aspect, we analysed a large set of cryptocurrencies along with their related tweets posted between 1 January 2021 and 31 December 2021. We proposed a hybrid DL -based model for predicting cryptocurrencies daily close prices. Our model can have 20 different versions by accepting input data in different ways so that for each cryptocurrency, the best structure of the model is used to predict daily close prices. We evaluated our models using three values for the length of input data sequence, that is, using data from 7, 14 and 21 previous days to predict today's close price. The results of our experiments show that the longer sequences predict the close prices more accurately. In fact, we found that using a history of 14 days data results in more accurate predictions on average.

While adding the CNN layer was found to be more useful for the case of $h = 14$, the attention layer improved the prediction performance when longer sequences of input data were fed into the models, that is, $h = 21$. We concluded that for more than 70% of the cryptocurrencies, the use of sentiment data results in higher prediction performances. This indicates that social media posts and in particular, tweets play a significant role in people's decision-making for cryptocurrency market trading. Thus, this study can be a guide for researchers and academics to take into consideration the effect of appropriate social media data in cryptocurrency market prediction.

There are several limitations in our study. For example, the sentiment analysis methods struggle to understand irony, sarcasm and so forth, that are found in the tweets. This limits their availability in producing very accurate sentiments. In addition, there are unrelated contents in the

TABLE 10 The best models of each cryptocurrency with the respective RMSE ($h = 21$).

Cryptocurrency	Best model	RMSE	RMSE of base model	RMSE reduction
Decentraland	LSTM_At(C,S)	3.778	5.805	34.91%
OMG_Network	CNN(S)_LSTM(C)	2.292	3.125	26.68%
EnjinCoin	LSTM_At(C)	0.8953	1.109	19.28%
Bitcoin	LSTM_At(C,S)	6131	7566	18.97%
BitcoinCash	CNN(S)_LSTM(C,S)	87.53	101.7	13.96%
Cardano	CNN(S)_LSTM(C)	0.2208	0.2564	13.87%
Zcash	CNN(C)_LSTM_At(S)	33.73	39.02	13.56%
Waves	CNN(C)_LSTM_At(C)	2.812	3.239	13.20%
BinanceCoin	CNN(C)_LSTM(S)	56.90	64.97	12.43%
Neo	LSTM_At(C,S)	7.866	8.744	10.04%
Tron	CNN(C)_LSTM_At(C,S)	0.0086	0.0095	9.49%
Ethereum	LSTM(C,S)	465.7	512.9	9.20%
Dai	CNN(S)_LSTM_At(C)	0.0014	0.0016	8.99%
EOS	CNN(S)_LSTM_At(C)	0.8487	0.9166	7.41%
Tether	CNN(C,S)_LSTM_At(C)	0.00032	0.00035	7.07%
Monero	LSTM_At(C)	29.00	31.04	6.57%
IOTA	CNN(C,S)_LSTM(C,S)	0.1003	0.1056	5.05%
Dash	LSTM_At(C)	29.68	31.04	4.36%
Stellar	CNN(S)_LSTM_At(C)	0.0371	0.0384	3.44%
XRP	CNN(C,S)_LSTM(S)	0.1019	0.1055	3.40%
Filecoin	LSTM_At(C)	13.82	14.20	2.69%
Tezos	CNN(C)_LSTM(S)	0.5600	0.5743	2.49%
EthereumClassic	LSTM(C,S)	7.321	7.497	2.35%
Theta	LSTM(C,S)	0.9804	1.002	2.20%
Maker	LSTM(C,S)	253.7	256.9	1.25%
Chainlink	LSTM(C)	3.870	3.870	0.00%
Litecoin	LSTM(C)	21.15	21.15	0.00%
Average		264.9	321.3	

Note: The models that are fed by sentiment scores (denoted by 'S') have been highlighted with green.

TABLE 11 *p*-values of *t*-tests: comparing RMSE of the best models with RMSE of the base model, that is, LSTM(C).

	p-value with all cryptocurrencies	*p*-value after removing bitcoin	*p*-value after removing bitcoin and Ethereum
$h = 7$	0.1272	0.0292**	0.0190**
$h = 14$	0.1300	0.0883*	0.0132**
$h = 21$	0.1490	0.0432**	0.0121**

Note: ** and * indicate a rejection of null hypothesis at the 5% and 10% of significance levels.

TABLE 12 *p*-values of *t*-tests: comparing average RMSE of the best models achieved for different values of *h*.

	RMSE_7	RMSE_14	RMSE_21
RMSE_7	—	0.0703[a]	0.0826[a]
RMSE_14		—	0.1369
RMSE_21			—

[a]Indicates a rejection of null hypothesis at the 10% of significance level.

tweets such as promotions, spams and so forth, and this also prevents us to extract more meaningful sentiments from tweets. Some researchers believe that tweets that are generated by bots should be removed from sentiment analysis. For the future research, we would like to develop a bot detection classifier and apply it on our Twitter data to remove tweets that are generated by bots. Then, we will find out if removing the tweets generated by bots could or could not improve the prediction performance of our models.

DATA AVAILABILITY STATEMENT

Research data are not shared.

ORCID

Salim Lahmiri https://orcid.org/0000-0002-9237-4100

REFERENCES

Anbaee Farimani, S., Vafaei Jahan, M., Milani Fard, A., & Tabbakh, S. R. K. (2022). Investigating the informativeness of technical indicators and news sentiment in financial market price prediction. *Knowledge-Based Systems*, *247*, 108742. https://doi.org/10.1016/j.knosys.2022.108742

Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. http://arxiv.org/abs/1908.10063.

Barbieri, F., Camacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). TweetEval: Unified benchmark and comparative evaluation for tweet classification. http://arxiv.org/abs/2010.12421.

Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A pretrained language model for scientific text. http://arxiv.org/abs/1903.10676.

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. https://doi.org/10.1109/72.279181

Chang, V., Baudier, P., Zhang, H., Xu, Q., Zhang, J., & Arami, M. (2020). How blockchain can impact financial services – The overview, challenges and recommendations from expert interviewees. *Technological Forecasting and Social Change*, *158*, 120166. https://doi.org/10.1016/j.techfore.2020.120166

Chang, V., Gagnon, S., Valverde, R., & Ramachandran, M. (2021). Emerging trends and impacts of the rise of AI, data analytics and blockchain. *Journal of Enterprise Information Management*, *34*(5), 1277–1286. https://doi.org/10.1108/JEIM-09-2021-555

Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting*, *37*(1), 28–43. https://doi.org/10.1016/j.ijforecast.2020.02.008

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. http://arxiv.org/abs/1810.04805.

Hiew, J. Z. G., Huang, X., Mou, H., Li, D., Wu, Q., & Xu, Y. (2019). BERT-based financial sentiment index and LSTM-based stock return predictability. http://arxiv.org/abs/1906.09024.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, K., Altosaar, J., & Ranganath, R. (2019). ClinicalBERT: Modeling clinical notes and predicting hospital readmission. http://arxiv.org/abs/1904.05342.

Hutto, C. J., & Gilbert, E. (2014). VADER: A parsimonious rule-based model for sentiment analysis of social media text. http://sentic.net/.

Ji, S., Kim, J., & Im, H. (2019). A comparative study of bitcoin price prediction using deep learning. *Mathematics*, *7*(10). https://doi.org/10.3390/math7100898

Kim, H. M., Bock, G. W., & Lee, G. (2021). Predicting Ethereum prices with machine learning based on blockchain information. *Expert Systems with Applications*, *184*, 115480. https://doi.org/10.1016/j.eswa.2021.115480

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). D convolutional neural networks and applications-a survey. *Mechanical Systems and Signal Processing*, *151*, 107398. https://doi.org/10.1016/j.ymssp.2020.107398

Kraaijeveld, O., & de Smedt, J. (2020). The predictive power of public twitter sentiment for forecasting cryptocurrency prices. *Journal of International Financial Markets, Institutions and Money*, *65*, 101188. https://doi.org/10.1016/j.intfin.2020.101188

Lahmiri, S., & Bekiros, S. (2019a). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons & Fractals*, *118*, 35–40. https://doi.org/10.1016/j.chaos.2018.11.014

Lahmiri, S., & Bekiros, S. (2019b). Deep learning forecasting in cryptocurrency high-frequency trading. *Cognitive Computation*, *13*, 485–487. https://doi.org/10.1007/s12559-021-09841-w

Lahmiri, S., & Bekiros, S. (2020). Intelligent forecasting with machine learning trading systems in chaotic intraday bitcoin market. *Chaos, Solitons and Fractals*, *133*, 109641. https://doi.org/10.1016/j.chaos.2020.109641

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*, 2278–2324. https://doi.org/10.1109/5.726791.

Loureiro, D., Barbieri, F., Neves, L., Anke, L. E., & Camacho-Collados, J. (2022). TimeLMs: Diachronic language models from Twitter. http://arxiv.org/abs/2202.03829.

Malo, P., Sinha, A., Takala, P., Korhonen, P., & Wallenius, J. (2013). Good debt or bad debt: Detecting semantic orientations in economic texts. http://arxiv.org/abs/1307.5336.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. www.bitcoin.org.

Ortu, M., Uras, N., Conversano, C., Bartolucci, S., & Destefanis, G. (2022). On technical trading and social media indicators for cryptocurrency price classification through deep learning. *Expert Systems with Applications*, *198*, 116804. https://doi.org/10.1016/j.eswa.2022.116804

Oyedele, A. A., Ajayi, A. O., Oyedele, L. O., Bello, S. A., & Jimoh, K. O. (2023). Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. *Expert Systems with Applications*, *213*, 119233. https://doi.org/10.1016/j.eswa.2022.119233

Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020). A deep learning-based cryptocurrency Price prediction scheme for financial institutions. *Journal of Information Security and Applications*, *55*, 102583. https://doi.org/10.1016/j.jisa.2020.102583

Pérez, J. M., Furman, D. A., Alemany, L. A., & Luque, F. (2021). RoBERTuito: A pre-trained language model for social media text in Spanish. http://arxiv.org/abs/2111.09453.

Pérez-Enciso, M., & Zingaretti, L. M. (2019). A guide for using deep learning for complex trait genomic prediction. *Genes*, *10*. https://doi.org/10.3390/genes10070553

Quoc Nguyen, D., Vu, T., Tuan Nguyen, A., & Research, V. (2020). BERTweet: A pre-trained language model for English Tweets. https://pypi.org/project/emoji.

Rosenthal, S., Farra, N., & Nakov, P. (2017). SemEval-2017 task 4: Sentiment analysis in twitter. https://trends24.in/.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. http://nlp.stanford.edu/.

Sun, X., Liu, M., & Sima, Z. (2020). A novel cryptocurrency price trend forecasting model based on LightGBM. *Finance Research Letters*, *32*, 101084. https://doi.org/10.1016/j.frl.2018.12.032

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. http://arxiv.org/abs/1706.03762.

Zhang, Z., Dai, H. N., Zhou, J., Mondal, S. K., García, M. M., & Wang, H. (2021). Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels. *Expert Systems with Applications*, *183*, 115378. https://doi.org/10.1016/j.eswa.2021.115378

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. http://arxiv.org/abs/1506.06724.

Zou, Y., & Herremans, D. (2022). A multimodal model with twitter FinBERT embeddings for extreme price movement prediction of bitcoin. http://arxiv.org/abs/2206.00648.

Zoumpekas, T., Houstis, E., & Vavalis, M. (2020). ETH analysis and predictions utilizing deep learning. *Expert Systems with Applications*, *162*, 113866. https://doi.org/10.1016/j.eswa.2020.113866

AUTHOR BIOGRAPHIES

Bahareh Amirshahi is a PhD candidate and lecturer in the department of supply chain and business technology management at John Molson School of Business, Concordia University, Montreal, Canada. Her research interests are in applications of machine learning, deep learning, and econometrics in business and economics. In particular, she designs and develops novel prediction models using machine learning techniques for forecasting financial time series such as stocks and cryptocurrencies.

Salim Lahmiri is associate professor of artificial intelligence and data science in the department of supply chain and business technology management, John Molson School of Business, Concordia University, Montreal, Canada. He holds a M. Eng degree from the department of electrical engineering of Ecole de Technologie Superieure (ETS), Montreal. Also, he holds a PhD degree from the department of computer science of the University of Quebec at Montreal (UQAM), Canada. Prior to join Concordia University, he was a post-doc fellow in the MNI at McGill University. His research interests are in the design of intelligent systems with applications in business, economics, biomedical engineering, and healthcare.

How to cite this article: Amirshahi, B., & Lahmiri, S. (2023). Investigating the effectiveness of Twitter sentiment in cryptocurrency close price prediction by using deep learning. *Expert Systems*, e13428. https://doi.org/10.1111/exsy.13428