IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 24, 2021, accepted December 11, 2021, date of publication December 14, 2021,
date of current version December 29, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3135620

An On-Chain Analysis-Based Approach to Predict Ethereum Prices

NISHANT JAGANNATH[1], (Member, IEEE), TUDOR BARBULESCU[1], KARAM M. SALLAM[1], (Member, IEEE), IBRAHIM ELGENDI[1], (Member, IEEE), BRADEN MCGRATH[2], ABBAS JAMALIPOUR[3], (Fellow, IEEE), MOHAMED ABDEL-BASSET[4], (Senior Member, IEEE), AND KUMUDU MUNASINGHE[1], (Senior Member, IEEE)

[1]School of Information Technology and Systems, University of Canberra, Canberra, ACT 2601, Australia
[2]Faculty of Human Factors and Behavioral Neurobiology, Embry–Riddle Aeronautical University, Oklahoma City, OK 73135, USA
[3]School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2006, Australia
[4]Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt

Corresponding author: Nishant Jagannath (nishant.jagannath@canberra.edu.au)

ABSTRACT The Ethereum blockchain generates a significant amount of data due to its intrinsic transparency and decentralized nature. It is also referred to as on-chain data and is openly accessible to the world. Moreover, the on-chain data is timestamped, integrated, and validated into an open ledger. This important blockchain feature enables us to assess the network's health and usage. It serves as a massive data warehouse for complex prediction algorithms that can effectively detect systemic trends and forecast future behavior. We adopt a quantitative approach using a subset of these metrics to determine the network's true monetary value by developing a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) with the metrics most closely associated with the price as inputs. Since several hyperparameters regulate the learning process in an RNN, they are highly sensitive to their values. It is thus critical, to select optimal hyperparameters so that the training is quick and effective. Determining the optimal parameters of an RNN model is a tedious and complex process. Hence, previous studies have developed several self-adaptive approaches to determine the optimal values for various parameters effectively. However, none of the prior studies explore self-adaptive algorithms in deep learning models in conjunction with on-chain data to predict cryptocurrency prices. In this paper, we propose three self-adaptive techniques, each of which converges on a set of optimal parameters to predict the price of Ethereum accurately. We compare our results to a traditional LSTM model. Our approach exhibits 86.94% accuracy while maintaining a minimum error rate.

INDEX TERMS Blockchain, deep neural network, differential evolution, Ethereum, evolutionary algorithms, jSO, LSTM.

I. INTRODUCTION

Blockchain is proving to be one of the most significant technological advancements in the public and private sectors. According to Australia's National Blockchain Roadmap proposed by the Department of Industry, Science, Energy and Resources, by 2025, blockchain technology is expected to generate over US 175 billion in annual business value and further grow to over US 3 trillion in 2030 [1]. To this end, the concept of blockchain technology has emerged to address the limitations of a centralized architecture, primarily the single point of failure alongside addressing security issues. The ability of blockchain to securely store and transfer digital assets in a decentralized, distributed, and tamper-proof way is a quality that many researchers attribute to it as a means of conducting transactions. Beyond cryptocurrencies, several applications use blockchain technology, such as electronic health care and identity management systems. [2].

Bitcoin [3], the most well-known use of blockchain technology to date, was introduced in 2008 through a paper written under the pseudonym, Satoshi Nakamoto. Bitcoin's success stems from the fact that it was the first digital currency to overcome the double-spending and the "Byzantine General" problems, allowing for a secure way

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

of transferring value online without relying on a trusted third party. The advent of other cryptocurrencies has given rise to a new asset class. Bitcoin provided a safe and inexpensive way to move currency across a decentralized, peer-to-peer network.

Though Bitcoin was the first currency to gain traction, the concept of digital currency has been around for decades. Since the 1990's Cypher-punks have been experimenting with the notion of using an alternative digital currency entirely independent of any financial institutions and government organisations issuing it. Some of the initial projects that solved cryptographic puzzles to create a concept for cryptocurrencies were b-money [4], bit gold [5] and reusable proof of works [6]. Nonetheless, these projects failed primarily as a result of their reliance on a centralized intermediary [7].

Bitcoin is the world's first decentralized cryptocurrency, established in 2009 with the creation of the genesis block. Alternative cryptocurrencies (also known as altcoins) were developed since 2011, with several of them continuing in use today. The majority of cryptocurrencies are forks of Bitcoin, meaning they were developed using the same codebase as Bitcoin, hence the name "**alt**ernative **coin**". Ethereum is the second most popular blockchain platform after Bitcoin, with a market cap of 267 billion dollars [8]. It permits decentralized money transmission and the design and participation in rule-based smart contracts (code) that run on a decentralized infrastructure comparable to Bitcoin. In 2013, virtually every week saw the birth of new altcoins, resulting in the growth of an extensive, highly speculative trading market [9].

Investors often use one of three approaches to evaluate a crypto asset: a technical, fundamental, or quantitative analysis. Technical analysis is the most frequently utilized method as a directional hypothesis in bitcoin trading. However, despite claims that profits from the technical analysis have declined over time, research reveals that profits in the foreign currency markets have grown considerably [10]. Even today, several investors still base their investing decisions on technical indicators since they focus on profitability via statistics and data volume. It is difficult to anticipate the price of a crypto asset using technical analysis since the indicators change rapidly in the cryptocurrency market. Researchers have used machine learning algorithms combined with technical indications to predict the price of crypto assets, particularly Bitcoin, to solve these difficulties [11]. A recent study discovered a link between the price of Bitcoin, Google, and Wikipedia [12]. Several studies used blockchain data such as transaction volume, hash rate, and difficulty to forecast bitcoin prices [13]–[15].

The research above has focused on modelling machine learning algorithms to predict Ethereum pricing without fully contextualizing the characteristics of the on-chain data and their potential impact on Ethereum prices. Furthermore, unpredictable swings in cryptocurrency prices due to various factors such as supply and demand, inflation, and political factors are significant barriers to accurate price prediction. This presents many opportunities to realise the value and information of on-chain data related to Ethereum that has been overlooked while utilising more advanced optimization algorithms to better adapt to the volatility associated with Ethereum prices.

This article is organised as follows. Section II provides an overview of relevant work, emphasising current research on Ethereum price prediction. Section III covers the research's proposed methodology and is divided into four subsections, including subsection A, B, C and D. Subsection A describes data collection methods, data characteristics, and analysis measures. We study on-chain events and their effects on price, correlation analysis with the price to understand price patterns better. Subsection B describes the architecture of an LSTM model, while subsection C introduces differential evolution algorithms, covers the three self-adaptive techniques and justifies their use. Subsection D describes the proposed system model. Section IV describes the experimental setup, whereas Section V presents the results and their interpretation. Finally, in Section VI, we summarise our findings, conclusions and outline areas for further research.

II. RELATED WORK

Technical Analysis is a topic that has been extensively explored in academic literature. Although simple to implement and interpret, traditional statistical methods require many statistical assumptions that could be unrealistic, leaving machine learning as the best technology capable of predicting price based on historical data, among others [16]. Numerous studies have examined the effectiveness of various machine learning techniques and technical indicators in predicting bitcoin values [11], [17]. The problem with cryptocurrencies is that their value is dependent on unpredictable market movements and social sentiment. Moreover, cryptocurrencies, primarily Bitcoin, have a low correlation with major financial assets as seen in [18], [19], indicating that conventional economic theories and models are insufficient for forecasting the volatility connected with the price of cryptocurrencies.

There has been much research interest in Bitcoin in the last decade, whereas only a few studies have investigated Ethereum. Ethereum is one of the most popular blockchain platforms after Bitcoin. It enables users to use blockchain-based applications with a smart contract to be easily deployed in a completely decentralized way. In 2016, a Decentralized Autonomous Organization (DAO) attack on Ethereum resulted in a loss of millions of dollars, more here [20]. Smart contract analysis can help detect the similarity of codes and calls, vulnerability, and fraud detection in contracts. Securify [21] was released publicly and has analyzed over eighteen thousand contracts for vulnerabilities to address this issue. Recent studies such as [22], [23] have proposed classification methods to detect frauds on smart contracts by analysing the data available on Ethereum, primarily Ponzi schemes.

TABLE 1. Correlation coefficient values between Bitcoin and other major financial assets between 2016 to 2020 [18].

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

Since Ethereum is a permission-less blockchain, it uses an open data structure. Therefore, data analysis of the data on the blockchain can provide tools that can further enable understanding of the underlying network and its usage, providing valuable information on the user, miner behavior and other activities on the blockchain [24]. A few studies explored the blockchain information to understand the characteristics and inherent usage of the network for Ethereum. For example, authors in [25] investigate the transactions data on the blockchain to understand the behavioral traits of addresses on the Ethereum Network. A study into the statistical characteristics of Ethereum pricing utilising blockchain information, mainly using Ethereum network features, determined that hash rate, difficulty, and transaction cost are strongly associated with price [14]. Nevertheless, the public Ethereum blockchain data is vast and has expanded exponentially over time, reaching 248.93GB in 2021 [26]. Parsing this large volume of data is time-consuming. To address this issue, several researchers have proposed open-source frameworks to extract crucial information from popular permissionless blockchain platforms such as Bitcoin and Ethereum [27]–[29]. A recent study proposed a framework that focuses on collecting and analysing datasets from Ethereum and released a series of datasets for developers and researchers to analyze user behavior and other blockchain system operations [30].

Smuts [31] adopted a LSTM model to predict short-term trends in cryptocurrency prices by investigating the influence of Telegram and Google Trends search analysis on the price of Ethereum. The above study achieved an accuracy rate of 56% for Ethereum and 63% for Bitcoin and concluded that Telegram proved to be a better indicator of the price of bitcoin and Ethereum. The machine learning model trained using historical data from non-blockchain sources, such as social media feeds and publicly accessible websites, may cause the model to make incorrect assumptions due to the datasets inherent bias [32], [33], particularly when the model is converged with a blockchain system. Since the data on the blockchain is timestamped, embedded, and vetted into an open ledger, this creates a massive data warehouse for advanced predictive models to identify trends and fraud in the system accurately. This is possible as the data on the blockchain is immutable. This important characteristic of blockchain has gained tremendous interest among researchers in using advanced predictive models with blockchain (on-chain) data.

Conversely, only a few studies have adopted various machine learning approaches using blockchain information to predict the price trends for Ethereum. Authors in [14] adopted a multivariate regression model using features of the Ethereum network to explain the Ethereum dynamics further, achieving a mean absolute error (MAE) of 0.0563 for Ethereum with 50% test data. A recent study proposed a stochastic neural network model utilising a perturbation factor (gamma) to introduce randomness into the model to simulate market volatility [15]. It achieved an average improvement of 4.15% on mean absolute percentage error (MAPE) for Ethereum compared to Multi-Layer Perceptron (MLP) and LSTM models.

Evolutionary Algorithms (EA) have been widely used to solve optimization problems in several fields and real-world applications. The term ''optimization'' refers to the process of identifying the most appropriate solution to a problem given some constraints. Portfolio optimization has traditionally relied on metaheuristics. The utilisation of EA for Multi-Objective Portfolio Management is reviewed in [34]. Evolutionary optimisers for continuous parameter spaces, such as Differential Evolution (DE) [35]–[38], [39], developed by Storn and Price, are becoming increasingly powerful and flexible. As a result, DE and its variations have established themselves as one of the most competitive and flexible groups of evolutionary computing algorithms. They have been effectively used to address a wide range of real-world issues in science and industry [40], [41]. However, none of the above studies investigates the application of DE algorithms for hyperparameter optimization in deep learning models in conjunction with on-chain data to predict cryptocurrency prices.

The above studies use a few metrics, such as hash rate, difficulty, transaction count, among other metrics, to identify price trends in Ethereum. This paper adopts a comprehensive approach by using on-chain datasets to help us better understand core activities related to blockchain. In comparison, researchers have used LSTM, Random Forest and Stochastic models, among other techniques, to predict Ethereum price. In our previous work, we proposed a self-adaptive technique for predicting bitcoin prices and achieved a higher accuracy rate compared to a traditional LSTM model [42]. This study further explores advanced algorithms (self-adaptive techniques) in conjunction with deep learning models that predict price trends better and adapt to the volatility associated with the Ethereum price.

III. PROPOSED DESIGN AND METHODOLOGY

This section explains the meaning of on-chain metrics and their purpose in the context of blockchain in general and Ethereum in particular. Additionally, this section discusses the methodologies utilized to analyze the information to

understand better the blockchain network's activity, which enables us to develop a prediction model.

A. ON-CHAIN ANALYSIS

1) WHAT ARE ON-CHAIN METRICS?

On-chain metrics are data points derived from information generated by the blockchain network, such as the size of the blockchain, the number of blocks attached to it, or the difficulty of mining blocks. Their purpose is to inform interested parties about the state of a blockchain network; this data inherits the properties of these networks, such as their inherent transparency, tamper-resistant and decentralized nature. Since on-chain data is often recorded in a time-series manner, each metric offers insight into the historical activity of a blockchain. As a result, everyone involved with law enforcement, such as agents tracking illegal activity [43] or those in the financial industry determining the viability of a proposed investment in enhanced decision-making, will benefit from this knowledge.

2) ON-CHAIN METRICS AS FINANCIAL TOOLS

The financial sector has already been noted as a potential use case for on-chain data. Comparing blockchain networks to companies, there are various similarities to be observed - most networks have a coin, equivalent to the stock of a company, and both have a price and are traded on exchanges. There are fundamental differences: blockchain networks are not structured like companies, there is no need for executives, and the economic parameters are written in code. The most recent story about Bitcoin is that it is an improved version of Gold [44] for its store of value and non-inflationary properties, coupled with its lower friction of moving assets around.

However, seeing as their popularity as financial instruments are increasing and ever more credible investors are adding them to their portfolios [44], we can ascertain that cryptocurrencies have gained their category as crypto-assets. They are traded on dedicated exchanges and are becoming significant parts of investors' portfolios around the world. Hence, the need arises for tools and protocols to analyze these crypto-assets as financial instruments and decide whether to make them part of one's portfolio or not. On-chain metrics provide the best basis for such tools, as decentralized and tamper-proof data that enables anyone access to in-depth information about a network's usage, with no entity having the ability to censor it. We view this as akin to everybody having access to insider information on any publicly traded company. We consider this could act as a way of levelling the financial playing field and getting us closer to the efficient market hypothesis [45] which states that all the publicly and privately accessible information available to market participants is already reflected in the price.

3) DATA COLLECTION

In this paper, the data has been gathered from the public Ethereum blockchain and application programming interfaces (APIs) of online resources [46], [47] from 2016 through 2021. We analyze metrics with the Ethereum price using on-chain data from these resources. This research aims to provide a broader overview by incorporating on-chain metrics relating to miners, users, and exchange activity and their possible impact on Ethereum pricing. The following metrics are obtained from the on-chain data; block size, block height, transaction count, daily active addresses, miners revenue, miner fees, miner to exchanges, total new addresses, supply in smart contracts, gas price, transactions rate, transfers count, hash rate, transactions gas limit, difficulty, transfer rate, total gas used, wallets address > 1, >10, >100 coins, exchange deposits, exchange withdrawals, external and internal contract calls, total addresses.

4) DATA CHARACTERISTICS

Cryptocurrencies use more accurate valuation methods as opposed to stocks. While some metrics are derived from data generated within the blockchain like block size, others are derived from external and on-chain information - such as the market capitalisation.

Market Cap = Circulating supply $\times$ Last market price

For Bitcoin, the price changes every day as it is traded on various exchanges, and the supply also increases as more coins are mined in circulation. At the time of writing this, the price of BTC is \$49,027.71, and the circulating supply is 18,792,243.00 BTC [48]. The supply of Bitcoin increases and it is capped at 21 million. While not all coins are in circulation (some have yet to be mined), there will never be more than 21 million bitcoins in circulation, with the mining reward halving roughly every four years. Considering that one satoshi is 1×10^{-8} bitcoin, this choice seems arbitrary. For Ethereum, on the other hand, the supply is not yet fixed. As it stands today, there is a circulating supply of 117,189,429 ETH [47], one of which has a price of \$3,254.38.

Market capitalisation is not unique to blockchains, and it is a suboptimal method of valuing a blockchain network. For one, it does not account for the fact that only a few coins have been traded at the latest spot prices, and it also ignores the significant amount of coins irrevocably lost due to lost private keys [49]. Enter realised capitalisation, developed by Carter *et al.* [50], uses the extra information provided by blockchains (coinage) to give a more accurate valuation. The formula for the realised cap is:

Realised Cap = Σ Coin amount $\times$ Price last transacted

In Ethereum, a block contains a block number, difficulty, gas limit, transaction list and the most recent state. A block also holds the answer to a unique, difficult-to-solve mathematical problem; miners gather all the information included in that block before the block is released into circulation. The first miner to solve the puzzle created by this information wins a block reward. Unlike Bitcoin, whose block rewards are halved every 210,000 blocks (about four years), there is no established number of ethers in circulation. The block rewards were adjusted from 3 ETH to 2 ETH after the Constantinople hard fork in 2018 [51].

FIGURE 1. State Transition Function in Ethereum.

Ethereum employs an account-based model to keep track of the state of the transactions in the network, unlike Bitcoin, which uses the Unspent Transactions Output (UTXO) model to calculate the sum of the unspent transactions. The account-based model is similar to bank accounts; the balance is the number of coins left in the account, using which the user can send or receive transactions. Using the UTXO model, it is easier to extract valuable on-chain information such as the age of the coins in circulation, and therefore there are fewer unique on-chain metrics available for Ethereum. However, effective ways have been developed for mapping between the account-based and UTXO models [52] which makes it easier to gain similar insights into the Ethereum network.

Exchange operations must be open and transparent in light of the numerous fraudulent activities that have been investigated in the past [53]. Analysis of exchange on-chain activity, such as deposits and withdrawals, provides a verifiable and immutable data source that might help researchers better understand the influence of exchange operations on pricing.

5) ON-CHAIN METRICS AND THEIR EFFECT ON THE PRICE

We examine a few metrics and critical aspects connected to pricing to illustrate the parallels we observed between metrics and price that led us to investigate this relationship further.

Perhaps the most prominent relationship between an on-chain metric and the pricing can be illustrated by referring to the daily transaction volume. On Ethereum, a transaction is defined as any interaction with the blockchain that changes the blockchain's state. Ethereum uses transaction fees to minimize network overload and a possible attack vector, Bitcoin, and other blockchains. As a result, the transaction count accurately represents the network's underlying activity and demand for Ether, which is also used to pay transaction fees. From 2017 through 2021, we can detect similar patterns in these graphs by plotting the normalized price against the normalized transaction count.

The number of active addresses is another simple metric to conceptualize and extract since all blockchain transactions are timestamped. We can easily find all the transactions that occurred on a specific day, out of which we count the number of individual addresses which initiated and sent transactions. The resulting number approximates the number of people that transacted on the network on a specific day (some participants have several wallets), with more active addresses indicating increased demand for the network's native coin. According to supply and demand theory, periods when more addresses are active than normal should result in a higher price, as seen in Fig. 3. We can also observe that the normalized transaction count graph looks very similar to the normalized active addresses.

On-chain metrics such as active addresses, total addresses and transaction volume indicate the usage and adoption of the network. In comparison, the hash rate indicates the number of computational resources dedicated to the network, which indicates the number of miners committed to this network. Miners are generally financially committed, having invested in equipment, infrastructure and ongoing power consumption, none of which have an instant return on investment. More miners committed to a network means that more people truly believe in the network (by having invested in it long term), but it also means the security of the network increases, making it more valuable. Hence, metrics such as hash rate and mining rate can help us understand the network's overall security.

The above relationships are straightforward to conceptualize, but they are by far the only ones. A recent article [54] outlined several less obvious relationships. We can determine the age of each currency traded on the Bitcoin network by using a UTXO-based ledger. For Ethereum, we do not have this information directly as it has an account-based accounting model but can estimate this information accurately enough for our purposes [52]. The ability to extract a coin's age allows us to group the coins of the entire supply based on the last time they were spent - thus creating a group of coins that were spent less than one day ago, a group last spent between 1 day and one week, one week and one month, 1-3 months, 3-6 months.

Given the number of fraudulent activities reported in the past, exchange operations must be transparent. Analysis of exchange on-chain activity, such as deposits and withdrawals, provides a verifiable and immutable data source that might help researchers better understand the influence of exchange operations on pricing. Supply distribution indicates the distribution of economic resources across the network. A more centralized distribution means that fewer entities have to gain from an increase in the coin's value and that a small number of people have much influence on the price, which might decrease the overall attractiveness of the network as an investment.

Following our exploration of the effects of on-chain metrics on the price of Ethereum, on-chain activity adds value to portfolio managers looking to invest in cryptocurrencies. Nonetheless, we are interested in determining if quantitative traders could incorporate some of these metrics into their trading algorithms. We devise a simple method for choosing a subset of metrics that may have a more significant influence in forecasting price: calculate the price-metric correlations and choose the most strongly associated metrics.

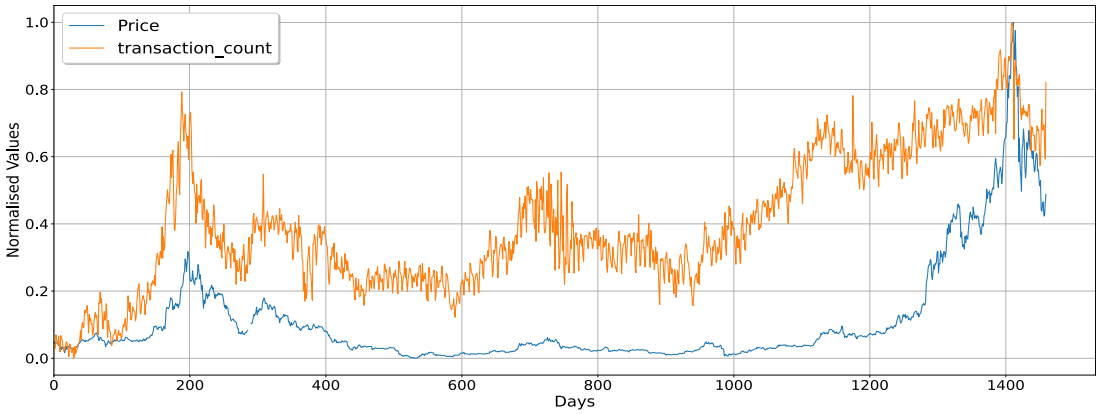

FIGURE 2. Correlation between normalized prices and transaction count.

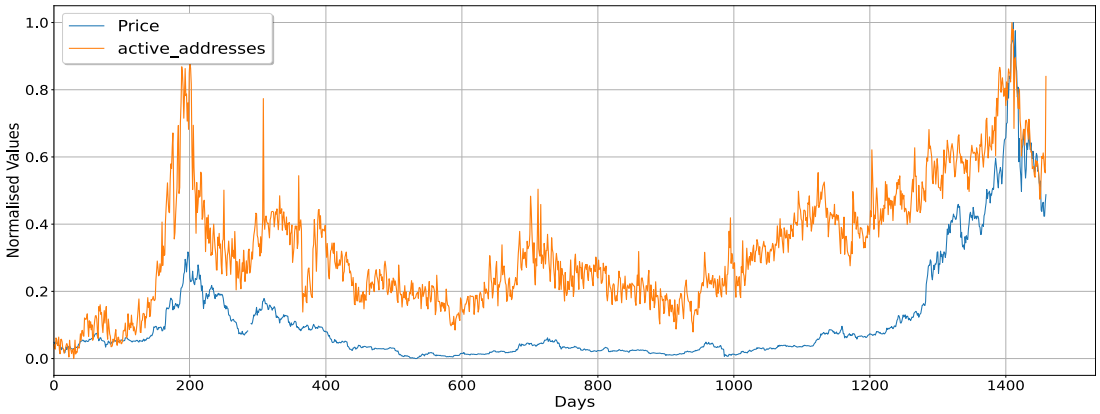

FIGURE 3. Correlation between normalized prices and active addresses.

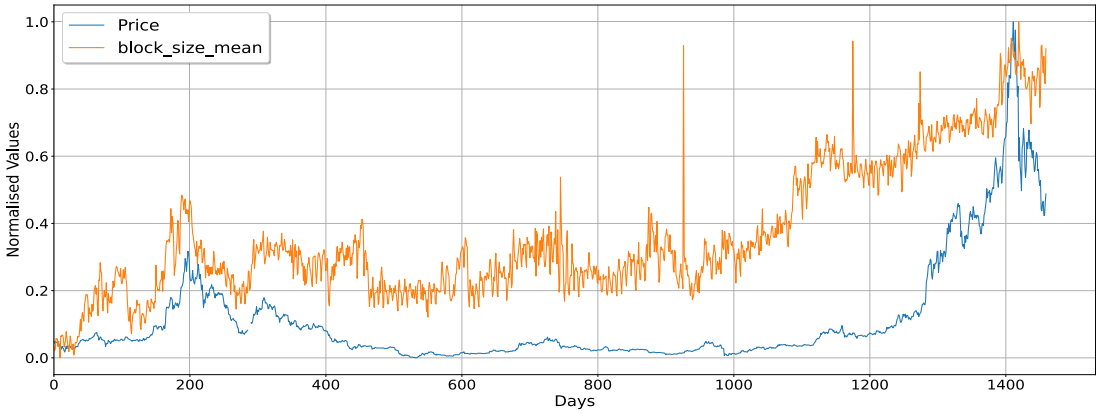

FIGURE 4. Correlation between normalized prices and block size.

For selecting the metrics, we chose two correlation methods - Pearson, which is the commonly used method for financial markets because of the linear nature of the data, and Spearman's, a rank-based correlation method, to introduce new metrics potentially. We also attempted to use Kendall's Tau, but our trials did not introduce any new metrics besides what the Spearman method had already provided.

6) ANALYSIS METRICS AND APPROACH

In this study, each significant Ethereum on-chain metric's linear impact on the price is investigated using Pearson's and Spearman correlation coefficients. Pearson coefficients vary from -1 to 1, with 1 indicating perfect positive correlation, -1 indicating perfect negative correlation, and 0 indicating no correlation at all. Equation (1) represents the Pearson

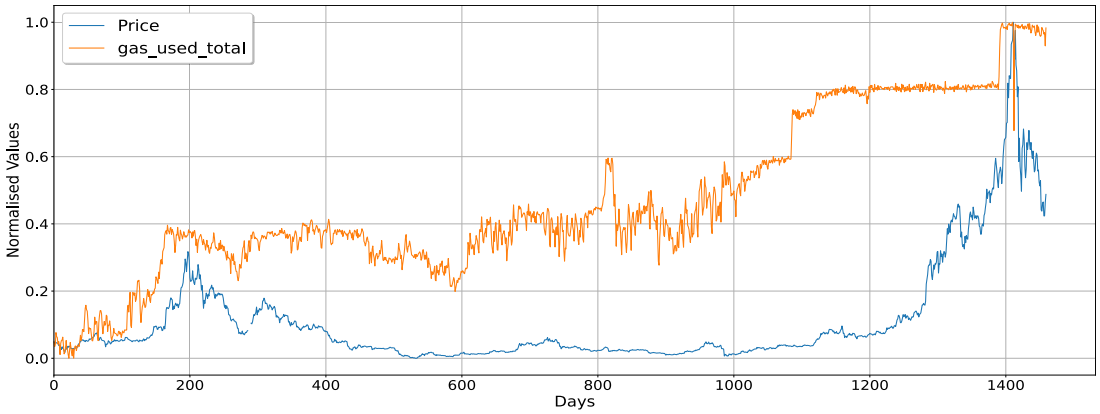

FIGURE 5. Correlation between normalized prices and total gas used.

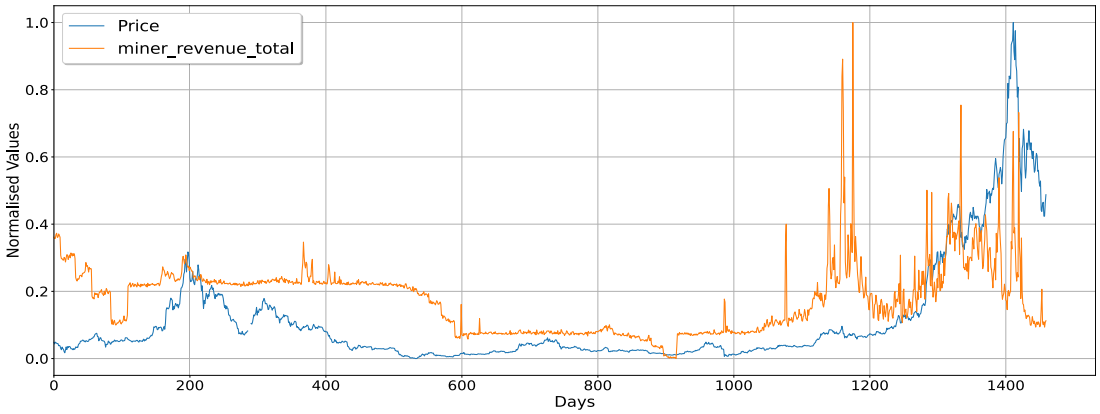

FIGURE 6. Correlation between normalized prices and miner revenue.

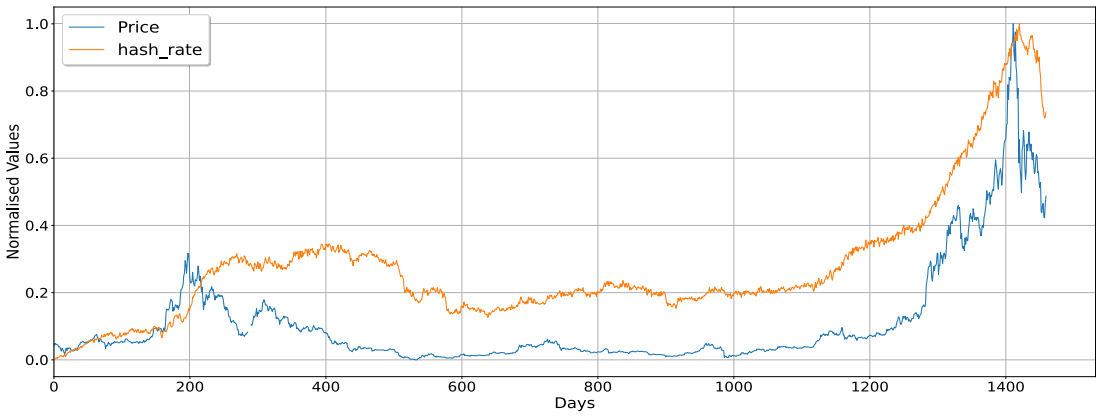

FIGURE 7. Correlation between normalized prices and hash rate.

correlation coefficient.

$$r = \frac{\sum_{i=1}^{n}(x_i - x)(y_i - y)}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (1)$$

The Spearman rank correlation coefficient [55] is a nonparametric measurement correlation used to evaluate the monotonic relationship between two variables.

$$\rho = 1 - \frac{6 \sum {d_i}^2}{n(n^2 - 1)} \quad (2)$$

Pearson correlation coefficients are used to quantify the linear connection between variables. In contrast, Spearman correlation coefficients are only applicable to monotonic

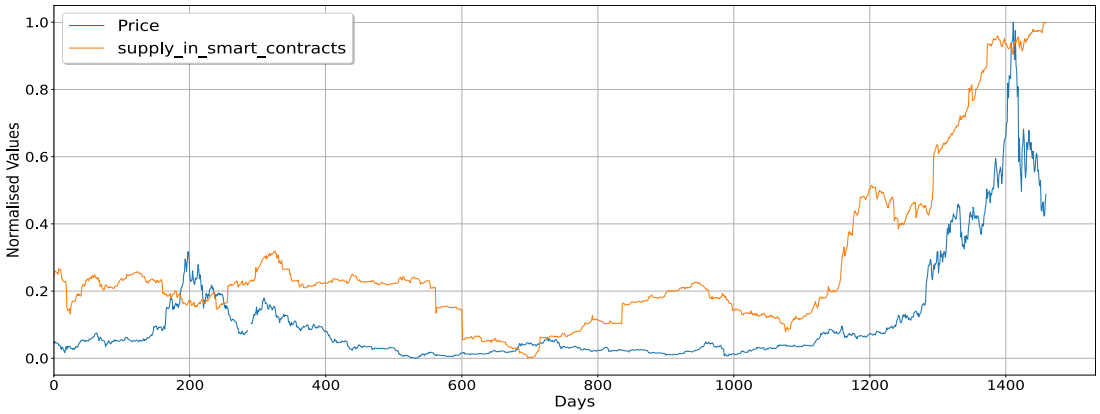

FIGURE 8. Correlation between normalized prices and supply in smart contracts.

TABLE 2. Mean correlation value for every on-chain metric between 2016 and 2021.

Data Category	Pearson Correlation	Spearman Correlation
Block Size	0.7465	0.5494
Block Height	0.4483	0.1640
Hash rate	0.8791	0.5681
Difficulty	0.8865	0.5663
Transactions Volume	0.3892	0.4213
Transactions rate	0.7651	0.7723
Gas price	0.4420	0.6019
Total Gas used	0.6038	0.3875
Supply in Smart Contracts	0.8725	0.6153
Internal Contract Calls	0.6218	0.3549
External Contract Calls	0.5128	0.3549
Miners Revenue	0.5924	0.4440
Miners Inflow	0.1335	0.3191
Miners Outflow	0.0612	0.2136
Miners to exchanges	0.0376	0.1904
Exchange Withdrawls	0.3943	0.5902
Exchange Deposits	0.1778	0.3161
Exchange Inflow	0.1589	0.3427
Exchange Outflow	0.0156	0.1139
Exchange NetFlow	0.4292	0.2105
Daily Active Address	0.7889	0.7277
Total Address	0.5297	0.1640
Total New Address	0.6509	0.6856
Addresses (> 1 coin)	0.5872	0.5365
Addresses (> 10 coins)	0.6215	0.5912
Addresses (> 100 coins)	0.4545	0.4163

connections, in which variables tend to move in the same or opposite direction but not necessarily at the same rate. In a linear relationship, the rate is constant.

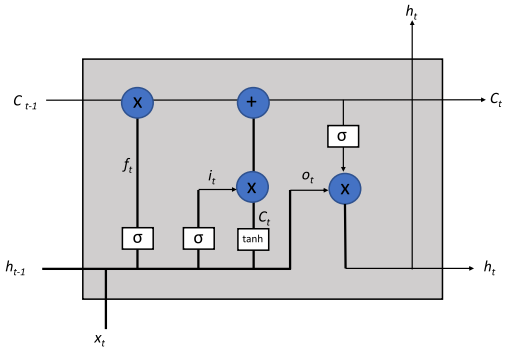

FIGURE 9. LSTM Memory Cell.

Correlated data cannot be displayed linearly to represent the scaling of negative to positive values accurately. We use the normalisation value to re-scale the data between 0 and 1. The normalising value is determined by using (3).

$$z = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{3}$$

This research aims to examine a wide range of on-chain metrics to determine the metrics that can be used in advanced prediction algorithms and those that should be excluded. Although a highly correlated metric seems significant, it cannot be used as a sophisticated measure to determine whether it is favourable. Hence, it is crucial to understand the characteristics and the effects of on-chain metrics on price and perform correlation analysis to determine if a metric requires further investigation.

B. LONG SHORT-TERM MEMORY MODEL

Long short-term memory (LSTM) is a specific type of recurrent neural network (RNN) capable of solving both long-term and short-term dependence issues in a resilient and efficient manner. The memory cell is the backbone of the LSTM network, which takes the role of the traditional neuron's hidden layers. It contains three gates (input, output, and forget); it can add or remove information from the cell's state.

Updating the cell state and computing the LSTM model's output is done by:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}C_{t-1} + b_i) \quad (4)$$
$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}C_{t-1} + b_f) \quad (5)$$
$$c_t = f \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (6)$$
$$O_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}C_{t-1} + b_o) \quad (7)$$
$$m_t = O_t \odot h(c_t) \quad (8)$$
$$y_t = W_{ym}m_t + b_y \quad (9)$$

where x_t represents the input data, y_t denotes the output data at time t. i_t denotes the input, o_t the output, and f_t forget gates at time t. m_t refers to the activation vector for each memory block, and c_t the activation vector for each cell, respectively. σ, g, and h are activation functions for gates, inputs, and outputs, respectively. Weight coefficients are denoted by W.

C. EVOLUTIONARY ALGORITHMS

Evolutionary Algorithms (EAs) have been around since the early 1950s. It is comprised of Genetic Algorithms (GA), Evolutionary programming (EP), Evolution Strategies (EP). For many years, EAs have been widely used to solve optimization problems in diverse fields and the real world. Applications began to arise to the optimization issues that stem from various research and engineering fields in the late 1990s. The Differential Evolution (DE) was developed by Price and Storn [35]–[38], [39] as a highly competitive optimization algorithm. Compared to other EA's, DE is an appealing optimization algorithm as it is easy to code and implement, and it uses minimal control settings (crossover rate (Cr), scaling parameter (F), and population size (NP)). The performance of DE has been extensively examined with regard to these factors [56]–[58].

DIFFERENTIAL EVOLUTION ALGORITHMS (DE)

Differential Evolution is a real-parameter optimization algorithm. Its operation process consists of four basic steps: initialisation, mutation, crossover and selection operations. It also contains three main parameters: population size (NP), scaling factor (F) and crossover rate (Cr). It is complex and time-consuming to determine the optimal combination of NP, F and Cr by trial and error. However, various self-adaptive approaches have evolved over the years to aid in finding suitable parameter values. Three of these self-adaptive techniques will be employed in this study [56], [59]–[61]. An in-depth explanation of the DE algorithm follows in the next subsections.

a: INITIALIZATION PHASE

In this phase, a random initial population of size NP, $p_0 = (\mathbf{x}_{1,0}, \mathbf{x}_{2,0}, \ldots \mathbf{x}_{\mathrm{NP},0})$ dis generated using (10).

$$x_{i,j} = x_{i,j}^{\min} + \mathrm{rand} \times (x_{i,j}^{\max} - x_{i,j}^{\min}) \quad i = 1, 2, \ldots, \mathrm{NP} \text{ and } j = 1, 2, \ldots, D \quad (10)$$

where D refers to the problem dimension (number of decision variables), and $x_{i,j}^{\max}$, $x_{i,j}^{\min}$ the upper and lower bounds of j^{th} decision variable, NP is the population size or the number of solutions evolved by the DE algorithm.

b: MUTATION OPERATOR

Mutation operators are employed after the initialisation phase to create a mutant solution. According to the literature, mutations may be categorised into a variety of methods with varying capacities and features [59]. While some are useful for investigation, others are capable of being exploited. The following techniques are the most often used mutation approaches.

- DE/best/1

$$\mathbf{v}_i^g = \mathbf{x}_{\mathrm{best}}^g + F(\mathbf{x}_{r_1}^g - \mathbf{w}\mathbf{x}_{r_2}^g) \quad (11)$$

- DE/best/2

$$\mathbf{v}_i^g = \mathbf{x}_{\mathrm{best}}^g + F(\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) + F(\mathbf{w}\mathbf{x}_{r_3}^g - \mathbf{x}_{r_4}^g) \quad (12)$$

- DE/rand/1

$$\mathbf{v}_i^g = \mathbf{x}_{r_1}^g + F(\mathbf{x}_{r_2}^g - \mathbf{x}_{r_3}^g) \quad (13)$$

- DE/rand/2

$$\mathbf{v}_i^g = \mathbf{x}_{r_1}^g + F(\mathbf{x}_{r_2}^g - \mathbf{x}_{r_3}^g) + F(\mathbf{x}_{r_4}^g - \mathbf{x}_{r_5}^g) \quad (14)$$

- DE/current-to-$p$best/1

$$\mathbf{v}_i^g = \mathbf{x}_i^g + F(\mathbf{x}_{\mathrm{best}}^g - \mathbf{x}_i^g) + F(\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \quad (15)$$

- DE/current-to-best/1

$$\mathbf{v}_i^g = \mathbf{x}_i^g + F(\mathbf{x}_{\mathrm{best}}^g - \mathbf{x}_i^g) + F(\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \quad (16)$$

where the indexes $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5 \neq i$ and r_1, r_2, r_3, r_4, r_5 are randomly generated integers in [1, NP]. The current individual or solution is represented using i. The current generation g, the best solution $\mathbf{x}_{\mathrm{best}}$ in the current iteration and $\mathbf{x}_{p\mathrm{best}}$ is an individual that is randomly chosen from the top $p\%$ solutions.

c: CROSSOVER OPERATOR

After mutating, every mutant vector $\mathbf{v}_i$ is converted to a trial/offspring vector $\mathbf{u}_i$ using a crossover operator. Exponential and binomial crossover operators [56], [60] are the two primary crossover operators.

The first crossover is the binomial crossover operator, which is performed using (17).

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & \text{if } \mathrm{rand}(0,1) \leq \mathrm{Cr} \text{ or } j = j_{\mathrm{rand}}, \\ x_{i,j}^g, & \text{Otherwise.} \end{cases} \quad (17)$$

where $j_{\mathrm{rand}} \in [1, 2, \ldots, D]$ and rand is a random number belongs to [0, 1]. The values of j_{rand} and rand are chosen randomly to ensure that at least one variable from the trial vector is picked. The crossover rate (Cr) is used to determine how many variables are inherited from the donor vector.

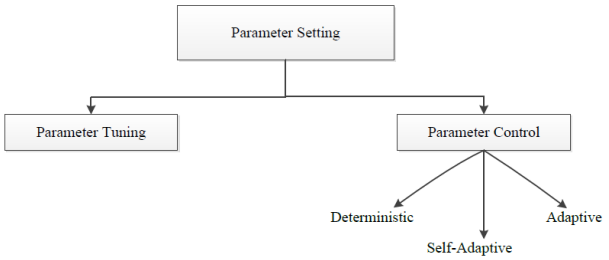

FIGURE 10. Taxonomy of parameter setting.

As per the literature review, the performance of the DE algorithm is mainly dependent on its parameter values [56], [60]. The balance between exploration and exploitation is achieved by using F. Smaller values of F indicate an increase in the convergence rate, while larger values indicate maintaining population diversity. The rate of change in an individual of the population is determined using Cr. Setting the control parameter values in DE can be split into two processes, parameter tuning and parameter control, as is shown in Fig. 10.

In parameter tuning, good parameter values can be determined before an algorithm is executed, and then these values are not changed during the optimization process; that is, all the parameter values are static during the search process. However, the main disadvantage of parameter tuning is its lack of flexibility during this process. While in the parameter control based method, the values of different parameters can be changed during the evolutionary process, and as per Fig. 10, it can be classified as deterministic, adaptive and self-adaptive. As per the literature [60]–[62], DE algorithms with adaptive and self-adaptive techniques have often been more successful than classical ones. Thus, all selected algorithms applied in this study used a self-adaptive technique to set the parameters [56], [61].

d: SELECTION OPERATOR

The parent ($\mathbf{x}_i^g$) and trial ($\mathbf{u}_i^g$) populations are subjected to a greedy selection to determine which solution from each group ($\mathbf{x}_i^g$) and ($\mathbf{u}_i^g$) enters the next generation. As long as the trial solution $f(\mathbf{u}_i^g)$ has a greater fitness function value than the parent solution $f(\mathbf{x}_i^g)$, the trial solution is included in the new population; else, the parent solution will be included. This procedure is performed mathematically in the following way:

$$\mathbf{x}_i^{g+1} = \begin{cases} \mathbf{u}_i^g, & \text{if } f(\mathbf{u}_i^g) \le f(\mathbf{x}_i^g), \\ \mathbf{x}_i^g, & \text{Otherwise.} \end{cases} \tag{18}$$

1) L-SHADE OPTIMIZATION ALGORITHM

Researchers in [59] recently proposed a DE-based evolutionary algorithm known as L-SHADE. It is an improved variant of the SHADE algorithm that was developed by [63]. In L-SHADE, the DE/current-to-$p$best/1 mutation operator described in the following equation has been used to generate new solutions.

$$\mathbf{v}_i^g = \mathbf{x}_i^g + F(\mathbf{x}_{\text{best}}^g - \mathbf{x}_i^g) + F(\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \tag{19}$$

Algorithm 1 L-SHADE Algorithm

$g = 1, N_g = N^{\text{init}}$, Archive $A = \emptyset$;
Randomly, generate an initial population $P_0 = (\mathbf{x}_1^g, \ldots, \mathbf{x}_N^g)$;
initialize M_{Cr}, M_F, and set their values to 0.5;
while The termination condition is not satisfied **do**
  $S_{\text{Cr}} = \{\}, S_F = \{\}$;
  for $i = 1$ to N **do**
    $r_i =$ Select from [1, H] randomly;
    If $M_{\text{Cr},r_i} = \perp$, $\text{Cr}_{i,g} = 0$. Otherwise
    $\text{Cr}_{i,g} = \text{randn}_i(M_{\text{Cr},r_i}, 0.1)$;
    $F_{i,g} = \text{randc}_i(M_{F,r_i}, 0.1)$;
    Generate trial vector $\mathbf{u}_i^g$ according to current-to-$p$best/1/bin;
  end for
  for $i = 1 : N$ **do**
    if $f(\mathbf{u}_i^g) \le f(\mathbf{x}_i^g)$ **then**
      $\mathbf{x}_i^{g+1} \leftarrow \mathbf{u}_i^g$;
    else
      $\mathbf{x}_i^{g+1} \leftarrow \mathbf{x}_i^g$;
    end if
    if $f(\mathbf{u}_i^g) < f(\mathbf{x}_i^g)$ **then**
      $\mathbf{x}_i^g \rightarrow A$;
      $\text{Cr}_{i,g} \rightarrow S_{\text{Cr}}, F_{i,g} \rightarrow S_F$;
    end if
  end for
  If necessary, delete randomly selected individuals from the archive such that the archive size is $|A|$.
  Update memories M_{Cr} and M_F (Algorithm 1);
  Calculate N_{g+1} according to (10);
  if $N_g < N_{g+1}$ **then**
    Sort individuals in P based on their fitness values and delete lowest $N_g - N_{g+1}$ members;
    Resize archive size $|A|$ according to new $|P|$;
  end if
  $g++$;
end while

Instead of using fixed population size, a linear population size mechanism is used, in which the population size is linearly reduced from a high value to a small value [59]. For each generation, the population size is set to one N_{init}, and the population after the run is set to one N_{min}. After each generation, the next generation's population number is calculated as follows:

$$\text{NP}^{g+1} = \text{round}[(\frac{\text{NP}^{\text{min}} - \text{NP}^{\text{init}}}{\text{max}_{\text{fes}}}) \times \text{fes} + \text{NP}^{\text{init}}] \tag{20}$$

where NP^{min} is the minimum number of individuals the algorithm can use, fes the current number of function evaluations (fes), max_{fes} the largest number of fes.

Algorithm 2 jSO Algorithm

Define $g \leftarrow 0, A \leftarrow \{\}$;
Generate an initial random population (P_0) of size NP, that represents the initial structure of the LSTM;
Evaluate $f(P_0)$.
Update fes; fes $\leftarrow$ fes + NP;
while fes $\leq \max_{\text{fes}}$ **do**
  $g \leftarrow g + 1$;
  for $i = 1 : \text{NP}$ **do**
    Use (21) to produce mutant vector ($\mathbf{v}_i^g$);
    Use (17) (crossover operation) to produce new solution ($\mathbf{u}_i^g$), which represents a newly generated LSTM structure;
    Evaluate $f(\mathbf{u}_i^g)$
  end for
  for $i = 1 : \text{NP}$ **do**
    if $f(\mathbf{u}_i^g) \leq f(\mathbf{x}_i^g)$ **then**
      $\mathbf{x}_i^{g+1} \leftarrow \mathbf{u}_i^g$;
    else
      $\mathbf{x}_i^{g+1} \leftarrow \mathbf{x}_i^g$;
    end if
    if $f(\mathbf{u}_i^g) < f(\mathbf{x}_i^g)$ **then**
      $\mathbf{x}_i^g \rightarrow A$;
      Update F and Cr as in [59], and F_w as in (22);
    end if
    If needed, update the archive A;
    Use (20) to linearly update the population size;
  end for
end while

2) jSO OPTIMIZATION ALGORITHM

As mentioned in the above section, the L-SHADE algorithm is an enhanced variant of a DE-based algorithm that employs a technique to reduce the population linearly. Later, an enhancement to the L-SHADE algorithm was proposed, named iL-SHADE, to solve optimization problems [64]. Brest *et al.* [56] introduced a more powerful version of iL-SHADE, named jSO, that employs a novel weighted mutation operator that is a modified version of the DE/current-to-*p*best/1 mutation operator as

$$\mathbf{v}_i^g = \mathbf{x}_i^g + F_w(\mathbf{x}_{p\text{best}}^g - \mathbf{x}_i^g) + F(\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \tag{21}$$

where F_w is a weighted version of the scaling factor F calculated by:

$$F_w = \begin{cases} 0.7 \times F, & n\text{fes} < 0.2 \times \max_{\text{fes}}, \\ 0.8 \times F, & n\text{fes} < 0.4 \times \max_{\text{fes}}, \\ 1.2 \times F, & \text{otherwise.} \end{cases} \tag{22}$$

where $n$fes is the current number of function evaluations and $\max_{\text{fes}}$ is the maximum number of function evaluations. The F values are computed using the same approach as Tanabe *et al.* [59]. These strategies employ a lower factor F_w to multiply the difference of vectors which appears early in the optimization process, while a larger factor F is used later in the process. The main steps of the jSO algorithm are presented in Algorithm 2. The jSO method, like the L-SHADE algorithm, employs a linear population size mechanism to update the total number of individuals in a population every generation [59].

Algorithm 3 MPEDE Algorithm

Set $\mu\text{Cr}_j = 0.5, \mu F_j = 0.5, \triangle f_j = 0$ and $\triangle \text{fes}_j = 0$ for each $j = 1, \ldots, 4$;
Initialize, NP, ng for each $j = 1, \ldots 4$;
Initialize the pop randomly distributed in the solution space;
Initial λ_j and set $\text{NP}_j = \lambda_j \times \text{NP}$;
Randomly partition pop into pop_1, pop_2, pop_3 and pop_4 with respect to their sizes;
Randomly select a subpopulation $\text{pop}_j (j = 1, 2, 3)$ and combine pop_j with pop_4. Let $\text{pop}_j = \text{pop}_j \cup \text{pop}_4$ and $\text{NP}_j = \text{NP}_j + \text{NP}_4$;
Set $g = 0$;
while $g \leq \max_g$ **do**
  $g = g + 1$;
  for $j = 1 \rightarrow 3$ **do**
    Calculate μCr_j and μF_j;
    Calculate $\text{Cr}_{i,j}$ and $F_{i,j}$ for each individual $\mathbf{x}_i$ in pop_j;
    Perform the jth mutation strategy and related crossover operators over subpopulation pop_j;
    Set $S_{\text{Cr,j}} = \emptyset$ and $S_{F,j} = \emptyset$;
  end for
  for $i = 1 \rightarrow \text{NP}$ **do**
    if $f(\mathbf{x}_i^g) \leq f(\mathbf{u}_i^g)$ **then**
      $\mathbf{x}_{i+1}^g = \mathbf{x}_i^g$;
    else
      $\mathbf{x}_{i+1}^g = \mathbf{u}_i^g$; $\triangle f_j = \triangle f_j + f(\mathbf{x}_i^g) - f(\mathbf{u}_i^g)$;
      $\text{Cr}_{i,j} \rightarrow S_{\text{Cr,j}}$; $F_{i,j} \rightarrow S_{F,j}$
    end if
  end for
  $\text{pop} = \cup_{j=1,\ldots,3}\text{pop}_j$
  if $\text{mod}(g, \text{ng}) == 0$ **then**
    $k = \arg(\max_{1 \leq j \leq 3} = (\frac{\triangle f_j}{\text{ng.NP}_j}))$
    $\triangle f_j = 0$;
  end if
  Randomly partition pop into $\text{pop}_1, \text{pop}_2, \text{pop}_3$ and pop_4;
  Let $\text{pop}_k = \text{pop}_k \cup \text{pop}_4$ and $\text{NP}_k = \text{NP}_k + \text{NP}_4$;
end while

3) MPEDE OPTIMIZATION ALGORITHM

The multi-population based ensemble of mutation strategies (MPEDE) is a multi-population based-DE algorithm [65], in which the whole population is dynamically divided into several indicator subpopulations (equal and relatively lesser sized) and one relatively large reward subpopulation for each generation to materialize the effective

ensemble of various mutation methods into one DE variant. A mutation strategy is allocated to each indicator subpopulation. The reward subpopulation is allocated to the mutation approach that performs best in the most recent generations based on the objective function values.

In MPEDE, the following mutation strategies have been utilized.

- DE/current-to-$p$best/1

$$\mathbf{v}_i^g = \mathbf{x}_i^g + F(\mathbf{x}_{p\text{best}}^g - \mathbf{x}_i^g) + F(\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \quad (23)$$

- DE/current-to-rand/1

$$\mathbf{v}_i^g = \mathbf{x}_i^g + F(\mathbf{x}_i^g - \mathbf{x}_i^g) + F(\mathbf{x}_{r_1}^g - \mathbf{x}_{r_2}^g) \quad (24)$$

- DE/pbad-to-$p$best/1

$$\mathbf{v}_i^g = \mathbf{x}_i^g + F(\mathbf{x}_{p\text{best}}^g - \mathbf{x}_{p\text{bad}}^g) \quad (25)$$

where $\mathbf{x}_{p\text{bad}}^g$ is a solution selected randomly from the bad top 100 $p\%$ of solutions in the current population with $p \in (0, 1]$ As a result, the top-performing mutation method can acquire additional computing resources.

D. PROPOSED SYSTEM MODEL

The proposed system model is represented in Fig. 11, where we discuss the two modules employed in this study, namely on-chain analysis and deep learning framework. In the on-chain analysis module, we collect on-chain metrics derived from data provided by the blockchain network, such as the size of the blockchain, the number of blocks linked to it, or the difficulty of mining blocks from the Ethereum public blockchain and API of online resources [46], [47]. We normalise the on-chain metrics to appropriately display data in graph form and analyze the data features of the obtained on-chain metrics. The effects of on-chain metrics on price are analyzed by plotting a graph of each normalised on-chain metric against the price. Correlation analysis enables us to choose a subset of metrics that may significantly affect price forecasting: we analyze the price-metric correlations and select the most strongly linked metrics. Pearson's correlation coefficient determines the linear relationship between two variables, while the Spearman Correlation coefficient determines the monotonic relationship between two variables. The highly correlated datasets are split into train and test datasets.

In the deep learning framework module, self-adaptive algorithm-based deep learning models use differential evolution algorithms to assist us in determining the optimal values for the deep-learning model's hyperparameters since determining the optimal values for hyperparameters is a time-consuming process. We use three distinct self-adaptive methods, namely L-SHADE, jSO, and MPEDE, to ensure effective and efficient training. Each self-adaptive approach returns the optimal values for the hyperparameters determined after the deep learning model's training. We develop a deep-learning model using the optimal hyperparameter values acquired from each self-adaptive method. We utilize test datasets to test self-adaptive algorithms-based deep learning models and conduct MAE, mean squared error (MSE) and MAPE analysis to assess the accuracy of self-adaptive algorithms.

TABLE 3. Lower and upper bounds of Hyperparameters.

Hyperparameters	Min	Max
No. of LSTM layers	2	4
LSTM Layer 1	8	64
Recurrent Dropout in Layer 1	30	40
LSTM Layer 2	32	128
Recurrent Dropout in Layer 2	30	50
LSTM Layer 3	86	256
Recurrent Dropout in Layer 3	30	50
LSTM Layer 4	86	256
Recurrent Dropout in Layer 4	30	50
Dropout between LSTM and Dense layers	30	50
Number of Dense layers	0	5
Number of neurons in each dense Layer		
Dense Layer 1	256	512
Dense Layer 2	128	256
Dense Layer 3	64	128
Dense Layer 4	32	64
Dense Layer 5	16	32

IV. EXPERIMENTAL SETUP

The proposed optimization methods were implemented in Python 3.7.6 and TensorFlow (TF-API). The training was done on Google Colab using the Tensor Processing Unit (TPU). We used three unique self-adaptive techniques to adjust the 17 hyperparameters involved in the RNN architecture. The proposed algorithms produced 150 distinct models, each with its own set of parameters. The optimal parameter combination with the most negligible loss is selected to develop a prediction model for Ethereum. The hyperparameters are listed in the Table 3.

V. RESULTS AND ANALYSIS

This section summarises and discusses the findings of our research. The correlation analysis results were used to develop a deep learning model based on the data characteristics of on-chain data and their influence on the Ethereum price. The deep learning model's hyperparameters were optimised using three self-adaptive techniques: L-SHADE, jSO, and MPEDE. We used optimization algorithms due to their unparalleled speed in generating ideal model parameters to precisely calculate and closely monitor the current Ethereum price. The proposed algorithms time performance is evaluated based on optimization, training, and testing times. It is worth emphasising that optimization time refers to the time it takes for algorithms that use the optimization operator to determine the best parameter combination for each algorithm with the most negligible loss, as shown in Table 4. The optimization time of the proposed algorithms L-SHADE-LSTM, jSO-LSTM and MPEDE-LSTM to achieve the optimal parameters are 23077.59s, 24547.86s and 23872.21s, respectively. As shown in Table 5, our proposed approach

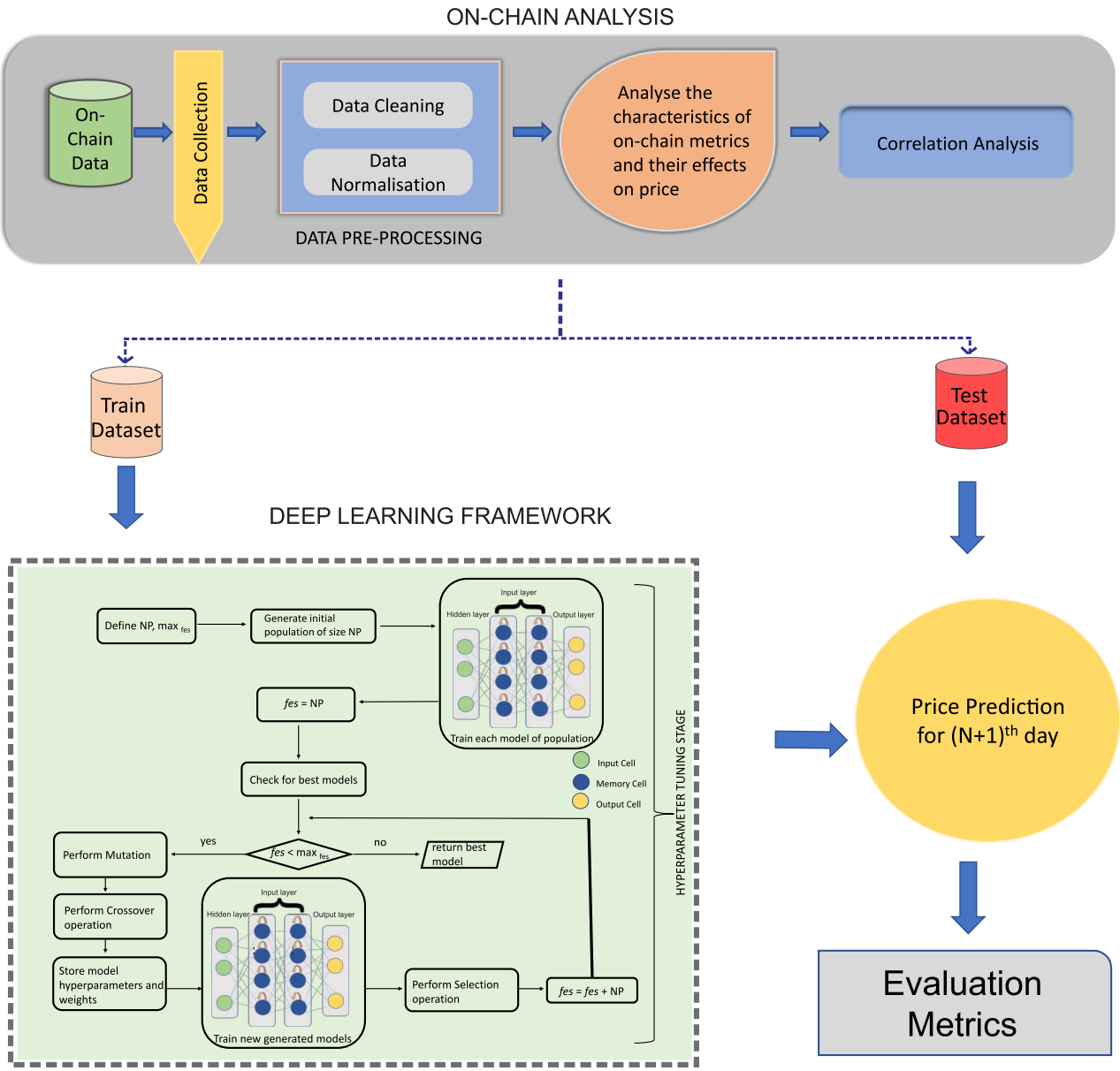

FIGURE 11. Proposed System Model.

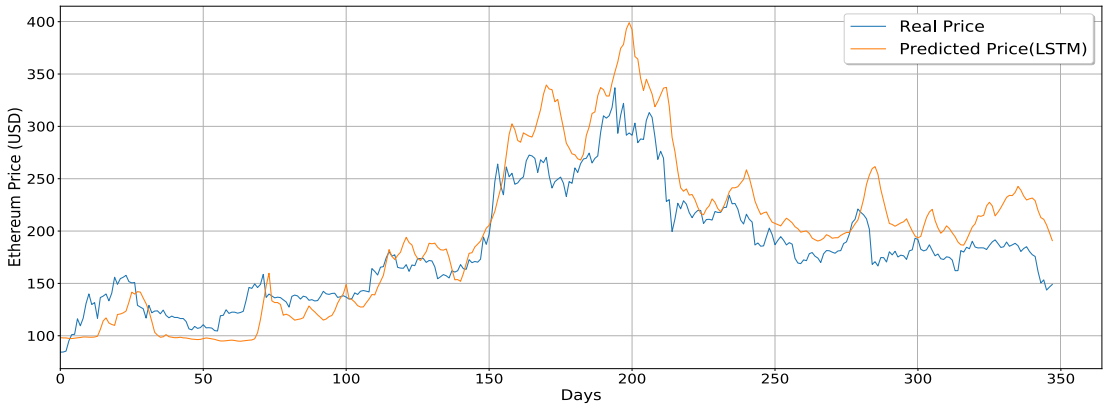

FIGURE 12. Real Price vs Predicted Price using LSTM.

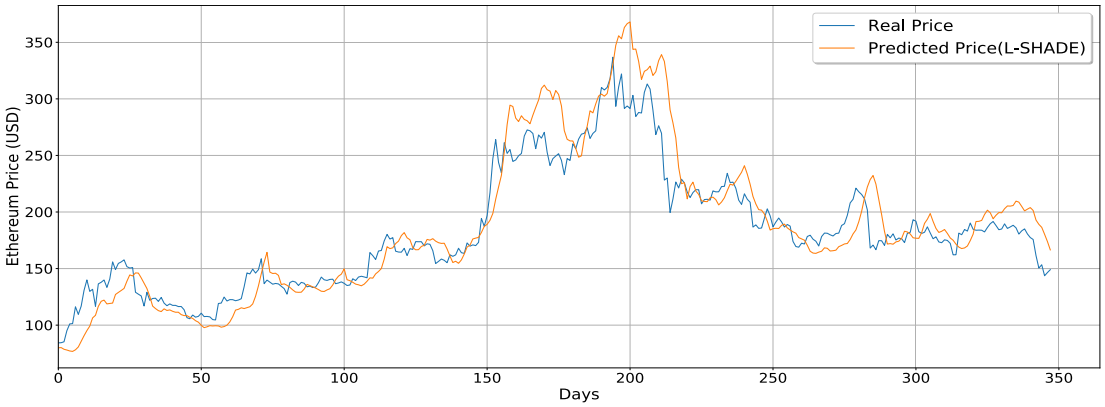

FIGURE 13. Real Price vs Predicted Price using L-SHADE-LSTM optimization algorithm.

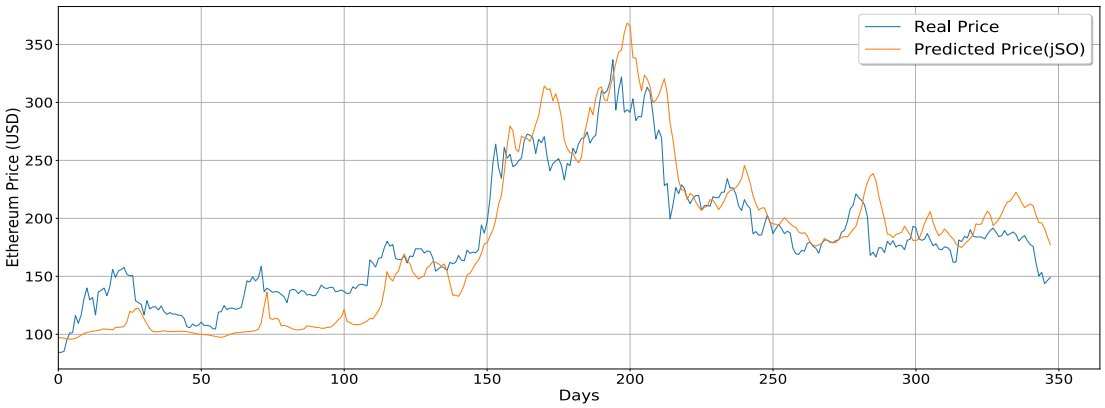

FIGURE 14. Real Price vs Predicted Price using jSO-LSTM optimization algorithm.

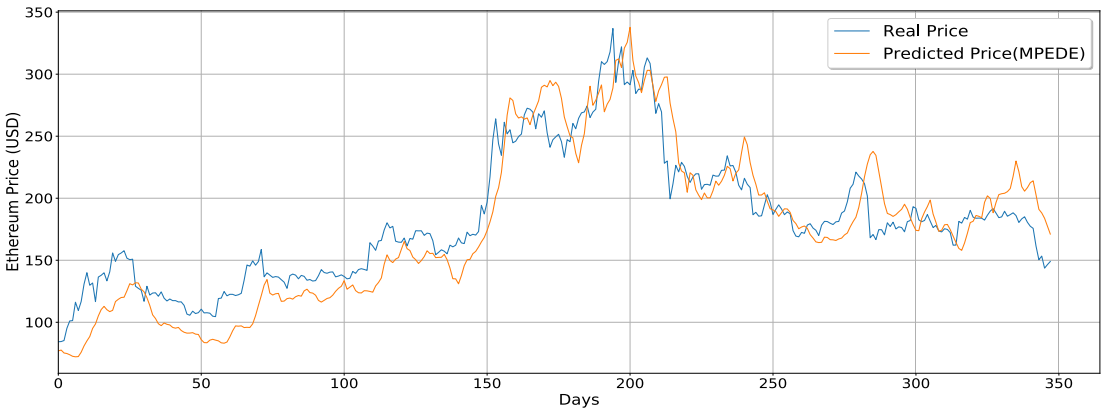

FIGURE 15. Real Price vs Predicted Price using MPEDE-LSTM optimization algorithm.

also takes the least training and testing time compared to a traditional LSTM model.

In Fig. 12 -15, a plot of the predicted price for Ethereum was obtained by using the traditional LSTM model and self-adaptive techniques- L-SHADE, jSO and MPEDE for 60% training size compared to the actual Ethereum price is depicted. Our model adopts a holistic approach by analysing the influence of core components of Ethereum, including miners, nodes, and exchanges activities, that helps us identify price trends better. Comparing Fig. 12-15, it is evident that the predicted price obtained using the L-SHADE, jSO, and MPEDE algorithms is relatively closer to the actual price of Ethereum when compared to the traditional LSTM model.

TABLE 4. The best Hyperparameters values obtained by L-SHADE, jSO & MPEDE algorithms.

Hyperparameters	L-SHADE	jSO	MPEDE
No. of LSTM layers	3	2	3
LSTM Layer 1	42	18	58
Recurrent Dropout in Layer 1	37	37	37
LSTM Layer 2	48	48	72
Recurrent Dropout in Layer 2	34	35	40
LSTM Layer 3	233	109	236
Recurrent Dropout in Layer 3	37	34	30
LSTM Layer 4	143	201	131
Recurrent Dropout in Layer 4	43	31	34
Dropout between LSTM and Dense layers	33	35	43
Number of Dense layers	2	2	3
Dense Layer 1	435	510	437
Dense Layer 2	142	239	214
Dense Layer 3	82	79	120
Dense Layer 4	38	32	32
Dense Layer 5	18	24	30

TABLE 5. Comparison of computational overhead times for training and testing between LSTM and different optimization algorithms.

Model	Training (s)	Testing (s)
LSTM	551.40	4.43
L-SHADE-LSTM	527.86	3.29
jSO-LSTM	538.16	4.15
MPEDE-LSTM	497.22	2.84

TABLE 6. MAE, MSE & MAPE values for LSTM and different optimization algorithms at 60% training size.

	MAE	MSE	MAPE
LSTM	26.41243	936.22145	43.64065
L-SHADE-LSTM	16.87477	559.22696	36.02115
jSO-LSTM	21.40334	735.06323	39.93681
MPEDE-LSTM	19.82180	597.57913	37.84366

We conducted a mean absolute error (MAE), mean square error (MSE) and mean absolute percentage error (MAPE) comparison between the different self-adaptive techniques and the traditional LSTM model to validate this finding. The MAE, MSE and MAPE values are calculated using the following formula:

$$\text{MAE} = \frac{1}{n}\sum_{k=1}^{n}|y_k - \hat{y_k}| \tag{26}$$

$$\text{MSE} = \frac{1}{n}\sum_{k=1}^{n}(y_k - \hat{y_k})^2 \tag{27}$$

$$\text{MAPE} = \frac{1}{n}\sum_{k=1}^{n}\frac{|y_k - \hat{y_k}|}{|y_k|} \times 100 \tag{28}$$

where n is the number of data points and y_k and $\hat{y_k}$ are the real and predicted prices of sth point.

The results of the MAE, MSE and MAPE analysis are presented in Table 6 and Table 7. It is observed that for training sizes of 60% and 90%, the self-adaptive

TABLE 7. MAE, MSE & MAPE values for LSTM and different optimization algorithms at 90% training size.

	MAE	MSE	MAPE
LSTM	19.87341	591.55436	29.06433
L-SHADE-LSTM	13.62016	286.46870	23.84312
jSO-LSTM	15.31869	381.55121	25.02581
MPEDE-LSTM	14.89678	383.43369	24.52417

algorithm-based LSTM models, namely L-SHADE-LSTM, jSO-LSTM, and MPEDE-LSTM, yield lower MAE, MSE and MAPE values. Correspondingly, our proposed model achieved the following improvements, 33.79%, 20.93% and 24.99% improvement for MAE, 45.91%, 28.49% and 35.67% improvement for MSE and 17.70%, 12.02% and 14.45% for MAPE over its LSTM model counterpart. The L-SHADE-LSTM model performed the best, yielding the lowest error rate compared to jSO and MPEDE algorithms. These results suggest that self-adaptive algorithm-based LSTM models provide a quicker and more accurate prediction of Ethereum prices.

VI. CONCLUSION AND FUTURE WORKS

In this article, we analyze factors that impact the price of Ethereum using a profusion of on-chain metrics relating to the user, miner, and exchange activity. Using these findings, we develop an LSTM model that employs three distinct self-adaptive methods to identify the best hyperparameter values for predicting the price of Ethereum. We compare each self-adaptive technique to one another as well as to a traditional LSTM model. In comparison, self-adaptive algorithm based LSTM models provide a quick and more accurate price prediction for Ethereum.

As demonstrated in this article, on-chain metrics can be utilized as a supplementary tool in combination with existing techniques to predict prices more accurately. They provide important insights into the blockchain network's utilisation and health. In future work, we aim to extend this research by including the attention mechanism into LSTM framework [66] and applying it for on-chain data for predicting cryptocurrency prices. In addition, it will be interesting to explore on-chain and off-chain variables such as Reddit, Twitter, and Google Trends, among others, in conjunction with a self-adaptive algorithm-based LSTM model to attain near-perfect price prediction accuracy.

REFERENCES

[1] Department of Industry, and Science, and Energy and Resources. *National Blockchain Roadmap*. Department of Industry, Science, Energy and Resources. Accessed: Jun. 29, 2021. [Online]. Available: https://www.industry.gov.au/data-and-publications/national-blockchain-roadmap
[2] X. Xiang, M. Wang, and W. Fan, "A permissioned blockchain-based identity management and user authentication scheme for E-Health systems," *IEEE Access*, vol. 8, pp. 171771–171783, 2020.
[3] S. Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*. Accessed: Jan. 3, 2021. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[4] W. Dai. *B-Money*. Satoshi Nakamoto Institute. Accessed: Dec. 29, 2020. [Online]. Available: https://nakamotoinstitute.org/b-money

[5] N. Szabo. *Bit Gold*. Satoshi Nakamoto Institute. Accessed: Nov. 15, 2020. [Online]. Available: https://nakamotoinstitute.org/bit-gold

[6] H. Finney. *RPOW–Reusable Proofs of Work*. Satoshi Nakamoto Institute. Accessed: Nov. 12, 2020. [Online]. Available: https://nakamotoinstitute.org/finney/rpow

[7] V. Buterin. *Ethereum Whitepaper*. Accessed: Mar. 6, 2021. [Online]. Available: https://ethereum.org/en/whitepaper

[8] CoinMarketCap. *Cryptocurrency Prices, Charts and Market Capitalizations*. CoinMarketCap. Accessed: Apr. 18, 2021. [Online]. Available: https://coinmarketcap.com/

[9] A. Eross, F. McGroarty, A. Urquhart, and S. Wolfe, "The intraday dynamics of bitcoin," *Res. Int. Bus. Finance*, vol. 49, pp. 71–81, Oct. 2019.

[10] R. Hudson and A. Urquhart, "Technical trading and cryptocurrencies," *Ann. Oper. Res.*, vol. 297, nos. 1–2, pp. 191–220, Feb. 2021.

[11] J.-Z. Huang, W. Huang, and J. Ni, "Predicting bitcoin returns using high-dimensional technical indicators," *J. Finance Data Sci.*, vol. 5, no. 3, pp. 140–155, Sep. 2019.

[12] L. Kristoufek, "BitCoin meets Google trends and wikipedia: Quantifying the relationship between phenomena of the internet era," *Sci. Rep.*, vol. 3, no. 1, pp. 1–7, Dec. 2013.

[13] 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.

[14] 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.

[15] P. Jay, V. Kalariya, P. Parmar, S. Tanwar, N. Kumar, and M. Alazab, "Stochastic neural networks for cryptocurrency price prediction," *IEEE Access*, vol. 8, pp. 82804–82818, 2020.

[16] A. M. Khedr, I. Arif, V. P. Raj, M. El-Bannany, S. M. Alhashmi, and M. Sreedharan, "Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey," *Intell. Syst. Accounting, Finance Manage.*, vol. 28, no. 1, pp. 3–34, 2021.

[17] S. McNally, J. Roche, and S. Caton, "Predicting the price of bitcoin using machine learning," in *Proc. 26th Euromicro Int. Conf. Parallel, Distrib. Netw.-Based Process. (PDP)*, Mar. 2018, pp. 339–343.

[18] T. Klein, H. P. Thu, 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.

[19] 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.

[20] M. I. Mehar, C. L. Shier, A. Giambattista, E. Gong, G. Fletcher, R. Sanayhie, H. M. Kim, and M. Laskowski, "Understanding a revolutionary and flawed grand experiment in blockchain: The dao attack," *J. Cases Inf. Technol.*, vol. 21, no. 1, pp. 19–32, 2019.

[21] P. Tsankov, A. Dan, D. Drachsler-Cohen, A. Gervais, F. Buenzli, and M. Vechev, "Securify: Practical security analysis of smart contracts," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 67–82.

[22] W. Chen, Z. Zheng, E. C.-H. Ngai, P. Zheng, and Y. Zhou, "Exploiting blockchain data to detect smart Ponzi schemes on ethereum," *IEEE Access*, vol. 7, pp. 37575–37586, 2019.

[23] M. Bartoletti, S. Carta, T. Cimoli, and R. Saia, "Dissecting Ponzi schemes on ethereum: Identification, analysis, and impact," *Future Gener. Comput. Syst.*, vol. 102, pp. 259–277, Jan. 2020.

[24] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[25] M. S. Bhargavi, S. M. Katti, M. Shilpa, V. P. Kulkarni, and S. Prasad, "Transactional data analytics for inferring behavioural traits in ethereum blockchain network," in *Proc. IEEE 16th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2020, pp. 485–490.

[26] Blockchair. *Ethereum Blockchain Size Chart*. Blockchair. Accessed: Apr. 21, 2021. [Online]. Available: https://blockchair.com/ethereum/charts/blockchain-size

[27] M. Bartoletti, S. Lande, L. Pompianu, and A. Bracciali, "A general framework for blockchain analytics," in *Proc. 1st Workshop Scalable Resilient Infrastruct. Distrib. Ledgers*, Dec. 2017, pp. 1–6.

[28] H. Kalodner, M. Möser, K. Lee, S. Goldfeder, M. Plattner, A. Chator, and A. Narayanan, "Blocksci: Design and applications of a blockchain analysis platform," in *29th USENIX Secur. Symp. (USENIX Secur.)*, 2020, pp. 2721–2738.

[29] Y. Li, K. Zheng, Y. Yan, Q. Liu, and X. Zhou, "EtherQL: A query layer for blockchain system," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2017, pp. 556–567.

[30] P. Zheng, Z. Zheng, J. Wu, and H.-N. Dai, "XBlock-ETH: Extracting and exploring blockchain data from ethereum," *IEEE Open J. Comput. Soc.*, vol. 1, pp. 95–106, 2020.

[31] N. Smuts, "What drives cryptocurrency prices? An investigation of Google trends and telegram sentiment," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 131–134, 2019.

[32] L. Muñoz-González, B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 27–38.

[33] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 19–35.

[34] K. Metaxiotis and K. Liagkouras, "Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review," *Expert Syst. Appl.*, vol. 39, no. 14, pp. 11685–11698, 2012.

[35] K. V. Price, "Differential evolution vs. the functions of the 2nd ICEO," in *Proc. IEEE Int. Conf. Evol. Comput. (ICEC)*, Apr. 1997, pp. 153–157.

[36] K. V. Price, "Differential evolution: A fast and simple numerical optimizer," in *Proc. North Amer. Fuzzy Inf. Process.*, 1996, pp. 524–527.

[37] R. Storn and K. Price, "Differential evolution–A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[38] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 842–844.

[39] R. Storn, "On the usage of differential evolution for function optimization," in *Proc. North Amer. Fuzzy Inf. Process.*, 1996, pp. 519–523.

[40] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 61–106, 2010.

[41] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[42] N. Jagannath, T. Barbulescu, K. M. Sallam, I. Elgendi, A. A. Okon, B. Mcgrath, A. Jamalipour, and K. Munasinghe, "A self-adaptive deep learning-based algorithm for predictive analysis of bitcoin price," *IEEE Access*, vol. 9, pp. 34054–34066, 2021.

[43] B. Wolf. *U.S. Law Enforcers Partner With Cryptocurrency Tracking Firm to Fight Financial Crime*. Thomson Reuters. Accessed: Jun. 24, 2021. [Online]. Available: https://www.thomsonreuters.com/en-us/posts/investigation-fraud-and-risk /cryptocurrency-financial-crime

[44] A. Nathan, G. L. Galbraith, and J. Grimberg. *Crypto: A New Asset Class?*. Goldman Sachs. Accessed: Jun. 21, 2020. [Online]. Available: https://www.goldmansachs.com/insights/pages/crypto-a-new-asset-class.html

[45] B. G. Malkiel, "Efficient market hypothesis," in *The New Palgrave: Finance*. London, U.K.: Palgrave Macmillan, 1989, pp. 127–134.

[46] G. Studio. *Glassnode Studio-on-Chain Market Intelligence*. Glassnode Studio. Aug. 15, 2021. [Online]. Available: https://studio.glassnode.com

[47] The Block. *Ethereum Archives*. Accessed: Aug. 6, 2021. [Online]. Available: https://www.theblockcrypto.com/data/on-chain-metrics/ethereum

[48] Bitcoin. *Bitcoin.Com Markets: Price, Charts, News*. Accessed: Aug. 18, 2021. [Online]. Available: https://markets.bitcoin.com/crypto/BTC

[49] E. Krause. *A Fifth of all Bitcoin is Missing. These Crypto Hunters Can Help*. The Wall Street J. Accessed: Jan. 12, 2021. [Online]. Available: https://www.wsj.com/articles/a-fifth-of-all-bitcoin-is-missing-these-cr ypto-hunters-can-help-1530798731

[50] N. Carter and A. L. Calvez. *Introducing Realized Capitalization*. Coin Metrics. Accessed: Oct. 5, 2020. [Online]. Available: https://coinmetrics.io/realized-capitalization

[51] A. Schoedon. *EIP-1234: Constantinople Difficulty Bomb Delay and Block Reward Adjustment*. Ethereum Improvement Proposals. Accessed: Dec. 3, 2020. [Online]. Available: https://eips.ethereum.org/EIPS/eip-1234

[52] K. Heeg and R. Schultze-Kraft. *Introducing Account-Based on-Chain Metrics for Bitcoin and Ethereum*. Glassnode Insights. Accessed: Mar. 25, 2021. [Online]. Available: https://insights.glassnode.com/account-based-metrics

[53] P. Xia, H. Wang, B. Zhang, R. Ji, B. Gao, L. Wu, X. Luo, and G. Xu, "Characterizing cryptocurrency exchange scams," *Comput. Secur.*, vol. 98, Nov. 2020, Art. no. 101993, doi: 10.1016/j.cose.2020.101993.
[54] Glassnode. *Bitcoin's on-Chain Market Cycles*. Glassnode Insights. Accessed: Apr. 12, 2021. [Online]. Available: https://insights.glassnode.com/bitcoin-onchain-market-cycles
[55] C. Spearman, "The proof and measurement of association between two things," *Amer. J. Psychol.*, vol. 100, nos. 3–4, pp. 441–471, 1987.
[56] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 646–657, Jul. 2006.
[57] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398–417, Apr. 2009.
[58] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
[59] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 1658–1665.
[60] K. M. Sallam, S. M. Elsayed, R. K. Chakrabortty, and M. J. Ryan, "Improved multi-operator differential evolution algorithm for solving unconstrained problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8, doi: 10.1109/CEC48606.2020.9185577.
[61] K. M. Sallam, S. M. Elsayed, R. K. Chakrabortty, and M. J. Ryan, "Evolutionary framework with reinforcement learning-based mutation adaptation," *IEEE Access*, vol. 8, pp. 194045–194071, 2020.
[62] K. M. Sallam, S. M. Elsayed, R. A. Sarker, and D. L. Essam, "Landscape-assisted multi-operator differential evolution for solving constrained optimization problems," *Expert Syst. Appl.*, vol. 162, Dec. 2020, Art. no. 113033, doi: 10.1016/j.eswa.2019.113033.
[63] R. Tanabe and A. Fukunaga, "Success-history based parameter adaptation for differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 71–78.
[64] J. Brest, M. S. Maucec, and B. Boskovic, "IL-SHADE: Improved L-SHADE algorithm for single objective real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 1188–1195.
[65] G. Wu, R. Mallipeddi, P. N. Suganthanc, R. Wang, and H. Chen, "Differential evolution with multi-population based ensemble of mutation strategies," *Inf. Sci.*, vol. 329, pp. 329–345, Feb. 2016.
[66] W. Zheng, P. Zhao, K. Huang, and G. Chen, "Understanding the property of long term memory for the LSTM with attention mechanism," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2021, pp. 2708–2717.

NISHANT JAGANNATH (Member, IEEE) received the master's degree in network engineering. He is currently pursuing the Ph.D. degree with the University of Canberra, Australia. He is a part-time Faculty Member with the Science and Technology Department, University of Canberra. His Ph.D. research investigates the current issues in adopting blockchain technology, given the diverse nature of its implementations from financial markets to supply chain. His research interests include integrating blockchain technology with the Internet of Things (IoT) and cyber-physical systems.

TUDOR BARBULESCU received the bachelor's degree in information technology from the Australian National University. He is currently pursuing the master's degree in computing/business informatics with the University of Canberra. He is currently working as a software developer on personal projects and for clients. His research interest includes the intersection of algorithmic trading and blockchain. Before this he co founded an augmented reality startup.

KARAM M. SALLAM (Member, IEEE) received the Ph.D. degree in computer science from the University of New South Wales at Canberra, Australian Defence Force Academy, Canberra, Australia, in 2018. He is currently a Lecturer at the University of Canberra, Canberra. His current research interests include evolutionary algorithms and optimization, constrained-handling techniques for evolutionary algorithms, operation research, machine learning, deep learning, cybersecurity, and the IoT. He was the winner of the IEEE-CEC2020 and IEEE-CEC2021 competitions. He serves as an organizing committee member for different conferences in the evolutionary computation field and a reviewer for several international journals.

IBRAHIM ELGENDI (Member, IEEE) received the Ph.D. degree in information technology from the University of Canberra, Australia. He is currently a Lecturer in networking and cybersecurity at the University of Canberra. He has 17 years' experience from industry in the field of automation and AI. He has over 14 refereed publications with over 63 citations (H-index 5) in highly prestigious journals and conference proceedings. His research interests include mobile and wireless networks, the Internet of Things, machine learning, and cyber-physical-security. He is a member of the Institution of Engineers Australia. He has served as a Reviewer for a number of journals, such as *IEEE Wireless Communications* magazine, IEEE TRANSACTIONS ON MOBILE COMPUTING, and IEEE/ACM TRANSACTIONS ON NETWORKING.

BRADEN MCGRATH received the master's degree in aeronautics and astronautics from MIT and the Ph.D. degree in aeronautical engineering from The University of Sydney. He is currently a Research Professor at Embry–Riddle Aeronautical University and the CEO of Cann Pharmaceutical Australia. He is also a Chartered Engineer at the Engineers Australia Colleges of Biomedical Engineering and Leadership and Management, a fellow of Engineers Australia, a QinetiQ Fellow, a member of the Society of Flight Test Engineers, and a Certified Sports Medicine Trainer. His current research interest includes the development of secure supply chain technology for pharmaceutical-quality medicinal cannabinoid drug development on the human neurological systems.

ABBAS JAMALIPOUR (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Nagoya University, Japan. He is currently a Professor of ubiquitous mobile networking at The University of Sydney, Australia. He has authored nine technical books, 11 book chapters, over 450 technical papers, and five patents, all in the area of wireless communications. He is a fellow of the Institute of Electronics, Information, and Communication Engineers (IEICE) and the Institution of Engineers Australia, an ACM Professional Member, and an IEEE Distinguished Lecturer. He is also the President and an Elected Member of the Board of Governors of the IEEE Vehicular Technology Society. He was a recipient of a number of prestigious awards, such as the 2019 IEEE ComSoc Distinguished Technical Achievement Award in Green Communications, the 2016 IEEE ComSoc Distinguished Technical Achievement Award in Communications Switching and Routing, the 2010 IEEE ComSoc Harold Sobol Award, the 2006 IEEE ComSoc Best Tutorial Paper Award, as well as 15 Best Paper Awards. He was the Editor-in-Chief of the IEEE Wireless Communications, the Vice President-Conferences, and a member of the Board of Governors of the IEEE Communications Society. He serves as an Editor for IEEE Access, IEEE Transactions on Vehicular Technology, and several other journals. He has been the General Chair or the Technical Program Chair of a number of conferences, including IEEE ICC, GLOBECOM, WCNC, and PIMRC.

MOHAMED ABDEL-BASSET (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in operations research from the Faculty of Computers and Informatics, Zagazig University, Egypt. He is currently an Associate Professor with the Faculty of Computers and Informatics, Zagazig University. He is working on the application of multi-objective and robust meta-heuristic optimization techniques. He has published more than 200 articles in international journals and conference proceedings. His current research interests include optimization, operations research, data mining, computational intelligence, applied statistics, decision support systems, robust optimization, engineering optimization, multi-objective optimization, swarm intelligence, evolutionary algorithms, and artificial neural networks. He is an editor/a reviewer in different international journals and conferences.

KUMUDU MUNASINGHE (Senior Member, IEEE) received the Ph.D. degree in telecommunications engineering from The University of Sydney. He is currently an Associate Professor in network engineering, the Leader of the IoT Research Group, Human Centred Research Centre, University of Canberra. He has over 100 refereed publications with over 900 citations (H-index 17) in highly prestigious journals, conference proceedings, and two books to his credit. He has secured over $ 1.6 million dollars in competitive research funding by winning grants from the Australian Research Council (ARC), the Commonwealth and State Governments, the Department of Defence, and the industry. He has also won the highly prestigious ARC Australian Postdoctoral Fellowship. His research interests include next generation mobile and wireless networks, the Internet of Things, green communication, smart grid communications, and cyber-physical-security. His research has been highly commended through many research awards, including two VC's Research Awards and three IEEE Best Paper Awards. He is currently a member of a Charted Professional Engineer and an Engineering Executive and a Companion (Fellow Status) of Engineers Australia. He served as a co-chair for many international conferences and an editorial board member for a number of journals.