

Received April 15, 2020, accepted April 23, 2020, date of publication April 27, 2020, date of current version May 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990659

Stochastic Neural Networks for Cryptocurrency Price Prediction

PATEL JAY¹, VASU KALARIYA¹, PUSHPENDRA PARMAR¹, SUDEEP TANWAR¹, (Member, IEEE),
NEERAJ KUMAR^{1,2,3,4}, (Senior Member, IEEE),
AND MAMOUN ALAZAB⁵, (Senior Member, IEEE)

¹Department of Computer Science and Engineering, Institute of Technology, Nirma University, Ahmedabad 382481, India

²Department of Computer Science Engineering, Thapar Institute of Engineering and Technology, Deemed to be University, Patiala 147004, India

³Department of Computer Science and Information Engineering, Asia University, Taichung 41354, Taiwan

⁴Department of Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

⁵College of Engineering, IT & Environment, Charles Darwin University, Casuarina, NT 0810, Australia

Corresponding authors: Neeraj Kumar (neeraj.kumar@thapar.edu) and Mamoun Alazab (mamoun.alazab@cdu.edu.au)

This work was supported by the Department of Corporate and Information Services, Northern Territory Government (NTG), Australia.

ABSTRACT Over the past few years, with the advent of blockchain technology, there has been a massive increase in the usage of Cryptocurrencies. However, Cryptocurrencies are not seen as an investment opportunity due to the market's erratic behavior and high price volatility. Most of the solutions reported in the literature for price forecasting of Cryptocurrencies may not be applicable for real-time price prediction due to their deterministic nature. Motivated by the aforementioned issues, we propose a stochastic neural network model for Cryptocurrency price prediction. The proposed approach is based on the random walk theory, which is widely used in financial markets for modeling stock prices. The proposed model induces layer-wise randomness into the observed feature activations of neural networks to simulate market volatility. Moreover, a technique to learn the pattern of the reaction of the market is also included in the prediction model. We trained the Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) models for Bitcoin, Ethereum, and Litecoin. The results show that the proposed model is superior in comparison to the deterministic models.

INDEX TERMS Cryptocurrency, multilayer perceptron, long short-term memory, random walk, stochasticity.

I. INTRODUCTION

In a continuously evolving technological landscape, there has been a paradigm shift in the mode of transactions from physical payments like cash and cheques to digital transactions. One important aspect of using currency, either as a medium of transaction or as an asset, is to predict its expected value. To a great extent, the value and stability of any currency depends on the controlling authority, which in the case of fiat currencies is the Government of the country. Detrimental Governmental interference in the financial system can lead to unforeseen consequences of devastating scales, as seen in Venezuela [1]. But, in the case of digital currencies, the value is determined by the consistency and security of the platform that the currency is deployed on. Conventional digital cash is prone to the flaw of double-spending. Digital currencies in cyberspace are exposed to security attacks which may lead to

transaction data manipulation. With an increasing number of such flaws, traditional currencies fall prey to instability and devaluation [2].

A plausible solution to the aforementioned issues is the usage of blockchain-based Cryptocurrencies. Blockchain is an emerging technology, which stores the information in an immutable way across a network to provide security, decentralization, and transparency, which is precisely what is needed for an effective currency [3]–[5]. Cryptocurrencies, unlike conventional money, use cryptographic ciphers to conduct financial transactions. Over the past decade, digital finance has grown exponentially, with Cryptocurrencies at the helm of this innovative stride forward [6]. The market capitalization of Cryptocurrencies is calculated to \$266 billion and is projected to have a growth of 11.9% by 2024 as per the CAGR reports [7]. The essential feature of a Cryptocurrency is that it cannot be controlled by a central authority due to its decentralized nature inherited from blockchain and thus it restricts corruption. Due to this, Cryptocurrencies are naturally robust

The associate editor coordinating the review of this manuscript and approving it for publication was Hong-Ning Dai.

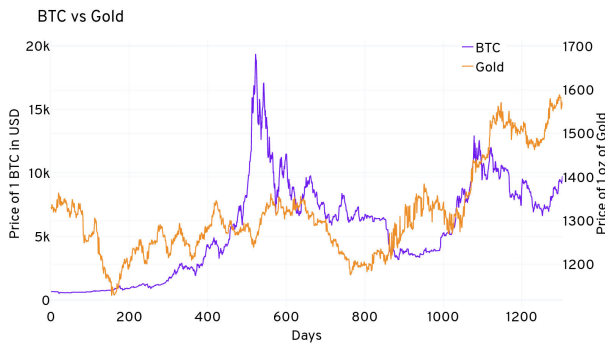


FIGURE 1. Bitcoin vs Gold prices over a span of 1300 days.

TABLE 1. Correlation of BTC with major financial assets [10].

Asset	Correlation with Bitcoin
Gold	0.0459
Silver	0.0071
WTI (Crude Oil)	0.0158
S&P 500	0.0491
MSCI World	0.0457
MSCI EM50	0.0042

towards corruption induced devaluations, which may occur in fiat currencies. Cryptocurrencies avert the problem of double-spending through multiple verifications from the neighboring nodes in the blockchain network. As the number of confirmation increases, the transaction becomes more and more reliable and irreversible. Transactional records in the ledger of blockchain are immutable as a record is virtually impossible to alter in all network nodes. Thus, after a successful transaction, the record can not be tampered.

As a consequence of the aforementioned advantageous characteristics and global access to Cryptocurrencies, they can be used as a medium of transaction, as well as a store of wealth [8], [9]. However, the value of Cryptocurrencies still heavily relies on erratic market trends and social sentiments. Also, Cryptocurrencies have a low correlation with major financial assets [10], [11], thus traditional methods, that have been used in finance are rendered ineffective. Table 1 shows the Pearson correlation of Bitcoin with other major financial assets. It is evident from Fig. 1 that the price Bitcoin has no correlation with the price of Gold.

Having vast amounts of openly available data on the Cryptocurrencies market and social trends information, machine learning algorithms can be used to forecast the prices with Cryptocurrencies [12]. These algorithms are a set of methods for learning mathematical models from data without explicitly programming the computer to do a specific task. But, with an increase in the complexity of the data for the Cryptocurrency market, there is a need of different models, which can capture more complex representations of data. Deep learning models [13] specifically recurrent neural networks can be used to solve the time-series problem of predicting the prices of Cryptocurrencies. Numerous research has been

explored by various authors in the last to predict the value of equity and securities using machine learning and deep learning algorithms [14], [15]. However, comparatively fewer research work has been carried out on forecasting the price of Cryptocurrencies.

A. MOTIVATION

Cryptocurrencies are primarily used as a means of money exchange. But, in the past few years, we have seen that trading with Cryptocurrencies has been an attractive investment opportunity [16]. It is the prevalent opinion of stock market professionals and other investors that the Cryptocurrency market is the most uncertain place for investment due to its volatility and heavy reliance on social sentiments. Along with this, effectively predicting the price of various cryptocurrencies will allow us to predict the total compute power of blockchain [17]. The value of Cryptocurrencies is primarily affected by a large number of factors like social sentiment, legislature, past price trends, and trade volumes. A significant amount of research work to anticipate Cryptocurrency prices using machine learning techniques has already been explored in the last few years [18], [19]. Motivated from the aforementioned discussion, in this paper, we provide a prediction model, which is used to predict the price of different cryptocurrencies using deep learning models. We address the problem of erratic fluctuations in the prices of Cryptocurrencies by inducing stochastic behaviour in deep neural networks to simulate market volatility.

B. CONTRIBUTION

Following are the research contributions of this paper.

- A technique to predict the prices of prevalent Cryptocurrencies, i.e., Bitcoin, Ethereum, and Litecoin is designed using a stochastic neural network process.
- A mathematical formulation of stochastic layers in deep neural networks is done that characterizes the erratic behavior of a financial system.
- We determine the pattern of the market's reaction to any updated information regarding the Cryptocurrency and exhibits improvement over existing models in predicting the prices of Cryptocurrencies.

C. ORGANIZATION

Rest of the paper is organized as follows. Section II discusses the previous work that has been done to forecast Cryptocurrency prices. Section III explains the concept of stochasticity and its application in the financial market. In Section IV, we present a mathematical formulation of stochasticity in the context of neural networks. Section V discusses the experimental details of models for predicting Cryptocurrency prices and finally, Section VI concludes the paper.

II. RELATED WORK

A substantial amount of research has been carried in the prediction of stability and prices of equity and other market

assets over the past decades. However, due to its newfangledness, there is unsubstantial research done on the value prediction of Cryptocurrencies. Nonetheless, there is an increasing trend in the research effort done to anticipate the prices of Cryptocurrencies. In this section, we present pivotal milestones in the field. Here, we acquaint the reader with a diversified set of machine learning approaches [20] that have been used to predict price trends of various currencies.

A. REGRESSION

The fundamental task in modeling a Cryptocurrency is to predict the price given to the priors. A simplistic approach to forecast the price over a continuous space is regression. Regression is a type of statistical method to determine the relationship between a dependant variable and one or more independent variables. This relationship is represented as a sum of products of independent variables with some relational constant weight. In the context of price prediction, market indicators and social sentiments can be used as the independent variables and price as the dependent variable.

$$y = \theta_0 + \theta_1x_1 + \theta_2x_2 + \dots + \theta_nx_n \quad (1)$$

Saad *et al.* [21], [22] used a multivariate regression model trained using gradient descent over mean square error. Features such as price, mining difficulty, hash rate, user count, etc were used regress and obtain the predicted price. They achieved a mean absolute error of 0.0162 and 0.0563 over bitcoin and Ethereum, respectively when testing over half of the dataset. Mittal *et al.* [23] extended usage of regression to social sentiments. They exhibited a positive correlation between the price fluctuations of Bitcoin and social sentiment. They showed that there is a significant correlation between Google trends, tweet sentiment, and tweet volume. Linear regression and polynomial regression were used to predict the prices. They evaluated the models by calculating the frequency of correct predictions within the bounds of margin accuracy. An accuracy of 77.01% and 66.66% was observed incorrectly predicting the trend of the price using tweet volume and google trends respectively. An obvious issue with regression models is that they are unable to learn non-linear and multi-leveled dependencies among the features.

B. MULTILAYER PERCEPTRON

Following the Moore's Law, faster and more efficient computing power has been harnessed to train Artificial Neural Networks (ANN). Neural nets can effectively learn and represent linear and non-linear dependencies between key variables and the output variable. They consist of input layers that are further connected to a hierarchy of hidden layers, which in turn pass the learned information to the output layer. Each edge connecting the neurons in different layers comprise of weights and bias representing the relation between the connected neurons. Activation functions are applied after linear matrix computation is carried out. These functions

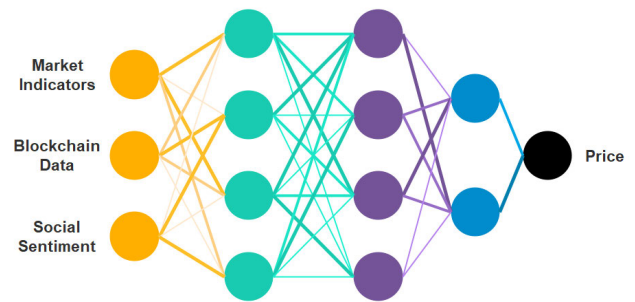


FIGURE 2. Cryptocurrency price prediction using MLP.

are responsible for introducing non-linearities into the network, which can essentially help in price forecasting [24] as required in our case. Commonly used non-linear activation functions are *sigmoid* and *tanh* functions. The architecture of the network determines the kind of dependencies the neural network learns. Neural nets are trained using a learning algorithm known as backpropagation [25]. The training process begins by initiating network weights and biases randomly, and then iteratively calculating error and propagating error signal as gradients of weights. Weights are updated iteratively till a feasible set of weight values is obtained. Fig. 2 illustrates an exemplification of MLP in case of Cryptocurrency price prediction.

The approach of an ensemble of neural networks was incorporated in the model proposed by Sin *et al.* [26], to predict the upward or downward trend using bitcoin market data of the 50 consecutive days. Each network module in the ensemble was a multi-layered perceptron three layers deep, taking a total of 190 features. They used the Lavenberg-Marquardt algorithm for training the MLPs. Genetic Algorithm based Selective Ensemble (GASEN) was employed to select the five best-performing perceptrons. They achieved an accuracy of 64% in classifying whether an upward trend or downward trend is to be expected. Their model did not predict the price, rather just a green or red signal. Their work was limited to only one Cryptocurrency, Bitcoin.

Using Bayesian theory, Jang *et al.* [27] explained the Bitcoin's high price volatility. They proposed a multi-layer perceptron that maximizes the value of posterior, instead of maximizing likelihood like traditional neural architectures. Their model was trained using the rollover framework, wherein an old price time-step value is discarded for every new price. In this way, they dispose of long term dependencies that may be irrelevant for anticipating newer prices. Using the rollover strategy means less computational cost while training as opposed to sequential recurrent architectures. They achieved a test Mean Absolute Percentage Error of 1% in predicting the log price of Bitcoin in the surge.

C. RECURRENT NEURAL NETWORKS

Forecasting prices of currencies is an inherently sequential task. To deal with time-dependent data, a new class of neural

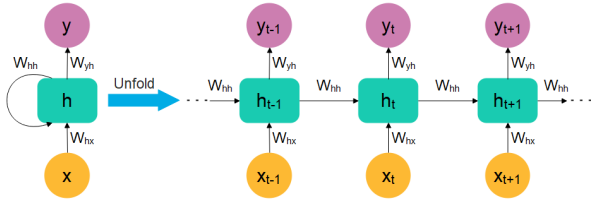


FIGURE 3. Unfolding of recurrent neural networks.

networks was introduced termed as Recurrent Neural Networks [28]. In this architecture, a directed graph along the sequence is formed to manage sequential data. Here, the data at the previous time step is used to feed as inputs to predict the values at the next time step. Typical RNNs use previous values as well as some inputs at that time step to predict the values at each time step.

$$h_t = f(W_{hh}h_{t-1} + W_{hx}x_t + b_h) \quad (2)$$

$$y_t = f(W_{yh}h_t + b_y) \quad (3)$$

Associating this with price prediction in Fig. 3, new market information, x is fed into the model at each timestep. The model picks up the necessary temporal dependencies, h utilizes them to predict the price, y_t of the Cryptocurrency.

D. LONG SHORT-TERM MEMORY

Recurrent Neural Networks are unable to capture long term dependencies after some amount of sequence length and thus again a new powerful architecture was proposed by Hochreiter *et al.* [35] known as LSTM. In this network, a new memory cell state is added along with gating functionality that controls what information is to be discarded and what new information is to be added to provide long term dependencies. Here, the cell states are passed across the network and accordingly they are updated and modified as per the importance of the previous cell state data which is being carried since it became important for the future. Gates are used to modifying the cell state as well as process inputs and produce output.

$$f_t = \text{sigmoid}(W_f.[h_{t-1}, x_t] + b_f) \quad (4)$$

$$i_t = \text{sigmoid}(W_i.[h_{t-1}, x_t] + b_i) \quad (5)$$

In Eq. (4) and (5) the f gate, by observing the previous activations, h and the current market and social indicators, x outputs a value between 0 and 1 for each cell states, where 1 represents to keep this information for future purposes while the 0 discards the information [36]. The i gate conveys the amount of current information that will be relevant in the future.

$$C_t^* = \text{tanh}(W_c.[h_{t-1}, x_t] + b_c) \quad (6)$$

$$C_t = f_t * C_{t-1} + i_t * C_t^* \quad (7)$$

The candidate for the cell update is the C^* . Using f , i and C^* , a suitable update is performed to the cell state that is passed

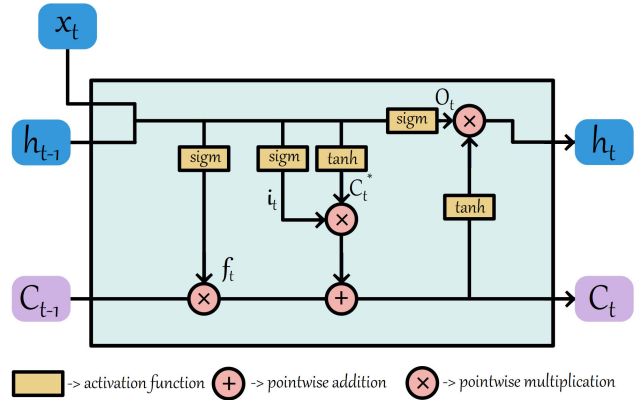


FIGURE 4. A long short-term memory cell.

throughout the network to capture long term dependencies.

$$o_t = \text{sigmoid}(W_o.[h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = o_t * \text{tanh}(C_t) \quad (9)$$

Based on the cell state, o gate determines what part of the cell is going to be the output. This output is combined with the current cell state activated by the tanh layer to give the current activation h .

Mittal *et al.* [23] proposed an RNN and LSTM model that utilizes google trends and tweet volume along with market factors to anticipate the price of bitcoin. They showed that sequential models outperformed ARIMA (Autoregressive integrated moving average) [37], the standard model to analyze time-series data in traditional statistics and econometrics. The RNN model achieves an accuracy of 62.45% and 53.46% on trends and volume respectively. On the other hand, their LSTM model achieves an accuracy of 50% and 49.89% on trends and volume respectively. Smuts [29] used VADER [38], a sentiment analysis python library for social media text, to correlate telegram sentiment and the price of Bitcoin and Ethereum and predict the price of the currencies using an LSTM. They achieved an accuracy of 63% on Bitcoin data and 56% on Ethereum data.

E. OTHER METHODS

Apart from the previously introduced methods, regression trees, decision trees, support vector regression [39], and other algorithms have been used. For example, Laura *et al.* [30] designed two models, an ensemble of regression trees and a recurrent neural net. Two versions of the ensemble of regression trees were considered, the first was a single model to describe the price change for all currencies combined. The second method was to construct individual models for each currency. The regression tree models were built using the XGBoost algorithm. To exploit temporal dependencies, a LSTM was selected as their second model. They evaluate the models by calculating the return on investment (ROI) [40] and comparing the performance against a simple moving average.

TABLE 2. A relative comparison of recent works in the field of Cryptocurrency price prediction.

Author	Year	Brief Summary	Methods	Results	Cryptocurrencies	Dataset
Saad et al. [21]	2018	The market of Cryptocurrency was analyzed with the help of correlation analysis of various attributes and finally a machine learning model was built.	Multivariate Regression	The model achieved a MAE of 0.0162 on BTC and 0.0563 on ETH	Bitcoin, Ethereum	blockchain.info, etherscan.io
Mittal et al. [23]	2019	Google Trends, Tweet Sentiment and Volume were used to predict the fluctuations in the Bitcoin prices and proposed that Wikipedia and Facebook Posts can also be included to improve the predictions. Tweet Sentiment showed the worst performance.	Regression techniques, RNN, LSTM	Polynomial Regression obtained an accuracy of 77.01% using tweet Volume and 66.66% using google trends.	Bitcoin.	Coindesk, Twitter API, Google Trends
Sin et al. [26]	2017	It explained the dependencies of next day price on features of Bitcoin using Genetic Algorithm based Selective Neural Network Ensemble (GASEN) based on ANN.	Ensemble of Neural Networks, MLP	The neural network ensemble model obtained an accuracy of 64% in predicting the trend of price movement.	Bitcoin.	blockchain.info, bitcoinity.org
Jang et al. [27]	2018	Deployed a Bayesian Neural Network to capture non-linear influences of blockchain information and other macro-economic factors on price formation. Explained the price volatility in BTC and predicted the log price of BTC.	MLP	The bayesian neural net obtained a mean absolute percentage error of 1% in the task of regressing price of BTC.	Bitcoin.	bitcoincharts.com, blockchain.info (empirical analysis)
Smuts et al. [29]	2019	This paper tested the capability of trends and telegram data to predict the short-term trends in Cryptocurrency price movements.	LSTM	The model obtained an accuracy of 63% for BTC and 56% for ETH	Bitcoin, Ethereum.	Google Trends, Telegram, Market data aggregator
Laura et al. [30]	2018	Two ensemble models of regression trees created using XGBoost algorithm and RNNs. For temporal dependencies LSTM was also used and ROI was calculated.	Ensemble of regression trees(using XGBoost), RNN, LSTM	They device a trading strategy that uses an LSTM trained using Sharpe ratio and geometric mean optimization.	A total of 1681 Cryptocurrencies used.	coinmarketcap.com
Chen et al. [31]	2020	Features engineering was done to obtain 4 significant classes of data as follows, 1) Bitcoin property and network 2) Marketing and trading data 3) Investors and media attention and 4) XAU gold spot price. They task this data to trend prediction.	Logitic regression, LDA, random forest, SVM, LSTM	LSTM achieved best accuracy of 67.2%. Average accuracy of statistical models was 53%, where as machine learning models had 62.2%.	Bitcoin	coinmarketcap.com
Lahmiri et al. [32]	2018	They use Largest Lyapunov Exponent (LLA) and Detrended Fluctuation Analysis (DFA) to detect chaos and fractal characteristics along with LSTM to predict prices of various Cryptocurrencies.	LSTM	They achieve RMSE of 2.75×10^3 on Bitcoin and 4.99×10^{-2} on Ripple.	Bitcoin, Ripple	Unknown data source
Marcell et al. [33]	2019	The Bitcoin transaction network's most frequent edges are used to predict the price movement. This is based on the work by Kondor et al. [34] that states that structural changes in the Bitcoin network affects the price of Bitcoin.	Single layer neural network	They obtain an accuracy of 60.05% in predicting the trend of price movement.	Bitcoin	coindesk.com
The Proposed Approach	2020	Used a stochastic module that captures the markets' reaction to new knowledge. Moreover, by utilizing the stochasticity factor in network models improves the results on all metrics used.	MLP, LSTM, Stochastic Neural Networks	There is a average relative improvement of 4.6% on MAPE using stochasticity.	Bitcoin, Ethereum, Litecoin	coinmarketcap.com

III. PROBLEM FORMULATION AND STOCHASTIC PROCESSES

In this section, we describe the key obstacle restricting the effective price prediction, which is erratic fluctu-

ations in Cryptocurrency prices. As randomness is at the core of the problem, we introduce stochastic processes and show how they are related to the Cryptocurrency market. At the end of the section,

we show how the market can be modeled as a random walk.

A. THE PROBLEM OF ERRATIC FLUCTUATIONS

The value of market assets is determined by various factors that include supply and demand, the performance of the economy, growth rate, inflation, political factors, and human psychology. With all of these factors continuously changing, the aggregation of these factors generates erratic and irregular fluctuations in the prices of market assets. Cryptocurrencies, like other market assets, are prone to the problem of random-like fluctuations in their prices. The financial market is not inherently stochastic, however, it is sufficiently complex to be incomprehensible to us and our systems. And thus, designing a deterministic model that takes into consideration, all the socio-economic factors is out of option. In this section, we will describe how to model market assets non-deterministically.

B. STOCHASTIC PROCESSES

All processes in nature can be classified as deterministic given all the information pertaining to them. However, most natural systems are too complicated to be modeled given limited information about them. Thus, stochastic processes come into the picture, wherein partial information of the system can be used to determine a possible outcome over the set of all possibilities in the probability space. Stochastic processes are sets of random variables that evolve over time in an arbitrary manner. Most natural systems can be considered as stochastic processes from our perspective. Examples of stochastic processes include the weather system, audio-video signals, and the financial market.

The value of a market asset, like a Cryptocurrency, is determined by intricate factors that are continuously evolving in a near-random direction at a seemingly random pace [41]. At the core, collective human behavior determines the supply and demand of an asset which in turn determines the current value of the said asset. Modeling, human behavior is an impossible task, and thus we consider the value of an asset to stochastically determined. Defining the behavior of the market as an erratic process gives us the advantage of using the well-defined mathematical field of stochasticity.

Before we introduce stochasticity into the market scene, we present the types of stochastic processes and how we can relate them to the Cryptocurrency market. The most rudimentary stochastic process is the Bernoulli process, where random variables hold binary value and the sequence produced by it is Identically and Independently Distributed (IID). The IID property, which states that the system of random variables is drawn from the same probability distribution and all the random variables are mutually independent, is at the essence of the market's stochasticity. Building on that, random walk is a stochastic process that sums up IID random variables and thus has the property of evolution in time. Burton Malkiel [42] proposed that market assets follow a random walk.

A random walk is defined as the path formed by a sequence of random IID steps, given the starting point.

$$y_t = y_0 + \sum_{i=1}^t \xi_i \quad (10)$$

where ξ_i is an IID at time step i and y_0 is the starting point of the process.

Alternatively, we can obtain a path uncovered by a random walk as a bootstrapping process, taking into account the most recent state of the system. Consider the second last state of a system y_{t-1} ,

$$y_{t-1} = y_0 + \sum_{i=1}^{t-1} \xi_i \quad (11)$$

Moving one step forward as defined by the process, we obtain the next state of system as shown below.

$$y_{t-1} + \xi_t = y_0 + \sum_{i=1}^{t-1} \xi_i + \xi_t \quad (12)$$

$$y_{t-1} + \xi_t = y_t \quad (13)$$

Eq. (13) is more convenient to use as it is more computationally efficient when t is extremely large. Hence, this notion is used in the implementation of the proposed approach.

Bringing this into context with the value of a market asset, like a Cryptocurrency, the value can be thought of as being produced by a random walk, where ξ_t can be considered as an aggregation of all market factors that may have possibly affected the value of the asset [32]. However, this approach does not take into consideration information regarding essential market factors. We address this issue by introducing neural networks to take into account important market statistics and social sentiment [43], [44].

IV. PROPOSED APPROACH: STOCHASTIC NEURAL NETWORKS

In this section, we incorporate stochasticity into neural networks and formulate the mathematics for a layer-wise stochastic walk. Moreover, we introduce the algorithm for stochastic forward propagation in neural networks. We propose stochastic MLP and LSTM models to predict the prices of Cryptocurrencies.

A. STOCHASTICITY IN NEURAL NETWORKS

According to the efficient market hypothesis by Malkiel [45], all the past information regarding the market asset is reflected in the current value of the asset and the market will instantly acknowledge new information and react to it accordingly. Therefore, all the effort of predicting prices by analyzing information is futile. However, we can observe how the market reacts to information and develop a pattern that exhibits the behavior of the market when new information is widely available. This pattern has to be stochastic [46] so as to accommodate the multiplicity of all possible outcomes to the arrival of new knowledge.

Before we introduce stochasticity into the picture, we need a way to distill market features and describe the interdependencies between market statistics and social sentiment. To do this, we use a neural network because a neural network is a universal function approximator that tries to map dependencies between variables. The final value of a market asset is determined by a hierarchy of features that roots from factors like supply and demand, economy and human behavior and a neural network is an excellent candidate to do just this [47].

There are two ways to inculcate randomness in a neural network, the first is to randomly change the weights by a small degree and the second way is by adding randomness to the activations at runtime. The first approach is not ideal because it would mean that feature detection will get noisy as the network evolves and may eventually forget dependencies. Intuitively, the second approach seems fitting because the randomness in activations can be interpreted as random changes in features, which in turn can be thought of as replicating the erratic behaviors of the market.

We propose a generalized formulation of the stochastic behavior of a layer in a deep neural network as follows,

$$s_t = h_t + \gamma \xi_t \times reaction(h_t, s_{t-1}), \quad 0 < \gamma < 1 \quad (14)$$

where h_i is the activation values of the i^{th} time step. We define γ as a perturbation factor that controls the amount of stochasticity. ξ is an operator that produces a vector of random variables of the same dimensions as the activation. *reaction* is a general function that determines how the current activations will react with respect to the activations of the previous time step. Finally, s_t is the vector of values of the post-stochastic operation.

Let us break down each of the terms in the generalized equation of stochasticity in the layers of the neural network. γ is the perturbation factor that determines the amount of randomness to be infused in the activations. *reaction* is a function that determines the direction to move based on the current activation values and the previous post-stochastic operation values. If we define ξ to be an operator that produces a vector of IIDs as a probability, i.e $0 < X < 1, \forall X \in \xi$, then we can interpret each neuron as having its own probability of absorbing randomness.

In determining the reaction function, we only include two parameters that are h_t and s_{t-1} . This choice is more suitable and intuitive due to the Markov property exhibited by the financial markets. This implies that given the prior stochastic-activation s_{t-1} , the current stochastic-activation s_t is independent of the other past activations. Thus, we model the *reaction* function as the difference between current activations and previous activations, showing the direction in which to move.

$$reaction(h_t, s_{t-1}) = h_t - s_{t-1} \quad (15)$$

Therefore from Eq. (15) and Eq. (16),

$$s_t = h_t + \gamma \xi_t (h_t - s_{t-1}) \quad (16)$$

Eq. (16) can be thought of as a random-like walk that takes into account the pattern of reaction of the market in progressive time steps.

In a continuously evolving market, it is of utmost importance that the direction of movement corresponds to the pattern that has been observed in the recent time steps as opposed to initial time steps. The pattern should be adapting to changes in market reaction. Here, we show how this formulation gives priority to recent activations over older activations. Eq. (16) for a system at time step, t can be written as follows.

$$s_t = (1 + \gamma \xi_t) h_t - \gamma \xi_t (s_{t-1}) \quad (17)$$

$$s_t = (1 + \gamma \xi_t) h_t - \gamma \xi_t ((1 + \gamma \xi_{t-1}) h_{t-1} - \gamma \xi_{t-1} (s_{t-2})) \quad (18)$$

$$s_t = (1 + \gamma \xi_t) h_t - \gamma \xi_t (1 + \gamma \xi_{t-1}) h_{t-1} + \gamma^2 \xi_t \xi_{t-1} (s_{t-2}) \quad (19)$$

Extending this till time, $t = 0$ where $s_0 = 0$, we obtain a general form of the Eq., which can be written as follows.

$$s_t = (1 + \gamma \xi_t) h_t + \sum_{i=1}^{t-1} (-\gamma)^{t-i} (1 + \gamma \xi_i) h_i \prod_{j=i+1}^t \xi_j \quad (20)$$

From the above Eq. (20), we can infer that the first term is given more attention than the second term. We observe that previous activations have exponentially decaying significance because $0 < \gamma, \xi < 1$. Thus, the recent activations have higher priority as compared to the previous ones in determining the direction of the stochastic walk. Implementation of Eq. (16) in a neural network, during runtime is shown in Algorithm 1.

Algorithm 1 describes the forecasting phase of the proposed model where the processed input data (X_t) is fed into the neural network at every timestep. A *reaction* vector is calculated at every layer from the previous timestep's stochastic activation (s_{t-1}) and current timestep's feature activations (h_t). At each layer in the proposed model a random vector (r) is created and multiplied to a perturbation factor (γ). Then, a hadamard product between the *reaction* vector and random vector is added to the current feature activation h_t . The resultant vector is s_t . The s_t of the last layer of the neural network is the predicted price.

B. STOCHASTIC MODELS

At the heart of the stochastic neural network is the stochastic module, which is appended at the output of every layer in a neural network. After performing a matrix multiplication and activation of the layer, the output is passed on to the stochastic module. Along with current activations, the module also takes in the previous timestep's stochastic activations as input. Fig. 5 is a neural network integrated with stochastic modules.

The stochastic module as shown in Fig. 6 consists of 3 core components; the *reaction* submodule, random variable vector generator, and perturbation factor. The *reaction* submodule takes in current timestep's activations and previous timestep's stochastic activations as input. Here, the *reaction* submodule can be any function capable of capturing the market's pattern of reaction to new information. In our model, we used a sim-

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
    
```

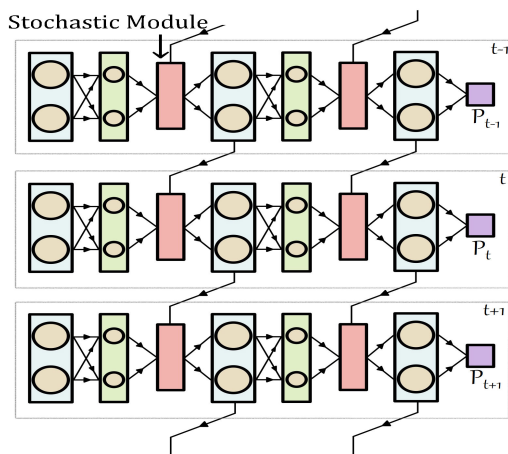


FIGURE 5. Architecture of stochastic neural network.

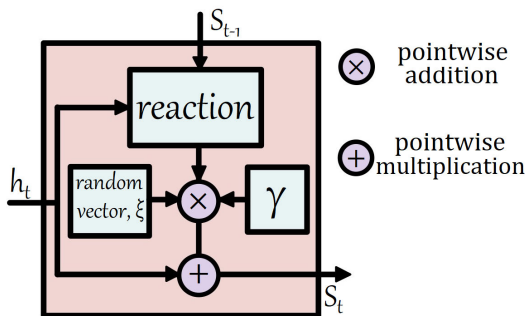


FIGURE 6. Stochastic module.

ple subtraction operation. However, the *reaction* submodule could be a neural network itself.

C. PROPOSED SYSTEM MODEL

To predict the price of a Cryptocurrency, we take 3 essential data sources [31], the first being market statistics. Market statistics include the day low/high and volume. Secondly,

we use blockchain network information that includes hash rate, transaction count, transaction fee, e.t.c. Finally, we use social sentiment information like google trends and tweet volume. All the data from the 3 sources are accumulated into a dataloader. The data features are mean-normalized and N-day data stacks are created. This data is sequentially forwarded to a stochastic neural network which predicts the $(N+1)^{\text{th}}$ day's price. Fig. 7 illustrates the design of the system model.

V. RESULTS AND DISCUSSION

In this section, we describe the features of the data we use to forecast the price of various Cryptocurrencies. Moreover, we explain the preprocessing task of the data. We present the evaluation metrics used to measure the performance of the models and the training process used. Finally, the results obtained by the proposed models are exhibited.

A. DATASET DESCRIPTION

A total of 23 features are used in the proposed models with the window side of 7. We trained the model with the previous data of the past 7 days to predict the price of the eighth day. We trained the model on the data ranging from mid of 2017 to the end of 2019. A total of 850 data points used in the proposed model to extract patterns from the data. We had used data available on bitinfocharts [48].

1) TRANSACTIONS

Number of transactions performed on a given particular day. Cryptocurrency, unlike the share markets, is not listed in any stock exchange. Here, there is no opening time or closing time because there is no regulatory body having power over it. We considered the number of the coins traded in a single day as one of our features.

2) MARKET VOLUME

This is the worth of market movement on a given particular day. It is necessary to know the total units of currency flowing through the market. The quantity of the coins flowing the market is an indicator of the value it possess.

3) MINING DIFFICULTY AND HASHRATE

Mining difficulty is the computational difficulty required to mine a single block of the coin. Mining and confirming transactions requires special hardware. The better the hardware, the more hashes it will hit to mine a block. Thus, there is a tradeoff between the hash rate and power consumption. So, we considered the mining difficulty and hash rate of coin and how much profit is it to mine it.

4) MINING PROFITABILITY

Profitable income to the miner against the use of resources for consuming power and time. With the increase in the number of miners, reward per miner is decreased exponentially.

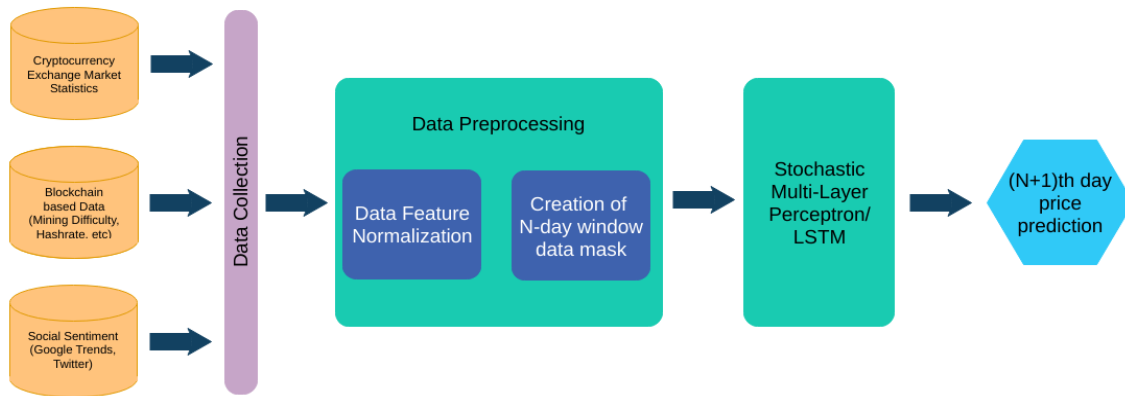


FIGURE 7. The proposed system model.

5) TRANSACTIONS FEE

It is the average transaction fee paid by the parties for the confirmation for their transactions. The fee is required in order to process the transaction in the network.

6) CONFIRMATION TIME

Average time required to confirm the transaction made by one party to another. Transaction made by one party to another party is to be confirmed by all the others and must be logged in the table of blocks. Logging the transaction requires time to confirm the transaction made known as confirmation time (usually around 10 minutes). That time depends upon the currently active users and their geographical location to update their block table.

7) MARKET CAPITALIZATION

Total amount of the Cryptocurrency present in the market on a given particular day in USD.

8) TWEETS AND GOOGLE TRENDS

It is observed that people tend to perform more transactions when tweet volume increases [23], [29]. These are not only correlations but also includes causation with a feedback loop [49]. That correlation does not end with tweet volume. People tend to search more about the trending topics to remain updated about it [50]. Spike in Google search also seemed to have an association with the price of the coins, which too is a trivial assumption [51].

9) HIGHEST AND LOWEST VALUE

The highest and the lowest value a coin had reached on a given particular day. The market for trading Cryptocurrencies remains open for the whole day. So, we considered the peak and the lowest value of the coin attained in a day.

B. DATA PREPROCESSING

We took the utmost care in choosing factors that may affect the price. Some factors that seemed redundant were either removed or removed in the later stage when the trained model was examined. The dataset was normalized in two different

ways. The first way was to normalize the whole dataset by the respective feature means. This dataset is referred to as the **Norm dataset**. The latter one includes the mean-normalization of all features except that of price. This is named as **UNorm dataset**. The reasoning for the same is that the magnitude of the future price is determined by the previous prices and the deviations from the previous price is captured by other features present in the dataset as described above. Accordingly, the Norm and UNorm datasets were further split into train and test. The training dataset is 75%, while the testing is 25% of the whole dataset. This was done for all the three Cryptocurrencies viz. Bitcoin, Ethereum, and Litecoin.

C. EVALUATION METRICS

The trained models are evaluated on the basis of the following different performance metrics. The Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE) are used to assess the proposed models. The formulas for the same are presented below [52].

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|y_t - \hat{y}_t|}{|y_t|} \times 100 \tag{21}$$

$$MAE = \frac{1}{n} \sum_{t=1}^n \frac{|y_t - \hat{y}_t|}{|y_t|} \tag{22}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n \left(\frac{y_t - \hat{y}_t}{y_t} \right)^2} \tag{23}$$

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \tag{24}$$

where y_t is the actual values and \hat{y}_t are the forecasted values.

D. RESULTS

Two prediction model classes were trained viz. Multi-layer perceptron and LSTM were trained on 3 market-dominating Cryptocurrencies i.e. Bitcoin, Ethereum, and

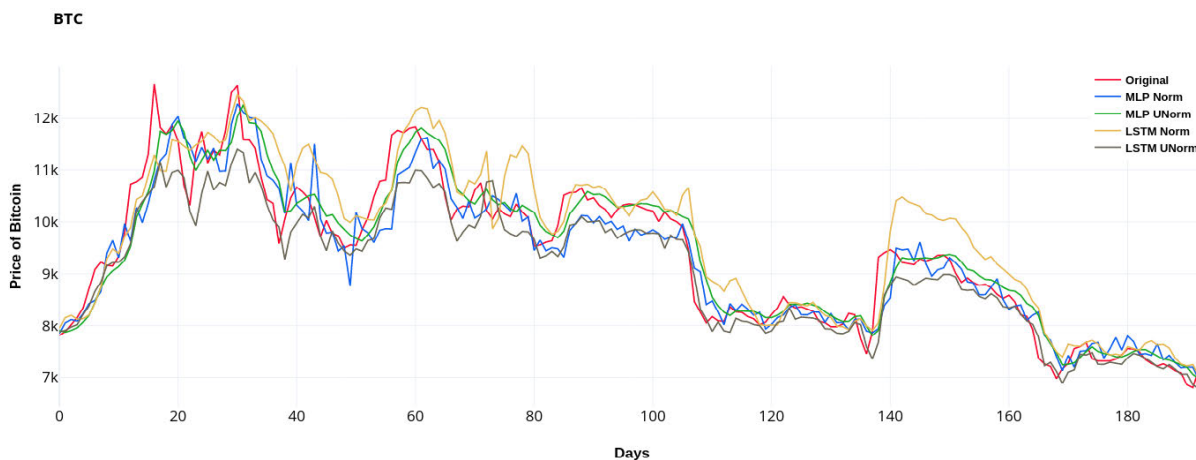


FIGURE 8. Price prediction of Bitcoin using stochastic neural networks.

TABLE 3. Train results of MLP & LSTM on Norm dataset.

Model	Currency	MAPE	MAE	RMSE	MSE
MLP	BTC	4.05496	0.04055	0.05307	149454.95002
	ETH	4.52162	0.04522	0.06010	750.41166
	LTC	4.82956	0.04830	0.06534	36.89689
LSTM	BTC	3.34076	0.03341	0.04477	122179.94284
	ETH	4.57214	0.04572	0.06105	759.24597
	LTC	4.53868	0.04539	0.06205	40.66641

TABLE 4. Train results of MLP & LSTM on UNorm dataset.

Model	Currency	MAPE	MAE	RMSE	MSE
MLP	BTC	4.04184	0.04042	0.05652	269811.44549
	ETH	4.12320	0.04123	0.05548	931.50266
	LTC	4.20070	0.04201	0.05960	60.90442
LSTM	BTC	5.67114	0.05671	0.08116	484755.96467
	ETH	5.14520	0.05145	0.06610	1418.24572
	LTC	4.91972	0.04920	0.07053	146.38440

Litecoin. We trained two model variations for each currency-architecture pair, with the difference in the choice of normalization of data. The first variation has all data features normalized by the feature mean of the train data. The second variation has all features but one, the price normalized. The intuition behind using leaving the price unnormalized is that we wanted the models to obtain the magnitude of the price of the currency directly from data instead of obtaining it from the model.

In this paper, we have used MLP and LSTM models because there is a need of capturing the non-linear dependencies between the market factors, blockchain data and social sentiment and neural networks are most suitable ones for these tasks. MLP is the most basic type of neural network and the LSTM models are most widely used in the scenarios of time dependent data, like the market prices of cryptocurrencies. Thus, for the amalgamation of stochasticity and non-linear dependencies these models were chosen.

1) DETERMINISTIC MODELS

In this section, we present the results obtained by the deterministic models on train and test data. Four models were trained for each currency thus leading to a total of twelve models.

We trained two models on both Norm and UNorm datasets. The architecture of models for all the Cryptocurrencies used is the same as mentioned below.

TABLE 5. Test results of MLP & LSTM on Norm dataset.

Model	Currency	MAPE	MAE	RMSE	MSE
MLP	BTC	4.01775	0.04018	0.06921	491529.85259
	ETH	3.20772	0.03208	0.04699	105.55482
	LTC	3.83638	0.03836	0.05707	20.53836
LSTM	BTC	4.48590	0.04486	0.06083	334129.49767
	ETH	4.01214	0.04012	0.05569	148.65727
	LTC	3.85545	0.03855	0.05468	16.00822

TABLE 6. Test results of MLP & LSTM on UNorm dataset.

Model	Currency	MAPE	MAE	RMSE	MSE
MLP	BTC	3.06223	0.03062	0.04438	185950.34664
	ETH	2.70989	0.02710	0.03900	64.20306
	LTC	2.67675	0.02677	0.04097	12.21042
LSTM	BTC	3.20205	0.03202	0.04406	219261.77088
	ETH	3.48131	0.03481	0.04849	119.59205
	LTC	2.76838	0.02768	0.04068	11.09798

The trained MLP model contains 6 layers, each with the same activation function i.e. ReLU and trained using Adam algorithm for 700 epochs. The input consists of 23 features over the past 7 days, flattened to form a vector of length 161. The layers contain 130, 100, 50, 25, 10, 1 neurons hierarchically in the input to the output direction.

The base sequential model consists of a single LSTM layer. The output of each LSTM unit is fed into a small MLP

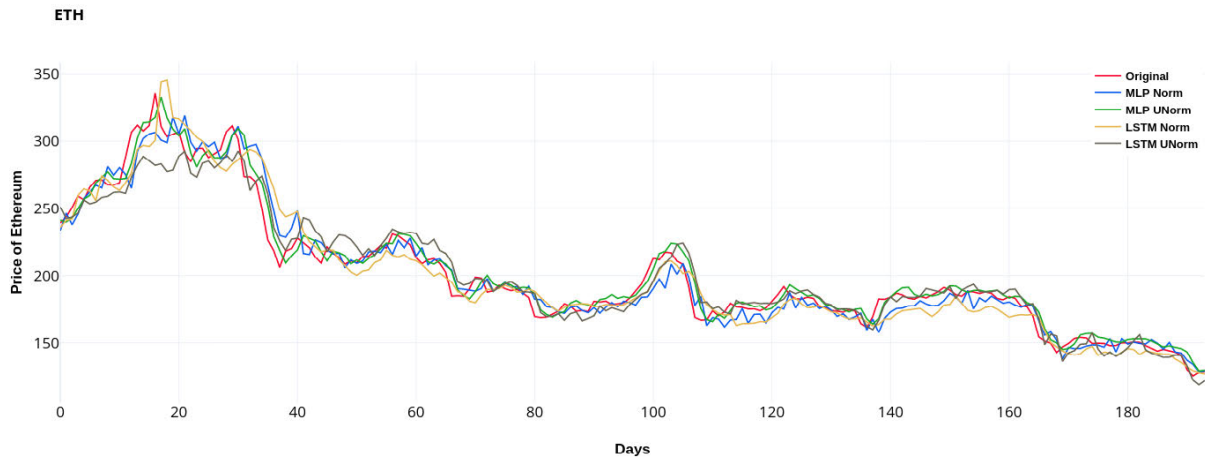


FIGURE 9. Price prediction of Ethereum using stochastic neural networks.

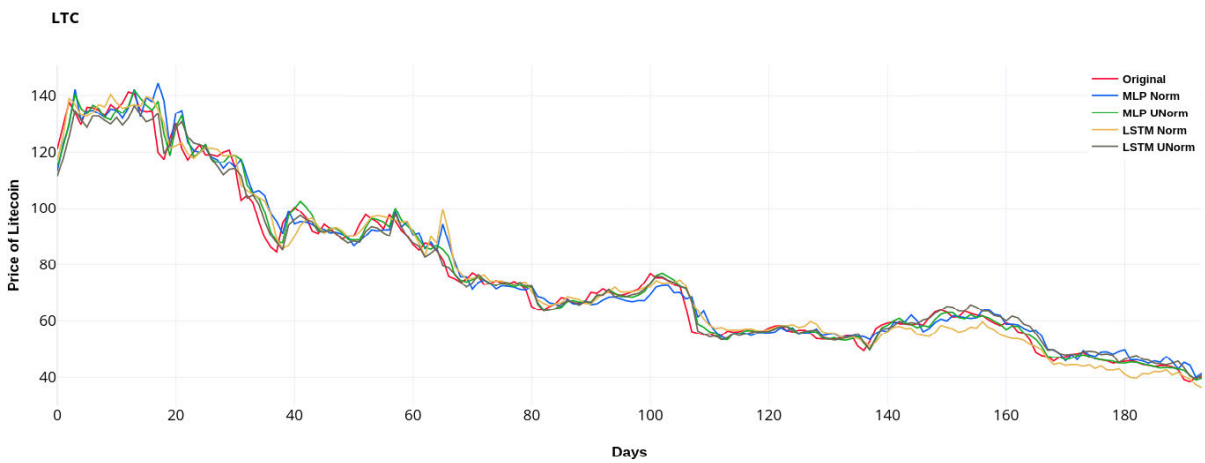


FIGURE 10. Price prediction of Litecoin using stochastic neural networks.

containing 20, 15, 7, 1 neurons respectively, to give the final output of the next day price. All the activations used were ReLU and trained using Adam algorithm for 1500 epochs.

2) STOCHASTIC MODELS

In this section, the results obtained by our models on test data using stochastic layers in the neural networks are presented. The parameters of the trained model remain the same as in the deterministic models.

However, here we test the models using non-zero perturbation factors thus inducing stochasticity in the models. The results are divided by the currency namely Bitcoin, Ethereum and Litecoin. Each run of a stochastic model is known as a realization as referred to in the literature of stochastic processes. To test out our hypothesis, we ran all twelve models with stochasticity activated for 100 realizations to investigate the effect of inducing randomness into the neural network layers.

We demonstrate the probability distribution of MAPE of predicted prices by stochastic neural networks over these realizations. Two perturbation factor values are tested for

each trained model to determine the effect of variation of the value of γ . With a very minute change in the perturbation factor, γ , the error distribution changed disproportionately as shown in Fig. 11. Furthermore, the model performed slightly poor as compared to the deterministic model. We speculate that a stochastic model will perform worse when minor changes in γ lead to major changes in the type of error distribution.

Hyperparameter tuning is the key to choosing the optimal values for γ . Choosing 0.12 over 0.1 significantly improves the performance of our MLP model for Litecoin as shown in Fig. 12. The LSTM model for Litecoin on UNorm dataset showed a better performance with $\gamma = 0.12$, while the MLP model for Ethereum using on UNorm dataset performed better with $\gamma = 0.1$ as shown in Fig. 13. Thus, we show that the magnitude of γ has no correlation with the performance of a stochastic neural net [53].

For every trained model, we see that the MAPE error distribution has no correlation with the value of γ . Therefore, we are posed with the problem of selecting the optimal values of γ for every model. The process for choosing these values

TABLE 7. Test results using stochastic neural networks and the corresponding average relative improvement for Bitcoin.

Bitcoin		$\gamma = 0.10$			$\gamma = 0.12$		γ is learnable	
	Model	Deterministic MAPE	Stochastic MAPE	% Improvement	Stochastic MAPE	% Improvement	Stochastic MAPE	% Improvement
Norm	MLP	4.01775	4.01781	-0.00151	4.01920	-0.03612	4.0189	-0.02774
	LSTM	4.48590	4.38102	2.33794	4.36324	2.73429	4.3699	2.58557
UNorm	MLP	3.06223	2.86007	6.60175	2.82071	7.88717	2.5589	16.43752
	LSTM	3.20205	3.19191	0.31683	3.19500	0.22013	3.1893	0.39799
Average Improvement				2.31375	2.70137		4.84833	

TABLE 8. Test results using stochastic neural networks and the corresponding average relative improvement for Ethereum.

Ethereum		$\gamma = 0.10$			$\gamma = 0.12$		γ is learnable	
	Model	Deterministic MAPE	Stochastic MAPE	% Improvement	Stochastic MAPE	% Improvement	Stochastic MAPE	% Improvement
Norm	MLP	3.20772	3.17348	1.06723	3.18117	0.82764	3.1835	0.75376
	LSTM	4.01214	3.96141	1.26458	3.95320	1.46925	3.9406	1.78415
UNorm	MLP	2.70989	2.55175	5.83583	2.52042	6.99196	2.3405	13.63218
	LSTM	3.48131	3.46424	0.49047	3.46898	0.35436	3.4655	0.45551
Average Improvement				2.16453	2.41080		4.15640	

TABLE 9. Test results using stochastic neural networks and the corresponding average relative improvement for Litecoin.

Litecoin		$\gamma = 0.10$			$\gamma = 0.12$		γ is learnable	
	Model	Deterministic MAPE	Stochastic MAPE	% Improvement	Stochastic MAPE	% Improvement	Stochastic MAPE	% Improvement
Norm	MLP	3.83638	3.74883	2.28208	3.73975	2.51899	3.7398	2.51648
	LSTM	3.85545	3.76282	2.40246	3.74540	2.85422	3.7426	2.92691
UNorm	MLP	2.67675	2.47681	7.46956	2.43774	8.92902	2.3886	10.76640
	LSTM	2.76838	2.69210	2.75539	2.69210	2.75539	2.6916	2.77497
Average Improvement				3.72737	4.26441		4.74619	

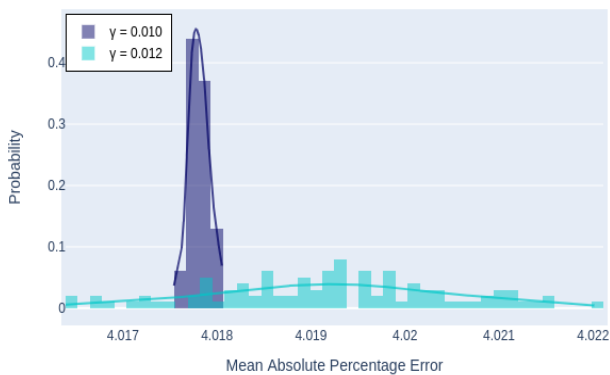


FIGURE 11. Drastic change observed in error distribution of Bitcoin over LSTM on minute change in γ .

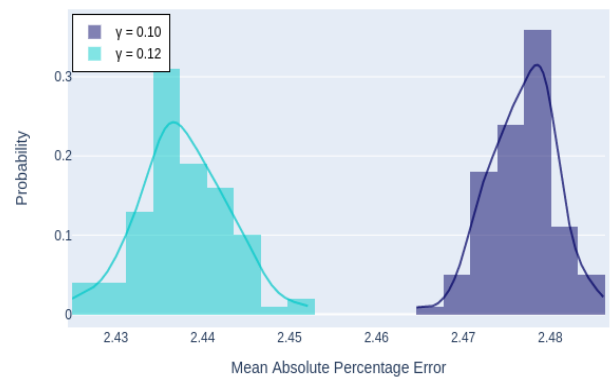


FIGURE 12. Error distribution of Litecoin on MLP.

manually is time-consuming and there is no guarantee for finding the optimum solution.

We can speed up this task by treating γ as a learnable parameter. After training the deterministic neural network, we can freeze all parameters except γ and optimize the values of γ by error backpropagation in combination with gradient descent. Another advantage of doing this is that each layer has its individual value of γ , and hence the optimization

process picks up the maximum value of γ that the individual layers can tolerate. By learning the values of γ , we observe a significant improvement in model performance as shown in tables 7, 8 and 9.

VI. CONCLUSION

In this paper, we predict the price of Cryptocurrencies by using a stochastic neural network model. We introduced a technique to adaptively learn the pattern of the market's

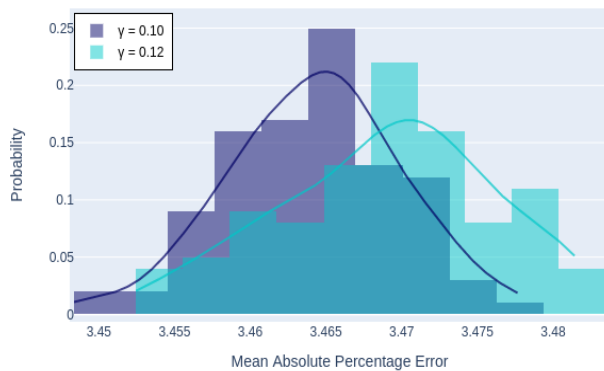


FIGURE 13. Error distribution of Ethereum on LSTM.

reaction to any updated information. The results show that the proposed hypothesis was not only valid but effective in decrypting market volatility. Almost all of the stochastic versions of the neural net models outperformed the deterministic versions. The average relative improvement by using stochastic neural networks over regular neural networks is 1.56% in the Norm dataset at $\gamma = 0.1$, 1.73% at $\gamma = 0.12$ and 1.76% when γ is set as a learnable parameter. The improvement is much more significant for the UNorm dataset where the average relative improvement is 3.91% at $\gamma = 0.1$, 4.52% at $\gamma = 0.12$ and 7.41% when γ is set as a learnable parameter.

In future, it may be worth exploring an optimizing technique to tune the hyperparameter, γ to find its most suitable value. Moreover, alternate *reaction* functions can be tested to learn the pattern of the market's reaction to fresh data. These functions can themselves be stochastic in nature to better simulate market volatility. In addition to that, there are a lot of unexplored territories in the cross-disciplinary field of stochastic processes and neural networks that can be exploited in Cryptocurrency markets.

REFERENCES

- [1] I. Messina, "The effects of Chavez's populism on Today's venezuela," *Perceptions*, vol. 5, no. 1, pp. 1–7, 2019.
- [2] A. A. Z. Hudaib, "E-payment security analysis in depth," *Int. J. Comput. Sci. Secur.*, vol. 8, no. 1, p. 14, 2014.
- [3] J. Vora, A. Nayyar, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and J. J. P. C. Rodrigues, "BHEEM: A blockchain-based framework for securing electronic health records," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2018, pp. 1–6.
- [4] P. Mehta, R. Gupta, and S. Tanwar, "Blockchain envisioned UAV networks: Challenges, solutions, and comparisons," *Comput. Commun.*, vol. 151, pp. 518–538, Feb. 2020.
- [5] S. Tanwar, Q. Bhatia, P. Patel, A. Kumari, P. K. Singh, and W.-C. Hong, "Machine learning adoption in blockchain-based smart applications: The challenges, and a way forward," *IEEE Access*, vol. 8, pp. 474–488, Dec. 2020.
- [6] M. J. Casey and P. Vigna, "Bitcoin and the digital-currency revolution," *The Wall Street J.*, vol. 23, pp. 1–8, Feb. 2015.
- [7] *Coinmarketcap*. Accessed: Mar. 13, 2020. [Online]. Available: <https://coinmarketcap.com/charts/>
- [8] S. Nakamoto. (2008). *Bitcoin: A Peer-To-Peer Electronic Cash System*. [Online]. Available: <http://bitcoin.org/bitcoin.pdf>
- [9] R. Ali, J. Barreard, R. Clews, and J. Southgate, "The economics of digital currencies," in *Proc. Bank England Quart. Bull.*, 2014, p. 3.
- [10] T. Klein, P. T. Hien, and T. Walther, "Bitcoin is not the new gold: A comparison of volatility, correlation, and portfolio performance," *Int. Rev. Financial Anal.*, vol. 59, pp. 105–116, Oct. 2018.
- [11] G. Gajardo, W. D. Kristjanpoller, and M. Minutolo, "Does bitcoin exhibit the same asymmetric multifractal cross-correlations with crude oil, gold and DJIA as the euro, great british pound and yen?" *Chaos, Solitons Fractals*, vol. 109, pp. 195–205, Apr. 2018.
- [12] P. Thakkar, K. Varma, V. Ukani, S. Mankad, and S. Tanwar, "Combining user-based and item-based collaborative filtering using machine learning," in *Information and Communication Technology for Intelligent Systems*, S. C. Satapathy and A. Joshi, Eds. Singapore: Springer, 2019, pp. 173–180.
- [13] P. Bhattacharya, S. Tanwar, U. Bodke, S. Tyagi, and N. Kumar, "BinDaaS: Blockchain-based deep-learning as-a-Service in healthcare 4.0 applications," *IEEE Trans. Netw. Sci. Eng.*, early access, Dec. 25, 2019, doi: 10.1109/TNSE.2019.2961932.
- [14] H. Vachhani, M. S. Obaidat, A. Thakkar, V. Shah, R. Sojitra, J. Bhatia, and S. Tanwar, "Machine learning based stock market analysis: A short survey," in *Innovative Data Communication Technologies and Application*, J. S. Raj, A. Bashar, and S. R. J. Ramson, Eds. Cham, Switzerland: Springer, 2020, pp. 12–26.
- [15] R. Gupta, S. Tanwar, S. Tyagi, and N. Kumar, "Machine learning models for secure data analytics: A taxonomy and threat model," *Comput. Commun.*, vol. 153, pp. 406–440, Mar. 2020.
- [16] J. Liew, R. Li, T. Budavári, and A. Sharma, "Cryptocurrency investing examined," *J. Brit. Blockchain Assoc.*, vol. 2, no. 2, pp. 1–12, Nov. 2019.
- [17] G. Li, Q. Zhao, M. Song, D. Du, J. Yuan, X. Chen, and H. Liang, "Predicting global computing power of blockchain using cryptocurrency prices," in *Proc. Int. Conf. Mach. Learn. Cybern. (ICMLC)*, Jul. 2019, pp. 1–6.
- [18] R. Āšlepaczuk and M. Zenkova, "Robustness of support vector machines in algorithmic trading on cryptocurrency market," *Central Eur. Econ. J.*, vol. 5, no. 52, pp. 186–205, Aug. 2019.
- [19] S. Tanwar, T. Ramani, and S. Tyagi, "Dimensionality reduction using PCA and SVD in big data: A comparative case study," in *Future Internet Technology Trends*, Z. Patel and S. Gupta, Eds. Cham, Switzerland: Springer, 2018, pp. 116–125.
- [20] S. K. L. S. K., A. Khanna, S. Tanwar, J. J. P. C. Rodrigues, and N. R. Roy, "Alzheimer detection using group grey wolf optimization based features with convolutional classifier," *Comput. Electr. Eng.*, vol. 77, pp. 230–243, Jul. 2019.
- [21] M. Saad and A. Mohaisen, "Towards characterizing blockchain-based cryptocurrencies for highly-accurate predictions," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 1–7.
- [22] M. Saad, J. Choi, D. Nyang, J. Kim, and A. Mohaisen, "Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions," *IEEE Syst. J.*, vol. 14, no. 1, pp. 321–332, Mar. 2020.
- [23] A. Mittal, V. Dhiman, A. Singh, and C. Prakash, "Short-term bitcoin price fluctuation prediction using social media and Web search data," in *Proc. 12th Int. Conf. Contemp. Comput. (IC)*, Aug. 2019, pp. 1–6.
- [24] D. Enke and S. Thawornwong, "The use of data mining and neural networks for forecasting stock market returns," *Expert Syst. Appl.*, vol. 29, no. 4, pp. 927–940, Nov. 2005.
- [25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.
- [26] E. Sin and L. Wang, "Bitcoin price prediction using ensembles of neural networks," in *Proc. 13th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Jul. 2017, pp. 666–671.
- [27] H. Jang and J. Lee, "An empirical study on modeling and prediction of bitcoin prices with Bayesian neural networks based on blockchain information," *IEEE Access*, vol. 6, pp. 5427–5437, 2018.
- [28] M. Jordan, "Serial order: A parallel distributed processing approach," *Inst. Cogn. Sci.*, Tech. Rep. 8904, Jun. 1986.
- [29] N. Smuts, "What drives cryptocurrency prices?" *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 131–134, Jan. 2019.
- [30] L. Alessandretti, A. ElBahrawy, L. M. Aiello, and A. Baronchelli, "Anticipating cryptocurrency prices using machine learning," *Complexity*, vol. 2018, pp. 1–16, Nov. 2018.
- [31] Z. Chen, C. Li, and W. Sun, "Bitcoin price prediction using machine learning: An approach to sample dimension engineering," *J. Comput. Appl. Math.*, vol. 365, Feb. 2020, Art. no. 112395.
- [32] S. Lahmiri and S. Bekiros, "Cryptocurrency forecasting with deep learning chaotic neural networks," *Chaos, Solitons Fractals*, vol. 118, pp. 35–40, Jan. 2019.
- [33] M. T. Kurbcuz, "Predicting the price of bitcoin by the most frequent edges of its transaction network," *Econ. Lett.*, vol. 184, Nov. 2019, Art. no. 108655.

- [34] D. Kondor, I. Csabai, J. Szále, M. Pósfai, and G. Vattay, "Inferring the interplay between network structure and market effects in bitcoin," *New J. Phys.*, vol. 16, no. 12, 2014, Art. no. 125003.
- [35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [36] W. Yiyang and Z. Yeze, "Cryptocurrency price analysis with artificial intelligence," in *Proc. 5th Int. Conf. Inf. Manage. (ICIM)*, Mar. 2019, pp. 97–101.
- [37] J. Rebane, "Seq2Seq RNNs and ARIMA models for cryptocurrency prediction : A comparative study,"
- [38] C. J. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proc. ICWSM*, 2014, pp. 1–16.
- [39] Y. Peng, P. H. M. Albuquerque, J. M. Camboim de Sá, A. J. A. Padula, and M. R. Montenegro, "The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression," *Expert Syst. Appl.*, vol. 97, pp. 177–192, May 2018.
- [40] G. T. Friedlob and F. J. Plewa, Jr., *Understanding Return on Investment*. Hoboken, NJ, USA: Wiley, 1996.
- [41] E. F. Fama, "Random walks in stock market prices," *Financial Anal. J.*, vol. 21, no. 5, pp. 55–59, Sep. 1965.
- [42] B. G. Malkiel, *A Random Walk Down Wall Street: Including a Life-Cycle Guide to Personal Investing*. New York, NY, USA: W. W. Norton & Company, 1973.
- [43] D. Vasan, M. Alazab, S. Wassan, B. Safaei, and Q. Zheng, "Image-based malware classification using ensemble of CNN architectures (IMCEC)," *Comput. Secur.*, vol. 92, May 2020, Art. no. 101748.
- [44] D. Mungra, A. Agrawal, P. Sharma, S. Tanwar, and M. S. Obaidat, "PRATIT: A CNN-based emotion recognition system using histogram equalization and data augmentation," *Multimedia Tools Appl.*, vol. 79, nos. 3–4, pp. 2285–2307, Jan. 2020.
- [45] A. C. Logue, "The efficient market hypothesis and its critics," *CFA Dig.*, vol. 33, no. 4, pp. 40–41, Nov. 2003.
- [46] W. Brock, J. Lakonishok, and B. LeBARON, "Simple technical trading rules and the stochastic properties of stock returns," *J. Finance*, vol. 47, no. 5, pp. 1731–1764, Dec. 1992.
- [47] R. U. Khan, X. Zhang, M. Alazab, and R. Kumar, "An improved convolutional neural network model for intrusion detection in networks," in *Proc. Cybersecurity Cyberforensics Conf. (CCC)*, May 2019, pp. 74–77.
- [48] *Bitinfocharts*. Accessed: May 1, 2020. [Online]. Available: <https://bitinfocharts.com/>
- [49] D. Garcia, C. J. Tessone, P. Mavrodiev, and N. Perony, "The digital traces of bubbles: Feedback cycles between socio-economic signals in the bitcoin economy," *J. Roy. Soc. Interface*, vol. 11, no. 99, Oct. 2014, Art. no. 20140623.
- [50] S. Rahman, J. N. Hemel, S. Junayed Ahmed Anta, H. A. Muhee, and J. Uddin, "Sentiment analysis using R: An approach to correlate cryptocurrency price fluctuations with change in user sentiment using machine learning," in *Proc. Joint 7th Int. Conf. Informat., Electron. Vis. (ICIEV) 2nd Int. Conf. Imag., Vis. Pattern Recognit. (icIVPR)*, Jun. 2018, pp. 492–497.
- [51] L. Kristoufek, "BitCoin meets Google trends and wikipedia: Quantifying the relationship between phenomena of the Internet era," *Sci. Rep.*, vol. 3, no. 1, p. 3415, Dec. 2013.
- [52] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 2162–2172, Mar. 2015.
- [53] D. Vasan, M. Alazab, S. Wassan, H. Naeem, B. Safaei, and Q. Zheng, "IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture," *Comput. Netw.*, vol. 171, Apr. 2020, Art. no. 107138.



VASU KALARIYA is currently pursuing the bachelor's degree with Nirma University, Ahmedabad, Gujarat, India. His research interests are computer vision, energy-based models, reinforcement learning, and quantum machine learning.



PUSHPENDRA PARMAR is currently pursuing the bachelor's degree from Nirma University, Ahmedabad, Gujarat, India. His research interests are computer vision, probabilistic graphical models, and quantum machine learning.



SUDEEP TANWAR (Member, IEEE) received the B.Tech. degree from Kurukshetra University, India, in 2002, the M.Tech. degree (Hons.) from Guru Gobind Singh Indraprastha University, Delhi, India, in 2009, and the Ph.D. degree with specialization in wireless sensor network in 2016. He is currently an Associate Professor with the Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad, Gujarat, India. He is also a Visiting Professor with Jan Wyzkowski University, Polkowice, Poland, and also with the University of Pitesti, Pitesti, Romania. He has authored or coauthored more than 130 technical research articles published in leading journals and conferences from the IEEE, Elsevier, Springer, Wiley, and so on. Some of his research findings are published in top cited journals such as the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Computer Communication*, *Applied Soft Computing*, the *Journal of Network and Computer Application*, *Pervasive and Mobile Computing*, the *International Journal of Communication System, Telecommunication System, Computer and Electrical Engineering*, and the IEEE SYSTEMS JOURNAL. He has also published six edited/authored books with International/National Publishers like IET, Springer. He has guided many students leading to M.E./M.Tech. and guiding students leading to Ph.D. His current interests include wireless sensor networks, fog computing, smart grid, the IoT, and blockchain technology. He has been awarded best research paper awards from IEEE GLOBECOM 2018, IEEE ICC 2019, and Springer ICRIC-2019. He was invited as a Guest Editors/Editorial Board Members of many International Journals, invited for a Keynote Speaker in many international conferences held in Asia and invited as a Program Chair, a Publications Chair, a Publicity Chair, and a Session Chair in many International Conferences held in North America, Europe, Asia, and Africa. He is an Associate Editor of *IJCS* (Wiley) and *Security and Privacy Journal* (Wiley).



PATEL JAY is currently pursuing the bachelor's degree from Nirma University, Ahmedabad, India. His research interests include computer vision, natural language processing, energy-based models, and reinforcement learning.



NEERAJ KUMAR (Senior Member, IEEE) received the Ph.D. degree in Computer Science Engineering from Shri Mata Vaishno Devi University, Katra (J&K), India. He was a Postdoctoral Research Fellow with Coventry University, Coventry, U.K. He is currently a Full Professor with the Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Deemed to be University, Patiala, India. He is also a Visiting Professor with Coventry

University, Coventry, U.K. He has published more than 300 technical research articles in leading journals and conferences from IEEE, Elsevier, Springer, John Wiley, and so on. Some of his research findings are published in top cited journals such as the IEEE TIE, IEEE TDSC, IEEE TITS, IEEE TCC, IEEE TKDE, IEEE TVT, IEEE TCE, IEEE NETWORKS, IEEE COMMUNICATION, IEEE WC, IEEE IoTJ, IEEE SJ, FGCS, JNCA, and ComCom. He has guided many Ph.D. and M.E./M.Tech. His research is supported by funding from Tata Consultancy Service, Council of Scientific and Industrial Research (CSIR), and Department of Science & Technology. He has awarded best research paper awards from IEEE ICC 2018 and the IEEE SYSTEMS Journal 2018. He is leading the research group Sustainable Practices for Internet of Energy and Security (SPINES), where group members are working on the latest cutting edge technologies. He is a TPC Member and a Reviewer of many international conferences across the globe.



MAMOUN ALAZAB (Senior Member, IEEE) received the Ph.D. degree in Computer Science from the School of Science, Information Technology and Engineering, Federation University of Australia. He is currently an Associate Professor with the College of Engineering, IT and Environment, Charles Darwin University, Australia. He is also a Cyber Security Researcher and a Practitioner with industry and academic experience. He has more than 150 research articles in many

international journals and conferences, such as the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, the IEEE TRANSACTIONS ON BIG DATA, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Computers & Security*, and *Future Generation Computing Systems*. He delivered many invited and keynote speeches, 24 events in 2019 alone. His research is multidisciplinary that focuses on cyber security and digital forensics of computer systems with a focus on cybercrime detection and prevention. He convened and chaired more than 50 conferences and workshops. He works closely with government and industry on many projects, including Northern Territory (NT) Department of Information and Corporate Services, IBM, Trend Micro, the Australian Federal Police (AFP), Westpac, and the Attorney Generals Department. He is the Founding Chair of the IEEE Northern Territory (NT) Subsection.

• • •