

Article

Stochastic Neural Networks-Based Algorithmic Trading for the Cryptocurrency Market

Vasu Kalariya ¹, Pushpendra Parmar ¹, Patel Jay ¹, Sudeep Tanwar ^{1,*}, Maria Simona Raboaca ^{2,*},
Fayez Alqahtani ³, Amr Tolba ⁴ and Bogdan-Constantin Neagu ⁵

¹ Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, Gujarat, India; 17bce131@nirmauni.ac.in (V.K.); 17bce098@nirmauni.ac.in (P.P.); 17bce077@nirmauni.ac.in (P.J.)

² National Research and Development Institute for Cryogenic and Isotopic Technologies—ICSI Rm. Valcea, Uzinei Street, No. 4, 240050 Ramnicu Valcea, Romania

³ Software Engineering Department, College of Computer and Information Sciences, King Saud University, Riyadh 12372, Saudi Arabia; fhalqahtani@ksu.edu.sa

⁴ Computer Science Department, Community College, King Saud University, Riyadh 11437, Saudi Arabia; atolba@ksu.edu.sa

⁵ Department of Power Engineering, “Gheorghe Asachi” Technical University of Iasi, 700050 Iasi, Romania; bogdan.neagu@tuiasi.ro

* Correspondence: sudeep.tanwar@nirmauni.ac.in (S.T.); simona.raboaca@icsi.ro (M.S.R.)

Abstract: Throughout the history of modern finance, very few financial instruments have been as strikingly volatile as cryptocurrencies. The long-term prospects of cryptocurrencies remain uncertain; however, taking advantage of recent advances in neural networks and volatility, we show that the trading algorithms reinforced by short-term price predictions are bankable. Traditional trading algorithms and indicators are often based on mean reversal strategies that do not advantage price predictions. Furthermore, deterministic models cannot capture market volatility even after incorporating price predictions. Thus motivated by these issues, we integrate randomness in the price prediction models to simulate stochastic behavior. This paper proposes hybrid trading strategies that take advantage of the traditional mean reversal strategies alongside robust price predictions from stochastic neural networks. We trained stochastic neural networks to predict prices based on market data and social sentiment. The backtesting was conducted on three cryptocurrencies: Bitcoin, Ethereum, and Litecoin, for over 600 days from August 2017 to December 2019. We show that the proposed trading algorithms are better when compared to the traditional buy and hold strategy in terms of both stability and returns.

Keywords: Bollinger bands; pairs trading; Awesome Oscillator; stochastic neural networks; cryptocurrency

MSC: 68T07



Citation: Kalariya, V.; Parmar, P.; Jay, P.; Tanwar, S.; Raboaca, M.S.; Alqahtani, F.; Tolba, A.; Neagu, B.-C. Stochastic Neural Networks-Based Algorithmic Trading for the Cryptocurrency Market. *Mathematics* **2022**, *10*, 1456. <https://doi.org/10.3390/math10091456>

Academic Editors: Haitao Li, June Feng, Ben Niu and Shihua Fu

Received: 27 March 2022

Accepted: 25 April 2022

Published: 26 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the advancements in computing technology and networking, we have seen a steep increase in digital currencies and the declining use of traditional currencies. When selecting a currency for trading, the most important factor is the currency's value. The value of a traditional currency is determined by the market, which infers it from the stability and strength of a central governing agency. However, mismanagement of financial systems by such central agencies can cause currency devaluation [1]. Digital currencies, in contrast, are free from such detrimental influences, and thus their value is solely dependent on the consistency and security of the underlying technology. But all that shines is not gold, and digital currencies, too, have their flaws in the form of double-spending and transaction data manipulation.

Blockchain technology can be used to overcome the aforementioned issues in the

current cryptocurrency systems. It stores data in an unalterable manner to establish decentralization, which in turn provides security, trust, and transparency among peers in the network [2–4]. Cryptocurrencies use complex cryptographical ciphers to conduct financial transactions that make the system robust against corruption, unlike in fiat currencies. It also tackles the double-spending problem because of the peers' verification. As the affirmation about a transaction increases from the peers, a quorum is established and the transaction becomes immutable.

As we discussed earlier, cryptocurrencies are not backed by any single institution. Therefore, their value heavily relies on public opinion and how people see them as assets [5,6]. People tend to share their beliefs and sentiments over social media platforms and, therefore, such information can be utilized by machine learning algorithms to forecast the price of cryptocurrencies [7–10]. McCoy et al. [11] used English and Japanese tweets regarding cryptocurrencies to create price predictors. They showed that inculcating tweet analysis in their predictors helped them generate decent profits. Various researchers have explored several techniques to trade commodities. However, very little research work has been carried out in algorithmic trading of cryptocurrencies with solutions that integrate trading as well as tackling high market volatility [12].

Traditional financial assets are very stable when compared to cryptocurrencies. Thus, investors can easily deduce the expected value of traditional assets like gold, bonds, and even the equities market. From Figure 1 it is clear that the price volatility of Bitcoin is exceeding high when compared to gold and the S&P 500. One of the main reasons for such high volatility is the uncertainty surrounding the expected future value, government regulations, and social sentiment [13,14]. Mokni et al. [13] examined the effects of economic policy uncertainty on the volatility of Bitcoin under various market conditions. They showed that casualty in quantiles test outlines the remarkable correlation between economic policy uncertainty and Bitcoin. Liu et al. [14] showed that there are strong linkages between cryptocurrency and financial markets in countries whose governments recognize cryptocurrencies as a medium of exchange. Although previous research uses neural networks to predict the prices of cryptocurrencies, the issue of excessively high volatility is not tackled. This paper shows how to address volatility by using stochasticity in neural networks. Hence, we show how decisions can be made under high uncertainty by incorporating stochastic neural networks in trading algorithms.

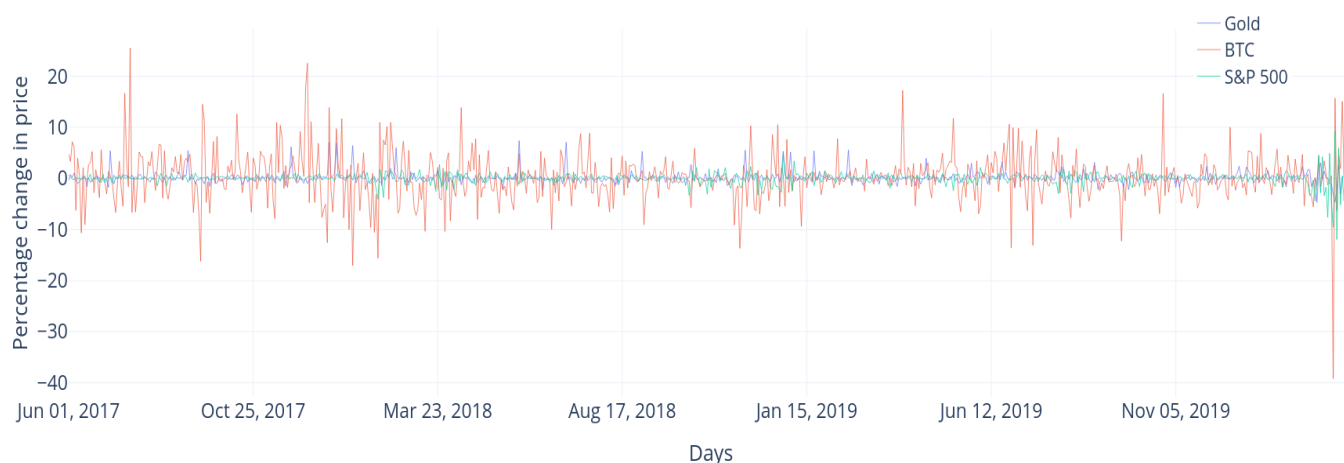


Figure 1. Percentage change in price vs. days for Bitcoin, gold, and the S&P 500.

1.1. Motivation

Cryptocurrencies have been predominantly used as a means of transaction. However, in the previous decade, there has been a significant increase in investors who believe that cryptocurrencies are going to be widely adopted [12]. According to the compound annual growth rate reports, by the end of 2024, the market capitalization of cryptocurrencies is forecasted to have an expansion of 11.9% [15]. For the time being, the cryptocurrency

market is a highly volatile market due to a substantial dependence on social sentiment to evaluate the market's worth [16]. The value of major cryptocurrencies relies on many factors like public opinion, government legislature [13,17], volatility, and trustworthiness. Bouri et al. [17] used a Google-trends-based metric to gauge the US–China trade war, showing that this metric helped improve the accuracy for different kinds of random forest models. To effectively trade cryptocurrencies, market timing is more important than asset allocation analysis done in traditional strategies. This is because the entry and exit signals of trades rely on the market's perception at any given time. Motivated by the potential to create value by taking advantage of inefficiencies in social sentiment, we present a framework for trading cryptocurrencies using algorithmic trading strategies propelled by robust predictions from stochastic neural networks.

1.2. Contribution

Research contributions of this paper are as follows:

- We propose a technique to embed future prices in current statistical-based trading algorithms
- We use stochastic neural networks based on price prediction, which is then pipelined with proposed trading algorithms.
- We present mechanisms for integrating price prediction from stochastic neural networks with well-known trading algorithms.

1.3. Organization

The paper is organized in the following manner. Section 2 describes the previous work in the field of machine-learning-based trading. Section 3 illustrates the background of stochastic neural networks and their application in price prediction. In Section 4, we state the mathematical formulation of the trading algorithms and their experimental details. Section 5 discusses the backtesting results of trading algorithms, and, finally, Section 6 concludes the paper.

2. Related Work

Garcia et al. [18] performed an analysis of Bitcoin, a highly volatile cryptocurrency. Their analysis incorporated market information such as transaction volume and prices. They also used social signals such as word-of-mouth volume, opinion polarization in tweets, and emotional valence over three years. They concluded that opinion polarization was the most relevant factor determining the worth of the currency, closely followed by exchange volumes and emotional valence. Based on their analysis, they devised a trading strategy that provided a 0.3% return per day.

Over the last couple of years, there has been a tremendous improvement in reinforcement learning algorithms with accurate deep learning systems. Deng et al. [19] used a deep recurrent neural network to analyze financial signals. They combined deep learning and reinforcement learning to create a robust algorithmic trading system. A recurrent neural network senses the dynamic market conditions for important abstract information. The reinforcement learning system then tries to make trading decisions based on the neural network's output. They verified the effectiveness of their system in the stock and commodity futures markets.

Nakona et al. [20] simulated intraday automated trading using artificial neural networks. Using 15 min of periodic market data, they analyzed market signals. Considering the costs of transaction executions, they demonstrated that their approach significantly supersedes the performance of a buy-and-hold strategy. They also performed a sensitivity analysis to verify the robustness of their approach. Furthermore, they devised strategies that included short positions by taking advantage of the futures market, CBOE Futures Exchange.

Fil et al. [21] proposed a pairs trading strategy based on the mean-reversion in prices. The cointegration method was applied to 26 different cryptocurrencies traded on 5 min, 1 h, and daily frequencies. They observed that high-frequency trading was more profitable when

compared to short-term and long-term holding strategies. The most common daily distance method returned -0.07% monthly. This increased to 11.61% monthly for a 5-min frequency. The Sharpe ratios for their 5 min, 1 h, and daily trading models were 19, 5.1, and 1.1.

Sentiment analysis has been widely used to estimate public opinion. Colianni et al. [22] proposed supervised learning algorithms to conduct sentiment analysis on BTC based on Twitter data using tweepy on the keyword search. Their approach led to a final hour-to-hour and day-to-day prediction accuracy exceeding 90%. They showed that public sentiment has a statistical significance in determining the day-to-day value of cryptocurrencies. Fister et al. [23] proposed an LSTM-based network for automated trading, and the proposed approach was compared with passive and rule-based trading strategies as well as with machine learning classifiers. The LSTM approach outperformed other trading strategies on blue-chip stocks like BMW when backtested on 2010–2018 data. They stated that stock markets implicitly consolidate long-term dependencies, which must be considered before selecting a trading strategy. To empirically assess the effectiveness of each trading strategy in highly volatile markets, they developed a mechanical trading system to enable fair evaluation of the strategies.

Energy-based models have succeeded in tackling recommendation tasks, with the increasing ability to run Monte-Carlo model-based simulations easily. Takeuchi et al. [24] proposed an autoencoder model consisting of stacked restricted Boltzmann machines, which were used to extract significant features from the historical market data. Their model discovered the momentum effect in stock prices. They devised a trading strategy that brought in annualized returns of 45.93% when backtested in stock markets from 1990 to 2009. Yong et al. [25] used a deep neural network to anticipate the index price of the Singapore exchange using the FTSE Straits Time Index. These forecasted results were used along with specified entry and exit trade rules. The model was evaluated using RMSE and MAPE, having a profit factor of 18.67 and a Sharpe ratio of 5.34 on simulation. The overall profit margin on the trades was observed to be 70.83% .

Zanc et al. [26] used a stacked LSTM based forecasting module consisting of 256 hidden units on a forex dataset. The model used Orderbook to represent the volume of bids and asks on the previous 100 transactions. The author extracted features using CNN and combined them with the daily price to generate multiple time scale features. Three LSTM units were used to learn the time-dependent multiple time scale features obtained from the previous step. Stephen et al. [27] used noise correlated stochastic differential equations employing geometric Brownian motion for modeling cryptocurrency rates and the geometric Ornstein–Uhlenbeck (GOU) process for modeling social media activities and trading volume of cryptocurrencies. The authors introduced stochastic variables in the stochastic differential equations and modeled social media activities and the daily trading volume of cryptocurrencies as the GOU process. The authors tested the models on simulated as well as real-world environments to test the robustness of their model.

Wong et al. [28] showed that the rate of convergence to the global minimum for their stochastic neural network (diffusion machine) is better than the rate of convergence of Boltzmann machines with the setup of three coupled diffusion machines, each performing one of the following activities: training, learning, and weigh-adjustment—all occurring concurrently. Florensa et al. [29] proposed a stochastic neural network that is composed of stochastic units and a feedforward neural network. Their method of integration is multiplicative integration. They used this kind of architecture as a building block to learning a diverse set of skills to perform reinforcement learning. They concluded that using the stochastic neural network can significantly improve the set of skills that a model acquires. Table 1 shows the relative comparison of state-of-the-art work in the field of trading of cryptocurrencies with the proposed approach.

Table 1. A relative comparison of recent work in the field of trading of cryptocurrencies.

Author	Year	Description	Parameters	Results
Garcia et al. [18]	2015	In their analysis, they incorporate market information such as transaction volume and prices. They also used social signals such as word-of-mouth volume, opinion polarization in tweets, and emotional valence over three years. They concluded that opinion polarization was the most relevant factor determining the worth of the currency, closely followed by exchange volumes and emotional valence.	Bitcoin market information and its social and economic signals	Author, along with market information like volume and prices, trained a model that provides a return of 0.3% per day.
Deng et al. [19]	2016	Author uses two biological-related learning concepts of deep learning (DL) and reinforcement learning (RL). DL model senses dynamic market conditions to important abstract information and the RL model tries to make a trading decision based on the output of DL.	Not provided	They verified the computational effectiveness of their model.
Nakona et al. [20]	2018	Used 7 layered ANN with ReLU and LeakyReLU to predict intraday price of bitcoin. They also perform a sensitivity analysis to verify the robustness of their approach.	Bitcoin data with the following features: volume, closing price, high price, and low prices over 15 min	Their best performing ANN model increased asset value by 64.46 times. Their Sharpe ratio for the best performing strategy is 24,601.06.
Fil et al. [21]	2020	They applied the distance and cointegration methods to a basket of 26 liquid cryptocurrencies traded on the Binance exchange, specifically at 5-min, 1-h, and daily frequencies. They observed that high-frequency trading was more profitable when compared to short-term and long-term holding strategies.	They used the Binance exchange data of 26 cryptocurrencies.	The most common daily distance method returns −0.07% monthly. This increases to 11.61% monthly for a 5-min frequency.
Colianni et al. [22]	2015	Authors proposed supervised learning algorithms to conduct sentiment analysis on BTC based on Twitter data using tweepy on the keyword search. They show that public sentiment has a statistical significance in determining the day-to-day value of cryptocurrencies.	Tweets related to Bitcoin were fetched from Twitter.	Their approach leads to a final hour-to-hour and day-to-day prediction accuracy exceeding 90%.
Fister et al. [23]	2019	Author used LSTM based network for automated trading and the proposed approach was compared with passive and rule-based trading strategies as well as with machine learning classifier.	Yahoo finance dataset	LSTM peaks for 2-day stock return at 190.2% of overall returns.
Takeuchi et al. [24]	2013	Author proposed an autoencoder model consisting of stacked restricted Boltzmann machines, which were used to extract significant features from the historical market data. Their model discovers the momentum effect in stock prices.	Ordinary shares traded on NYSE, AMEX, and NASDAQ from 1965 to 1989	They devised a trading strategy that brought in annualized returns of 45.93% for basic momentum when backtested in stock markets from 1990 to 2009.
Yong et al. [25]	2017	Authors used a deep neural network to anticipate the index price of the Singapore exchange using the FTSE Straits Time Index. These forecasted results were used along with specified entry and exit trade rules.	Singapore exchange	The model was evaluated using RMSE and MAPE, having a profit factor of 18.67 and a Sharpe ratio of 5.34 on simulation.
Proposed approach	2022	This paper proposes hybrid trading strategies that take advantage of the traditional mean reversal strategies alongside robust price predictions from stochastic neural networks. We trained stochastic neural networks to predict prices based on market data and social sentiment. The backtesting was conducted on three cryptocurrencies: Bitcoin, Ethereum, and Litecoin, for over 600 days from August 2017 to December 2019.	Market information, Google Trends and Tweet volume of Bitcoin, Ethereum, and Litecoin	SNN-enhanced Bollinger bands trading algorithm was the best strategy giving an annualized returns of 241.44%, 293.89%, and 332.65% for BTC, ETH, and LTC, respectively.

3. Background: Stochastic Neural Networks

The following section illustrates the problem of high market volatility and how it affects the process of predicting the prices of various cryptocurrencies. First, we present stochastic processes with price irregularities in the cryptocurrency market. Furthermore,

we introduce the integration mechanism of randomness in the activation of neural networks to simulate market volatility and propose the architecture and mathematical formulations for the described mechanism. Finally, a general stochastic module that various layers can use in neural networks is presented, along with its use in forwarding feed networks.

3.1. The Need for Stochasticity

The variables that influence market assets price fluctuations, such as financial outlook, inflation rate, policy and regulation, and human psychology, are all unstable display behaviour. Ref. [30], which, when aggregated, cannot be comprehended by us based on the presently available information. Generally, all processes can be considered deterministic, given the complete information set. The cryptocurrency market is not inherently stochastic. However, because all of the essential information on the asset is not accessible, it becomes sufficiently ungraspable for our systems. Thus, deterministic approaches are not suitable for this problem, and therefore we need a stochastic perspective. We are defining the system as erratic benefits from modelling it using stochasticity.

3.2. Stochasticity in Neural Networks

To model the stochastic behavior in trading, we incorporate stochastic neural networks in the traditional trading algorithms. Before incorporating the stochastic factor, we need a method to extract important market information and derive interdependencies between market statistics and social tendencies. Neural networks are remarkably suitable and widely used function approximators to find the dependencies between the variables. There are two means to incorporate randomness into neural networks. To get random fluctuations, the initial one is to alter the weights to a minuscule extent. The latter one is to add randomness into the activations during runtime. The first method is not preferable, as it directly perturbs the neural network's weights. This perturbation in the weights of a neural network can affect its learning and, also, the error would gradually increase as we evaluate it. The latter seems more suitable because the random changes would not directly affect the weights and hence only minute changes would come through the activations.

In our earlier work [31], we developed the following formulation that induces stochasticity in the layers of deep neural networks:

$$s_t = h_t + \gamma \xi_t \times \text{reaction}(h_t, s_{t-1}), 0 \leq \gamma \leq 1 \quad (1)$$

where h_i represents the activations of the layer at the i th time step; γ is defined as the perturbation factor that regulates the magnitude of stochasticity induced in the layer; ξ is a random (independently and identically distributed) variable that injects stochasticity in the activations; and *reaction* is a user-defined operation that calculates how activations of the previous timestep will transform current activations. Finally, the layer outputs the activation value s_i , which is passed on to the next module in the pipeline.

The financial market follows the Markov property that the conditional value of the price in the future time step depends only upon the price at the current time step [32]. Koki et al. [32] demonstrated that hidden Markov models often improved forecasts on Bitcoin, Ethereum, and Ripple. Thus, these findings bolster the fact that the financial and cryptocurrency markets follow the Markov property. Therefore, only two parameters h_t and s_{t-1} were used for determining the reaction function. This implies that the prior distribution of s_{t-1} and the current distribution s_t are independent of other past distributions. Therefore, we model the *reaction* function as the difference between current and previous activations:

$$\text{reaction}(h_t, s_{t-1}) = h_t - s_{t-1}. \quad (2)$$

Therefore, from Equations (1) and (2),

$$s_t = h_t + \gamma \xi_t (h_t - s_{t-1}). \quad (3)$$

Equation (3) is essentially a random walk amplified by the market reaction at each time step.

Figure 2 illustrates components of the stochastic module. The inputs to the *reaction* component are the feature activations of the current timestep (h_t) and stochastic activations of the previous timestep (s_{t-1}), thus creating a bridge between two consecutive timesteps. This is the most crucial component of the stochastic module because it is responsible for modelling the market's short-term movement. The *reaction* function itself can be an elaborate neural network. However, experimental results show that even a subtraction operation can give effective results [31] and, therefore, we have used it in our experiments. Figure 3 describes the flow of stochastic activations across timesteps via a stochastic module in a standard multi-layer perceptron (MLP). The stochastic module can also be similarly integrated with different variations of neural networks like LSTM, etc.

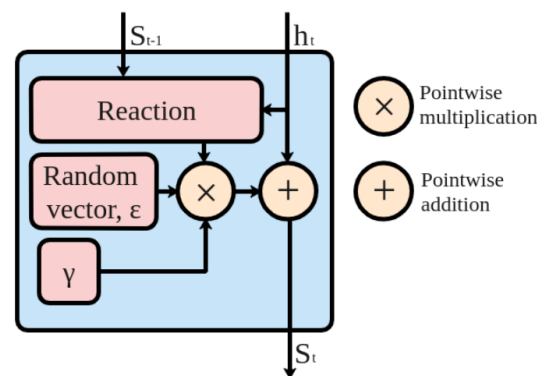


Figure 2. Stochastic module.

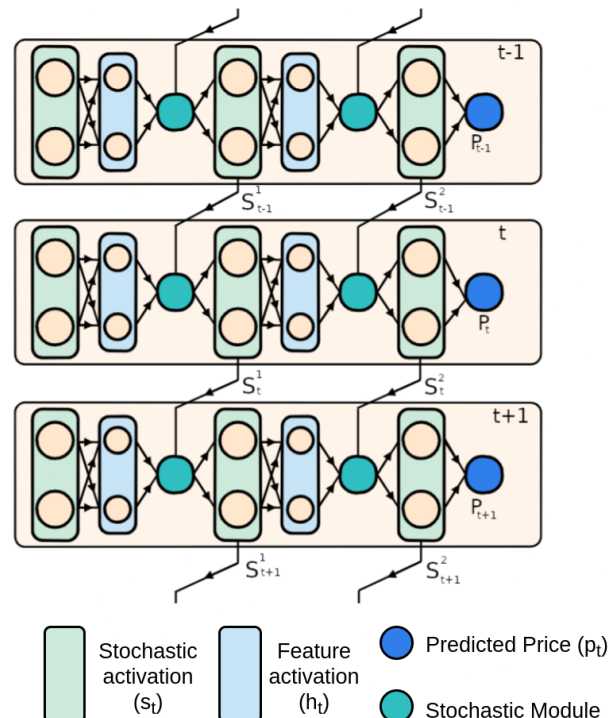


Figure 3. Unrolled architecture of stochastic neural network over consecutive timesteps.

Algorithm 1 describes the integration of the stochastic module with a multi-layer perceptron where input features X_t at time t is passed to the network. A *reaction* vector is calculated from values computed from the previous time step at every layer.

Algorithm 1 Forward propagation in a stochastic neural network.**Input:** X , Market indicators and cryptocurrencies data.**Input:** Model architecture having l hidden layers.**Input:** Trained $W^i, b^i, i \in \{1, \dots, l\}$.**Output:** $P_t, t \in \{1, \dots, n\}$.**Forward Propagation:**

```

for  $t = 1$  to  $n$  do
   $s_t^0 = X_t$ 
  for  $k = 1$  to  $l$  do
     $z_t^k = b_t^k + W_t^k s_t^{k-1}$ 
     $h_t^k = \text{activation}(z_t^k)$ 
     $r \leftarrow$  random variable vector,  $\xi_t^k$ 
     $s_t^k = h_t^k + \gamma \times r \odot \text{reaction}(h_t^k, s_t^{k-1})$ 
  end for
  Predicted price  $P_t \leftarrow s_t^l$ 
end for

```

4. Algorithmic Trading

The advantage of using price prediction with deep learning on statistical trading algorithms is that the traditional algorithms invoke buy and sell signals after a small movement in the market or change in the asset price is encountered. However, with predictions, decisions can be made before events. Furthermore, stochastic neural networks [33] are better than the deterministic approaches due to slight randomness present in the market that cannot be fully captured using currently available techniques [31]. Therefore, we have incorporated the stochastic neural network that produced predicted prices in the widely used statistical-based trading algorithms.

4.1. Bollinger Bands

Bollinger Bands are a fairly popular technique used in trading [34,35]. They are based on simple moving averages (SMA). A significant amount of action in prices occurs between the bands [36]. The two ends of the bands are based on the difference between the standard deviation (σ) from the mean. The multiplication factor that is popularly used is 2 and the SMA period and the period over which standard deviation is calculated is 20 days. Therefore, the upper Bollinger band is 2 standard deviations more than the SMA, whereas the lower one is 2 standard deviations lower than the SMA, as shown in Equations (4) and (5). This approach also gives insights such as if the bands are close to each other, the prices have less volatility, and there may be a sudden price change in the upcoming future.

$$UBB = SMA_{20} + 2 * \sigma_{20} \quad (4)$$

$$LBB = SMA_{20} - 2 * \sigma_{20} \quad (5)$$

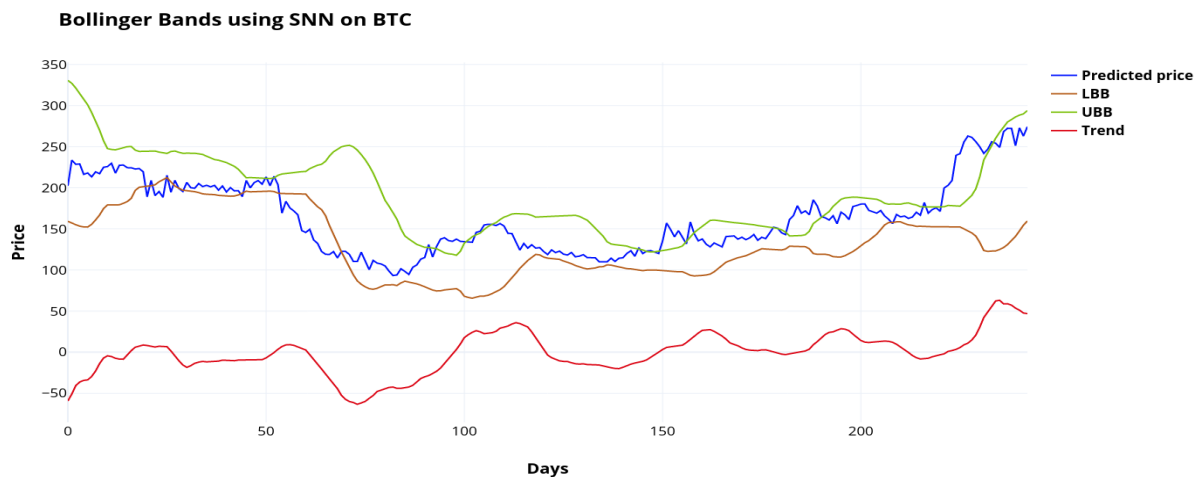
Popularly, the trading decisions are made by checking the trend and the difference in the price with the bands. The trend is generally captured as the difference between the SMA over 5 days and SMA over 30 days. If the trend is positive and the price is greater than the upper Bollinger band, it signifies that the price is increasing, so we should buy. If the trend is negative and the price is lower than the lower Bollinger band, it states that the prices will fall and thus, we should sell. These buy and sell decisions in standard trading algorithms are made using the current day price, but as we are using the predictions from our stochastic neural network, we can invoke buy and sell signals before the actual market reaction takes place; refer to Algorithm 2 and Figure 4.

Algorithm 2 Bollinger Bands using stochastic neural networks.**Input:** P , Predicted price of the next day using stochastic neural networks**Input:** SMA_n , SMA of prices for past n days.**Input:** UBB , Upper Bollinger Band.**Input:** LBB , Lower Bollinger Band.**Output:** Buy and sell signals $trend = SMA_5 - SMA_{30}$ **if** $trend > 0$ **then****if** $P - UBB \geq 0$ **then**

Generate Buy Signal

end if**else****if** $P - LBB \leq 0$ **then**

Generate Sell Signal

end if**end if****Figure 4.** Bollinger Bands strategy using SNN on BTC.

4.2. Pairs Trading

Pairs trading is a strategy that involves observing the market movements of two positively correlated assets [37]. In the event wherein the correlation is temporarily weakened, the trading strategy would be to go short on the outperforming asset and go long on the underperforming asset, in this way speculating that the correlation would gradually increase [21,38].

It is important to note that pairs trading is a market-neutral trading strategy. The long position on the under-performing asset is dollar matched with the short sell of the outperforming asset. In this way, profit is made in short selling and betting on the under-performing asset to re-stabilize and narrow the correlation spread. Standard pairs trading is done by taking into consideration the Z_{score} (Equation (6)) of two positively correlated market assets.

$$Z_{score} = \frac{(\mu_{diff5} - \mu_{diff30})}{\sigma_{30}} \quad (6)$$

where μ_{diff5} and μ_{diff30} is the mean difference between prices of the two currencies over the last 5 days and 30 days, respectively, and σ_{30} is the standard deviation of the difference in prices of the two currencies over 30 days. For the SNN-enhanced pairs trading strategy, we use data from the previous 5 days combined with the predicted price obtained from our stochastic neural to calculate μ_{diff6} and, similarly, μ_{diff31} is calculated using the previous 30 days and the predicted price; μ_{diff6} and μ_{diff31} are used to calculate Z_{score}^{pred} (Equation (7)), which is used for making buy and sell decisions; refer to Algorithm 3 and Figure 5.

$$Z_{score}^{pred} = \frac{(\mu_{diff6} - \mu_{diff31})}{\sigma_{31}} \quad (7)$$

Algorithm 3 Pairs trading using stochastic neural networks.

Input: *date*, date to calculate z score for.

Input: X_{30} , 30 days and X_5 , 5 days prices for X currency.

Input: Y_{30} , 30 days and Y_5 , 5 days prices for Y currency.

Input: Stochastic neural networks trained on currencies X and Y.

Obtain predictions:

$$X_{pred} = model(X, day)$$

$$Y_{pred} = model(Y, day)$$

$$X_{31}^{pred} = concat(X_{30}, X_{pred}), X_6^{pred} = concat(X_5, X_{pred})$$

$$Y_{31}^{pred} = concat(Y_{30}, Y_{pred}), Y_6^{pred} = concat(Y_5, Y_{pred})$$

Calculate metrics:

$$\mu_{diff31} = mean(X_{31}^{pred} - Y_{31}^{pred})$$

$$\mu_{diff6} = mean(X_6^{pred} - Y_6^{pred})$$

$$\sigma_{31} = std(X_{31}^{pred} - Y_{31}^{pred})$$

$$Z_{score}^{pred} = \frac{(\mu_{diff6} - \mu_{diff31})}{\sigma_{31}}$$

Decision making:

if $Z_{score}^{pred} > 1$ **then**

Go short currency on Y and long on X.

end if

if $Z_{score}^{pred} < -1$ **then**

Go short currency on X and long on Y.

end if

Z Score for pairs trading

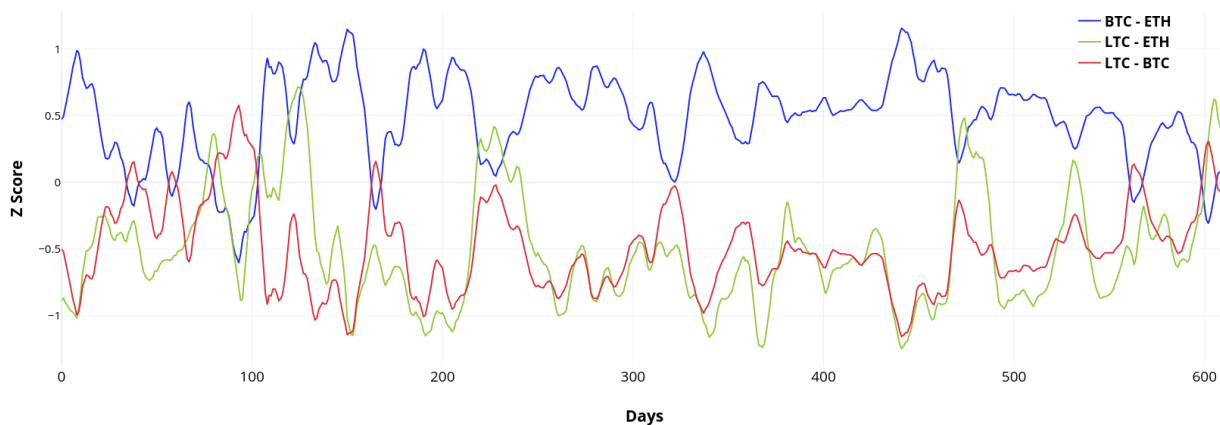


Figure 5. Z-Scores of the combinations of two currencies.

4.3. Awesome Oscillator

Awesome Oscillator (AO) is a market indicator that tries to decipher the direction of market momentum by comparing recent market movements with prior market movements to determine whether the market is bullish or bearish [39,40]. Awesome Oscillator gives entry signals when market momentum is driven by bullish forces and exit signals when market momentum is driven by bearish forces [41]. A standard AO strategy is to take a 5-day simple moving average (SMA) (Equation (8)) and subtract it from the 34-day SMA price to obtain the market trend (Equation (9)). If the AO_{line} is greater than 0, a buy signal is generated. Otherwise, if AO_{line} is less than 0, a sell signal is generated.

$$SMA_n^t = \frac{\sum_{i=t-n}^t price_i}{n} \quad (8)$$

$$AO_{line} = SMA_5 - SMA_{34} \quad (9)$$

In our modified version of AO , we predict AO_{line} by inferring from the previous timestep's AO value and the current timestep's predicted AO value. Here, SMA_5^{pred} is the average of SMA_5 and the predicted price P_{pred} (Equation (10)). Similarly, SMA_{34}^{pred} is the average of SMA_{34} and the predicted price P_{pred} (Equation (11)). The value of AO_{line} is calculated by taking average of current AO_{line} and previous day AO_{line}^{prev} value (Equation (12)); refer to Algorithm 4 and Figure 6.

$$SMA_5^{pred} = \frac{SMA_5 + P_{pred}}{2} \quad (10)$$

$$SMA_{34}^{pred} = \frac{SMA_{34} + P_{pred}}{2} \quad (11)$$

$$AO_{line} = \frac{SMA_5^{pred} - SMA_{34}^{pred} + AO_{line}^{prev}}{2} \quad (12)$$

Algorithm 4 Awesome Oscillator using stochastic neural networks.

Input: P_{pred} , Predicted price of the next day using stochastic neural networks

Input: AO_{line}^{prev} , Previous day AO value

Input: SMA_5 , 5-day simple moving average

Input: SMA_{34} , 34-day simple moving average

Output: Buy and sell signals

$$SMA_5^{pred} = \frac{SMA_5 + P_{pred}}{2}$$

$$SMA_{34}^{pred} = \frac{SMA_{34} + P_{pred}}{2}$$

$$AO_{line} = \frac{SMA_5^{pred} - SMA_{34}^{pred} + AO_{line}^{prev}}{2}$$

$$AO_{line}^{prev} = AO_{line}$$

if $AO_{line} > 0$ **then**

 Generate Buy Signal

end if

if $AO_{line} < 0$ **then**

 Generate Sell Signal

end if

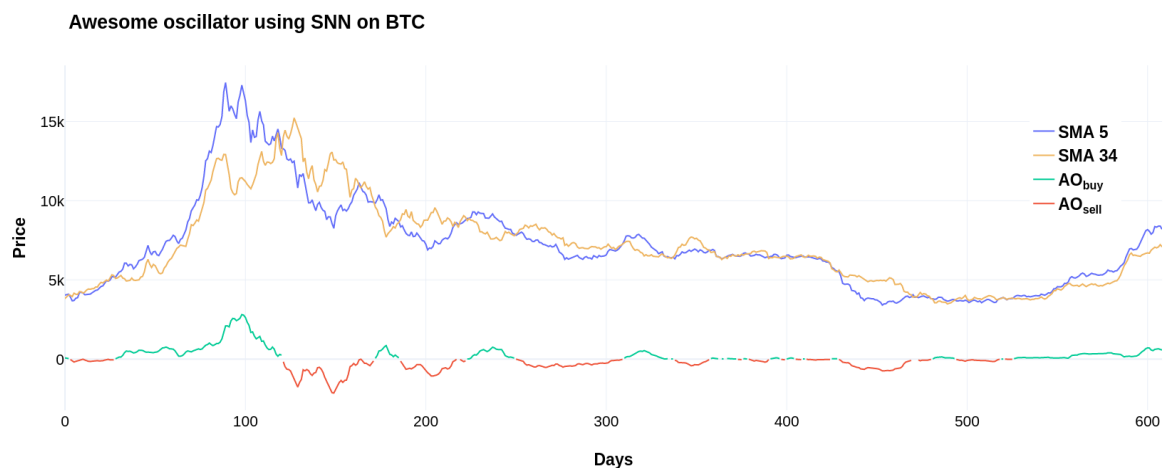


Figure 6. Awesome Oscillator strategy using SNN on BTC.

5. Backtesting Results

To test the trading strategies, we have simulated an investment of \$1000 in each of the three cryptocurrencies: Bitcoin, Ethereum, and Litecoin. The returns on investments are shown in Tables 2 and 3. The money was invested in August 2017 and the return was obtained in December 2019. We have done trading at low frequency (daily) because the market is better understood on medium to long-range intervals than in short intervals.

Table 2. Trading results for the combination of neural network and stand-alone models.

Algorithm	Currency	Net Returns	NP (%)	PP (%)	PF	MDD (%)	Total Trades
Bollinger Band	BTC	6768.55	676.86	80	22.99	18.38	5
	ETH	8861.68	886.17	100	-	20.73	3
	LTC	10,534.62	1053.46	40	9.74	26.27	5
Awesome Oscillator	BTC	3438.42	343.84	68.42	2.33	61.84	19
	ETH	898.14	89.81	55.56	1.80	68.65	18
	LTC	2433.40	243.34	52.94	2.51	69.36	17

Note: All the returns calculated are net returns obtained on an investment of \$1000.

Table 3. Comparison based on returns for pairs trading.

Currency 1	Currency 2	Stand-Alone Model		Combination of Neural Network and Stand-Alone Model	
		Net Returns (\$)	NP (%)	Net Returns (\$)	NP (%)
ETH	BTC	−43.68	−4.37	478.39	47.84
ETH	LTC	765.64	76.56	281.33	28.13
LTC	BTC	3167.49	316.75	851.22	85.12

5.1. Price Prediction Using Stochastic Neural Networks

The first step in our proposed system is to predict the prices of currencies and feed them into algorithmic trading modules. Our stochastic neural network uses the following data features with a time window of 7 days and predicts the expected price on the eighth day. The following features were taken into account while designing the prediction system.

Transactions: This is the number of transactions carried out on a given day. The blockchain network stores information about all transactions carried out and thus, this information is easy to come by. With the number of transactions, we expect the prediction system to infer market volatility along with other features.

Mining Difficulty and Hashrate: Cryptocurrencies are created by utilizing tremendous amounts of energy to mine and maintain transaction records. And it is, therefore, of utmost importance to consider the feasibility of mining concerning the rewards.

Tweets and Google Trends: an essential factor used to determine the price of a cryptocurrency is the social sentiment of the people in the trading ecosystem. Research has shown that tweet volume is correlated with the number of transactions conducted in a given period [42,43]. We use tweet volume and Google search trends as a feature for the price prediction system.

The features used to train the model were ledger size of cryptocurrency, median transaction fee, transaction value transferred, the highest price in the day, the opening price of the day, number of active addresses, market capitalization, transaction fees, volume of cryptocurrency transferred, mining profitability, sent in USD, fees to reward, Google trends daily, the average price in the day, sent by address, average confirmation time in the day, lowest price in the day, hash rate, mining difficulty, volume in USD, the closing price of the day, number of transactions, and tweets.

5.2. Investment Strategy

The Bollinger Band-based algorithms use an investment strategy where all the money is invested in buying the coins if a buy signal is generated. Whenever a sell signal is generated, all the coins available are sold and the money is cashed out. For pairs trading, we used a risk-averse strategy explained as follows. Given the Z_{score} , suppose a signal is generated to go short on currency 1, we check if currency 2 has a positive trend, and

then we invest half of the total cash in currency 2. Otherwise, we convert currency 1 to cash. The same goes for the complimentary scenario. By doing so, we take advantage of a market-neutral strategy by combining it with trend checks.

In this paper, we compare the results obtained by SNN-enhanced models with the results obtained by buy and hold, an RSI based in strategy and standard models. The buy-and-hold strategy is simple. We buy them first and sell them on the last day. The strategy based on the RSI indicator is as follows: after calculating the RSI value for a certain time period, trading is done using buy and sell signals. A buy signal is generated when the currency is oversold (buy threshold). A sell signal is generated when the currency is overbought (sell threshold). The RSI can be calculated using Equations (13)–(16), where SMA is a simple moving average over an n -day period.

If the current closing price is greater than the previous price, i.e., price has increased, then U, D are calculated as

$$U = P_{now} - P_{previous}, D = 0. \quad (13)$$

If current closing price is less than the previous price, i.e., price has decreased, then U, D are calculated as

$$U = 0, D = P_{previous} - P_{now} \quad (14)$$

$$RS = \frac{SMA(U, n)}{SMA(D, n)} \quad (15)$$

$$RSI = 100 - 100 / (1 + RS). \quad (16)$$

The results calculated for stand-alone neural networks (as shown in Table 4) are trend-based, where if the predicted price is greater than the current price and we have not invested, we will buy the cryptocurrencies, and if the predicted price is lower than the current price and we own cryptocurrencies, then we will liquidate all of them and use that cash for future next trade.

Table 4. Returns on stand-alone strategies.

Algorithm	Threshold	Currency	Net Returns (\$)	NP (%)
Buy and Hold		BTC	1245.29	124.53
		ETH	−52.28	−5.23
		LTC	1240.10	124.01
RSI (Time Period 21 days)	Buy 20 Sell 80	BTC	341.60	34.16
		ETH	−193.13	−19.31
		LTC	1340.56	134.06
RSI (Time Period 14 days)	Buy 30 Sell 70	BTC	−278.78	−27.88
		ETH	−624.76	−62.48
		LTC	−133.01	−13.30
Stand-alone Neural Network		BTC	−117.74	−11.77
		ETH	147.91	14.79
		LTC	568.56	56.86
Stand-alone Bollinger Band		BTC	1261.75	126.18
		ETH	2550.59	255.06
		LTC	3836.42	383.64
Stand-alone Awesome Oscillator		BTC	2311.57	231.16
		ETH	805.74	80.57
		LTC	3458.75	345.88

Note: All the returns calculated are net returns obtained on an investment of \$1000.

We observe that the profits obtained by our modified trading algorithms outperformed the traditional buy and hold strategy and a strategy based on the RSI indicator as shown in Tables 2–4. In addition, when compared with the stand-alone models and stand-alone neural network, we infer that the combination of neural network and Bollinger Band has a very low frequency of trades. Still, the model outperforms all other models for all the cryptocurrencies. The combination of neural network and Awesome Oscillator outperformed the stand-alone models in two currencies. Although, in the pairs trading strategy, we see that only one pair outperformed the stand-alone model, the combination of neural network and pairs trading gave consistent results for all three pairs, whereas the stand-alone model holds a loss on one of the pairs. We have also added other trading

results such as net profit percentage (NP), percentage of profitable trades (PP), profit factor (PF), i.e., defined as the ratio of gross profits to gross losses, and maximum drawdown (MDD), i.e., defined as the maximum drop in the value of portfolio before a new peak is achieved [44]; refer to Table 2.

6. Conclusions

Cryptocurrency trading is an arduous task, considering that it has a very volatile market. We try to tackle this problem using stochastic neural networks. Using the predicted prices produced by stochastic neural networks, we try to generate actionable decisions using various market indicators like Bollinger Bands and market-neutral strategies like pairs trading. This paper conducted backtesting for algorithmic trading on three cryptocurrencies: Bitcoin, Ethereum, and Litecoin. We show that our algorithms have significant returns on the invested capital. We believe that there is a huge scope for future improvement and development of algorithms that combine widely used statistical algorithms and prediction-based algorithms to tackle erratic markets. Stochastic networks are not limited to supervised learning algorithms but can be extended to reinforcement learning.

Author Contributions: Conceptualization: A.T., P.P., M.S.R. and S.T.; writing—original draft preparation: F.A. and V.K.; methodology: F.A., B.-C.N. and S.T.; writing—review and editing: P.J. and A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Researchers Supporting Project No. (RSP2022R509) King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No data is associated with this research work.

Acknowledgments: This work was funded by the Researchers Supporting Project No. (RSP2022R509) King Saud University, Riyadh, Saudi Arabia and by the Ministry of Research, Innovation and Digitization through Program 1-Development of the national research and development system, Subprogram 1.1. Institutional performance-Projects to finance excellence in RDI, Contract No. 19PFE/30.12.2021 and a grant of the National Center for Hydrogen and Fuel Cells (CNHPC)—Installations and Special Objectives of National Interest (IOSIN).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Messina, I. The Effects of Chavez's Populism on Today's Venezuela. *Perceptions* **2019**, *5*, 1–5. [CrossRef]
2. Vora, J.; Nayyar, A.; Tanwar, S.; Tyagi, S.; Kumar, N.; Obaidat, M.S.; Rodrigues, J.J.P.C. BHEEM: A Blockchain-Based Framework for Securing Electronic Health Records. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
3. Mehta, P.; Gupta, R.; Tanwar, S. Blockchain envisioned UAV networks: Challenges, solutions, and comparisons. *Comput. Commun.* **2020**, *151*, 518–538. [CrossRef]
4. Tanwar, S.; Bhatia, Q.; Patel, P.; Kumari, A.; Singh, P.K.; Hong, W. Machine Learning Adoption in Blockchain-Based Smart Applications: The Challenges, and a Way Forward. *IEEE Access* **2020**, *8*, 474–488. [CrossRef]
5. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: <http://bitcoin.org/bitcoin.pdf> (accessed on 26 March 2022).
6. Ali, R.; Barrdear, J.; Clews, R.; Southgate, J. The economics of digital currencies. *Bank Engl. Q. Bull.* **2014**, *54*, 276–286.
7. Thakkar, P.; Varma, K.; Ukani, V.; Mankad, S.; Tanwar, S. Combining User-Based and Item-Based Collaborative Filtering Using Machine Learning. In *Information and Communication Technology for Intelligent Systems*; Satapathy, S.C., Joshi, A., Eds.; Springer: Singapore, 2019; pp. 173–180.
8. Bhattacharya, P.; Tanwar, S.; Bodke, U.; Tyagi, S.; Kumar, N. BinDaaS: Blockchain-Based Deep-Learning as-a-Service in Healthcare 4.0 Applications. *IEEE Trans. Netw. Sci. Eng.* **2019**, *8*, 1242–1255. [CrossRef]
9. Vachhani, H.; Obaidat, M.S.; Thakkar, A.; Shah, V.; Sojitra, R.; Bhatia, J.; Tanwar, S. Machine Learning Based Stock Market Analysis: A Short Survey. In *Innovative Data Communication Technologies and Application*; Raj, J.S., Bashar, A., Ramson, S.R.J., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 12–26.

10. Gupta, R.; Tanwar, S.; Tyagi, S.; Kumar, N. Machine Learning Models for Secure Data Analytics: A taxonomy and threat model. *Comput. Commun.* **2020**, *153*, 406–440. [\[CrossRef\]](#)
11. McCoy, M.; Rahimi, S. Prediction of highly volatile cryptocurrency prices using social media. *Int. J. Comput. Intell. Appl.* **2020**, *19*, 2050025. [\[CrossRef\]](#)
12. Liew, J.; Li, R.; Budavári, T.; Sharma, A. Cryptocurrency Investing Examined. *J. Br. Blockchain Assoc.* **2019**, *2*, 1–12. [\[CrossRef\]](#)
13. Mokni, K. When, where, and how economic policy uncertainty predicts Bitcoin returns and volatility? A quantiles-based analysis. *Q. Rev. Econ. Financ.* **2021**, *80*, 65–73. [\[CrossRef\]](#)
14. Liu, J.; Serletis, A. Volatility in the cryptocurrency market. *Open Econ. Rev.* **2019**, *30*, 779–811. [\[CrossRef\]](#)
15. CoinMarketCap. Available online: <https://coinmarketcap.com/charts/> (accessed on 26 March 2022).
16. Begušić, S.; Kostanjčar, Z.; Stanley, H.E.; Podobnik, B. Scaling properties of extreme price fluctuations in Bitcoin markets. *Phys. A Stat. Mech. Its Appl.* **2018**, *510*, 400–406. [\[CrossRef\]](#)
17. Bouri, E.; Gkillas, K.; Gupta, R.; Pierdzioch, C. Forecasting realized volatility of bitcoin: The role of the trade war. *Comput. Econ.* **2021**, *57*, 29–53. [\[CrossRef\]](#)
18. Garcia, D.; Schweitzer, F. Social signals and algorithmic trading of Bitcoin. *R. Soc. Open Sci.* **2015**, *2*, 150288. [\[CrossRef\]](#)
19. Deng, Y.; Bao, F.; Kong, Y.; Ren, Z.; Dai, Q. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 653–664. [\[CrossRef\]](#)
20. Nakano, M.; Takahashi, A.; Takahashi, S. Bitcoin technical trading with artificial neural network. *Phys. A Stat. Mech. Its Appl.* **2018**, *510*, 587–609. [\[CrossRef\]](#)
21. Fil, M.; Kristoufek, L. Pairs trading in cryptocurrency markets. *IEEE Access* **2020**, *8*, 172644–172651. [\[CrossRef\]](#)
22. Colianni, S.; Rosales, S.; Signorotti, M. Algorithmic trading of cryptocurrency based on Twitter sentiment analysis. *CS229 Proj.* **2015**, *1*, 1–5.
23. Fister, D.; Mun, J.; Jagrič, V.; Jagrič, T. Deep Learning for Stock Market Trading: A Superior Trading Strategy? *Neural Netw. World* **2019**, *29*, 151–171. [\[CrossRef\]](#)
24. Takeuchi, L.; Lee, Y.Y.A. *Applying Deep Learning to Enhance Momentum Trading Strategies in Stocks*; Technical Report; Stanford University: Stanford, CA, USA, 2013.
25. Yong, B.X.; Rahim, M.R.A.; Abdullah, A.S. A stock market trading system using deep neural network. In Proceedings of the Asian Simulation Conference, Melaka, Malaysia, 27–29 August 2017; Springer: Singapore, 2017; pp. 356–364.
26. Zanc, R.; Cioara, T.; Anghel, I. Forecasting Financial Markets using Deep Learning. In Proceedings of the 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 5–7 September 2019; pp. 459–466.
27. Dipple, S.; Choudhary, A.; Flamino, J.; Szymanski, B.K.; Korniss, G. Using correlated stochastic differential equations to forecast cryptocurrency rates and social media activities. *Appl. Netw. Sci.* **2020**, *5*, 17. [\[CrossRef\]](#)
28. Wong, E. Stochastic neural networks. *Algorithmica* **1991**, *6*, 466–478. [\[CrossRef\]](#)
29. Florensa, C.; Duan, Y.; Abbeel, P. Stochastic neural networks for hierarchical reinforcement learning. *arXiv* **2017**, arXiv:1704.03012.
30. Fama, E.F. Random Walks in Stock Market Prices. *Financ. Anal. J.* **1965**, *21*, 55–59. [\[CrossRef\]](#)
31. Jay, P.; Kalariya, V.; Parmar, P.; Tanwar, S.; Kumar, N.; Alazab, M. Stochastic neural networks for cryptocurrency price prediction. *IEEE Access* **2020**, *8*, 82804–82818. [\[CrossRef\]](#)
32. Koki, C.; Leonardos, S.; Piliouras, G. Do Cryptocurrency Prices Camouflage Latent Economic Effects? A Bayesian Hidden Markov Approach. *Proceedings* **2019**, *28*, 5.
33. Ginzburg, I.; Sompolinsky, H. Theory of correlations in stochastic neural networks. *Phys. Rev. E* **1994**, *50*, 3171. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Bollinger, J. *Bollinger on Bollinger Bands*; McGraw-Hill: New York, NY, USA, 2002.
35. Bollinger, J. Using bollinger bands. *Stock. Commod.* **1992**, *10*, 47–51.
36. Lento, C.; Gradojevic, N.; Wright, C. Investment information content in Bollinger Bands? *Appl. Financ. Econ. Lett.* **2007**, *3*, 263–267. [\[CrossRef\]](#)
37. Elliott, R.J.; Van Der Hoek, J.; Malcolm, W.P. Pairs trading. *Quant. Financ.* **2005**, *5*, 271–276. [\[CrossRef\]](#)
38. Vidyamurthy, G. *Pairs Trading: Quantitative Methods and Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2004; Volume 217.
39. Trading Strategy Guides. Bill Williams Awesome Oscillator Strategy—Big Profits, Small Losses. 2017. Available online: <https://tradingstrategyguides.com/indicator-combinations-in-forex-trading/> (accessed on 26 March 2022).
40. Trading Strategy Guides. The 10 Best Indicator Combinations in Forex Trading. 2017. Available online: <https://tradingstrategyguides.com/bill-williams-awesome-oscillator-strategy/> (accessed on 26 March 2022).
41. Trading Strategy Guides. NEO Cryptocurrency Strategy—Scalping with BB and AO. 2018. Available online: <https://tradingstrategyguides.com/neo-cryptocurrency-strategy/> (accessed on 26 March 2022).
42. Mittal, A.; Dhiman, V.; Singh, A.; Prakash, C. Short-Term Bitcoin Price Fluctuation Prediction Using Social Media and Web Search Data. In Proceedings of the 2019 Twelfth International Conference on Contemporary Computing (IC3), Noida, India, 8–10 August 2019.
43. Smuts, N. What Drives Cryptocurrency Prices? *ACM Sigmetrics Perform. Eval. Rev.* **2019**, *46*, 131–134. [\[CrossRef\]](#)
44. Cohen, G. Forecasting bitcoin trends using algorithmic learning systems. *Entropy* **2020**, *22*, 838. [\[CrossRef\]](#)