A Hybrid Model for Bitcoin Prices Prediction using Hidden Markov Models and Optimized LSTM Networks

Iman Abu Hashish[1], Fabio Forni[3], Gianluca Andreotti[4] and Tullio Facchinetti[5]
Dept. of Electrical, Computer and Biomedical Engineering
University of Pavia
Pavia, Italy

Shiva Darjani[2]
Dept. of Electrical Engineering
University of Zanjan
Zanjan, Iran

***Abstract*—With the recent advances in the Blockchain technology, and due to its decentralized nature, it has been a much considered approach for solving issues in the Internet of Things (IoT) sector, in particular, for IoT payment platforms. As Machine-to-Machine (M2M) payments are fundamental in the IoT economy, the development of Blockchain-based payment platforms, using cryptocurrency, is continuously increasing as it enables a pure M2M, secure and private financial transactions. Unlike traditional assets, cryptocurrencies have a higher index of volatility, which makes it essential to understand the movement of their prices, as a first step to optimize Blockchain-based M2M payment transactions. In this paper, we propose a novel hybrid model that deals with this challenge from a descriptive, as well as predictive points of view. We use Hidden Markov Models to describe cryptocurrencies historical movements to predict future movements with Long Short Term Memory networks. To evaluate the proposed hybrid model, we have chosen 2-minute frequency Bitcoin data from Coinbase exchange market. Our proposed model proved its effectiveness compared to traditional time-series forecasting models, ARIMA, as well as a conventional LSTM.**

***Index Terms*—Crypto-currency, Bitcoin, Prediction, Hidden Markov Models, Genetic Algorithms, Long Short Term Memory, Neural Networks**

I. Introduction

Bitcoin was first introduced by Satoshi Nakamoto along with the Blockchain, a public decentralized ledger where Bitcoin transactions are recorded [1]. The core idea is removing trusted third parties for financial transactions, which allows the Blockchain to be distributed across a large network, resulting in a true peer-to-peer (P2P) transactions. Introducing such concepts has enabled addressing many issues in different domains and has attracted much attention recently.

The Internet of Things (IoT) is one of those emerging domains that has grown significantly. Given the different contexts where IoT can be involved, it is difficult to set one specific definition to the term [2]. However, from a technology perspective, it can be defined as smart machines interacting with each other, as well as other objects, infrastructures and environments resulting in volumes of generated data which is then processed to useful insights. In this complex and heterogeneous scenario, many challenges faced in the IoT domain were addressed using the Blockchain technology, such as enhancing security issues [3], [4], data storage management, trade of good and even rating systems [5]. Particularly, the Blockchain technology is considered a much needed solution for IoT payment platforms as, due to its decentralized nature, it allows a true machine-to-machine (M2M) financial transactions, since traditional e-business models rely mainly as a trusted third party to act as an intermediary during financial transactions [6]. As a result, the development of Blockchain-based payment platforms, using cryptocurrencies, is being recognized and getting attention.

As M2M financial transactions are considered essential to IoT economy [7], and given that unlike traditional assets, cryptocurrencies have a higher index of volatility [8], it is important to understand how to deal with implementing Blockchain-based M2M payment platforms, by firstly addressing and analyzing the dynamics of cryptocurrencies to understand what may or may not affect the movement of their prices. Accordingly, such studies are essential for optimizing the costs of transactions for services that utilize cryptocurrencies for M2M payment.

The objective of this work is to study cryptocurrencies prices dynamics, specifically Bitcoin, by proposing a new thorough model that addresses this issue from a descriptive point of view through Hidden Markov Models, as well as a predictive one using a Long Short Term Memory network.

This paper is organized as follows. In section II we review the most recent related studies on Blockchain-based M2M payment platforms and Bitcoin prices prediction approaches. Section III explains our proposed model in details, followed by section IV in which we introduce the collected data. Section V showcases the set of features extracted from the collected data while section VI focuses on the implementation of the proposed method. In section VII experimental results are illustrated and, finally, section VIII concludes our work and presents future investigation directions.

II. Related Work

In this section, relative studies have been highlighted in Blockchain-based M2M payment platforms and Bitcoin prices prediction.

978-1-7281-0303-7/19/$31.00 ©2019 IEEE

A. Machine-to-Machine Payment Platforms

J. Poon et al. [9] introduced the so-called Lightning Network as a decentralized system for instant, high-volume micropayments, without the need for a trusted third party. They highlighted that Lightning Network is scalable to a magnitude of high transactions to meet automated payments and so it can suitably supports M2M transactions. On the other hand, Z. Hao et al. [10] argue that Lightning Network suffers from hidden transactions, i.e., the payments are batched and the Blockchain only records combined payments which results in losing lots of information on the raw payments. Accordingly, they introduced FastPay to solve such problems, which is also found in [11]. Moreover, [12] and [13] are two platforms that have the ability to handle large amounts of transactions from IoT devices, evolving networks and data-intensive sensors.

B. Bitcoin Prices Prediction

The number of studies that address Bitcoin prices prediction is relatively small. Most of such studies are focused on predicting daily closing prices. C. Wu et al. [14] introduced a hybrid model of LSTM, paired with AutoRegressive(2) to predict Bitcoin daily price, using AutoCorrelation Function (ACF) and Partial AutoCorrelation function (PACF). The proposed method achieves a Root Mean Squared Error (RMSE) of 247.33, compared to a conventional LSTM, which achieved an RMSE of 256.41. Karakoyum et al. [15] compared AutoRegressive Integrated Moving Average (ARIMA) and LSTM for daily Bitcoin price prediction, and reported a Mean Absolute Percentage Error (MAPE) of 11.86% and 1.4% for ARIMA and LSTM, respectively.

LSTM and Generalized Regression Neural Networks (GRNN) are compared in [16] in terms of performance, and an RMSE of 2750 has been reported for LSTM, which is significantly better than the performance of GRNN. T. Guo et al. [17] proposed a temporal mixture model to predict the volatility of Bitcoin prices. The proposed model is based on Gaussian and log-normal temporal mixture, where an RMSE of 0.025 was reported for Gaussian Temporal Mixture model. In [18], Support Vector Machine (SVM), Linear Regression (LR), Neural Networks (NN), LSTM and rolling LSTM have been implemented and 17 features have been extracted. An RMSE and MAPE of 59.04 and 0.044 has been reported for rolling LSTM model, which outperforms the other considered models.

On the other hand, fewer studies have focused on predicting intra-day Bitcoin prices. Two Regression models, LSTM and Gated Recurrent Unit (GRU) are implemented in [19] to predict 1-minute interval Bitcoin price. Among the models, the GRU model showed the best result with an MSE of 0.00002. S. McNally et al. [20] has implemented Bayesian optimized RNN and LSTM models to predict the direction of Bitcoin price. They showed that the LSTM model outperforms other implemented models with an accuracy of 52%.

In [21], the Bayesian regression model was implemented to predict the average price movement of Bitcoin with a data frequency of 10 seconds. They used their proposed approach to simulate a trading strategy that nearly doubles the investment in less than 60 day period. Classification models such as Baseline, Logistic Regression, SVM and ANN were implemented in [22], to predict the movement of Bitcoin prices one hour in advance. Using blockchain network-based features, the results showed that ANN outperforms other models with an accuracy of 55%. Finally, [23] has implemented random forests and generalized linear models to predict the sign of Bitcoin prices for daily and 10-min time intervals. They presented an accuracy of 98.7% for the predicting daily prices and 50-55% for 10-min time intervals prediction.

Considering the state of the art, this paper addresses two main challenges, namely, the need to provide an innovative model that predicts Bitcoin prices accurately by introducing new features that are not usually considered in the literature. Our proposed model indeed achieved better performance results than those available in the literature, while considering both a validation and a test datasets, due to the newly presented features as explained in details in section V.

III. Methodology

Considering the aim of this study, we propose a new hybrid model for Bitcoin prices prediction. Our proposed model is a hybrid of Hidden Markov Models (HMM) and Long Short Term Memory networks (LSTM). Furthermore, we exploit Genetic Algorithms (GA) to optimize and fine-tune the parameters of our neural network. Fig. 1 shows a block diagram of our proposed method.

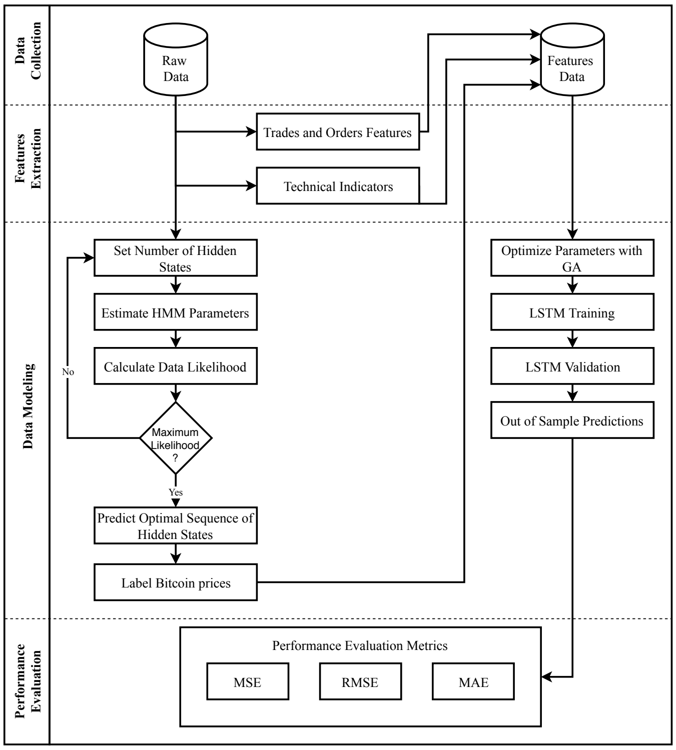

Fig. 1. Proposed Method using HMM and LSTM.

Looking at fig. 1, our proposed method includes four main phases. We start from Data Collection where Bitcoin raw data are collected and used to extract features in the Features Extraction phase. Followed by the Data Modeling phase where HMM, GA and LSTM networks are implemented. Finally, few metrics are introduced for Performance Evaluation. Combining these models is intended to better describe the target data and use the encapsulated information about historical prices through HMM, in addition to other features, and use such information to predict future prices through LSTM networks.

A. Hidden Markov Models

Hidden Markov Models are generative probabilistic models in which a sequence of observations Y is generated by a sequence of internal hidden states S. HMM are used to model time series data and have been implemented in various applications such as speech recognition systems, molecular biology and computer vision applications [24].

HMM are constructed by two assumptions. Firstly, a HMM assumes that an observation at time t, denoted by Y_t, was generated by a process whose state S_t is hidden from the observer. Secondly, it assumes that such state S_t satisfies a first-order Markov property; given the value of the previous state S_{t-1}, the current state S_t is independent of all the states prior to $t-1$. Similarly, the output of a HMM also follows the Markov property. Thus, the joint distribution of a sequence of hidden states and observations can be factorized by:

$$P(S_{1:T}, Y_{1:T}) = P(S_1)P(Y_1|S_1)\prod_{t=2}^{T} P(S_t|S_{t-1})P(S_t|Y_t) \tag{1}$$

To simplify, a HMM is defined by A, B, and π, and implicitly by the number of observations N, as well as the number of hidden states M. Where A represents the state transition probability $M \times M$ matrix, B represents the observations probability $M \times N$ matrix, and π is the initial state distribution. Thus, a HMM can be defined as:

$$\lambda = (A, B, \pi) \tag{2}$$

HMM are used to solve three fundamental problems [25], that can be summarized as follows:

- Problem 1: given the model $\lambda = (A, B, \pi)$, and a sequence of observations Y, determine the likelihood of the observed data to the given model.
- Problem 2: given the model $\lambda = (A, B, \pi)$, and a sequence of observations Y, determine the optimal sequence of hidden states underlying the Markov process.
- Problem 3: given a sequence of observations Y, estimate the model's parameters A, B and π.

In this study, given a sequence of observations, we will start by constructing a HMM, followed by calculating the likelihood of the data and then determining the optimal sequence of the hidden states, following the approach presented in [26] and illustrated in fig. 1, as a part of Data Modeling.

B. Genetic Algorithms

Genetic Algorithms are a type of optimization algorithms that are used to find the optimal solution(s) to a target problem [27], by mimicking the biological processes of evolution and natural selection. As GA are inspired from biological processes, terminologies such as chromosomes, populations, crossover and mutations are also adapted. Each potential solution to the optimization problem is represented by a chromosome, expressed in the form of binary strings [28].

The motivation behind using GA in this study is that they are powerful and more efficient than random search and exhaustive search algorithms [29]. Moreover, GA do not require other information than a solution representation and a fitness function with accordance to a given problem. This makes their applicability suitable for general problems, and especially appealing to our specific problem.

C. Long Short Term Memory Networks

Long Short Term Memory networks are a special kind of Recurrent Neural Networks (RNN). They are specifically designed to overcome common problems in RNN, i.e., vanishing gradients, exploding gradients, and long term dependencies, as they are able to remember information for more than 1000 time steps [30].

The cell of an LSTM network has mainly three gates; input gate, forget gate and output gate. Using these gates, LSTM has the ability to remove or add information to the cell state. Each gate is composed of a sigmoid layer and a point-wise multiplication operation, which outputs a number between 0 and 1 that indicates how much information should be passed or thrown away [31].

IV. Data Collection

To prepare the dataset, Bitcoin data has been collected from the Coinbase exchange market; one of the biggest platforms with a trading volume of 63 million USD per daily trading [32]. Coinbase Pro Public API (previously named GDAX) [33] was leverage to collect the real-time updates. All the available public data has been collected since January, 2018. The analysis will be carried out on a subset of the period, namely from August 20, 2018 to September 20, 2018.

For the scope of this study, our main focus is collecting:

- Market orders data: requests to buy or sell a specified amount of an asset (Bitcoin) at the best possible price, which includes ask/bid price and ask/bid amount.
- Market trades data: which includes buy/sell price and buy/sell amount. The i-th sample includes the price p_i and the amount of traded assets (or volume) v_i. A trade occurs when two orders at the opposite side, buy and sell, match. A trade can be either a perfect fill, meaning that both the price and the volume coincide, or a partial fill, meaning that only the prices are matching.

All the mentioned prices are in US Dollar (USD).

V. Features Extraction

The raw data that has been collected provides valuable information about Bitcoin orders and trades within the market. However, a further step is needed to extract information that can not be directly revealed by modeling such data alone. Thus, we further investigated the raw data and extracted a set of features that can be beneficial to the purpose of this study. The set of features can be divided into two categories, namely: Orders and Trades and the so-called Technical Indicators.

A. Orders and Trades

Orders and trades are features that were calculated based on the raw data. Let's define the best ask price (p^*_{ask}), i.e., the highest price that a buyer is willing to pay for a Bitcoin bid order, and the best bid price (p^*_{bid}), the lowest price that a seller is willing to accept for a Bitcoin ask order. The set of extracted features can be described as follows:

- Mid-Market Price mmp: it indicates the average market price; it can be calculated as follows:

$$mmp = \frac{p^*_{bid} + p^*_{ask}}{2} \tag{3}$$

  This feature represents the target variable to be predicted, where mmp at time t represents an accurate estimate of the true price of Bitcoin at that time instant. Fig. 2 illustrates the Bitcoin Mid-Market Price for the considered time period.

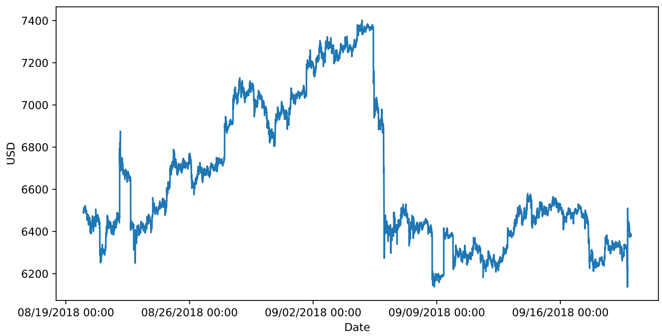

Fig. 2. Bitcoin Mid-Market Price in the time period from August 20, 2018 to September 20, 2018.

- Market Spread $mspread$: it indicates the difference between the best ask price and the best bid price, and can be calculated by:

$$mspread = p^*_{ask} - p^*_{bid} \tag{4}$$

  Smaller values of $mspread$ indicate a lower volatility, which result in an insignificant movement of the price.
- Ask/Bid Depth $D_{\{ask,bid\}}(t)$: indicate the number of available orders per ask side ($D_{ask}(t)$) and bid side ($D_{bid}(t)$), respectively, at time t[1]. Ask/Bid depth represents the liquidity of the market. A market is said to be deep when it is able to fulfill larger buy and sell orders before an order moves the price of Bitcoin. Fig. 3 shows the ask/bid depth for the considered time interval.

[1]The reference to time t will be dropped when not necessary.

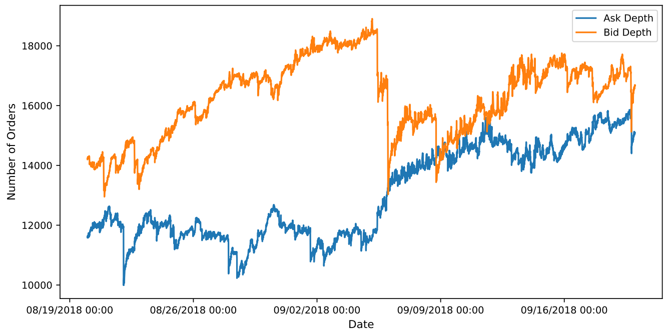

Fig. 3. Ask/Bid Depth in the considered time frame.

- Ask/Bid Volume $volume_{\{ask,bid\}}$: they indicate the total volume of ask and bid orders. Given a sample data, the ask/bid volume can be computed by:

$$volume_{ask} = \sum_{i=-0}^{-D_{bid}} v_i \tag{5}$$

$$volume_{bid} = \sum_{i=0}^{D_{ask}} v_i \tag{6}$$

- Weighted Ask/Bid Volume $weighted\_volume_{\{ask,bid\}}$: they indicate the weighted volume of ask and bid orders to better capture the relevance of orders to the movement of the price, and is calculated by:

$$weighted\_volume_{ask} = \sum_{i=-0}^{-D_{bid}} v_i . \frac{1}{mmp - p_i} \tag{7}$$

$$weighted\_volume_{bid} = \sum_{i=0}^{D_{ask}} v_i . \frac{1}{p_i - mmp} \tag{8}$$

- Depth Chart Quantization: generally, depth charts are bin charts that hold information about the cumulative supply and demand of an asset, at different prices. Thus, valuable information is encapsulated inside such charts that may not be tackled by the previously extracted features. A depth chart is composed of a horizontal axis that depicts Bitcoin prices $pc(i)$ and a vertical axis that depicts the corresponding tradable amount of Bitcoin $vc(p)$ at a specific price p, for both sides, namely, ask and bid. The quantization of the depth chart depends on which side of the chart is considered. For the ask side of the chart, the depth is quantized by collecting the tradable amounts of Bitcoin that are at or below a specific price and take their sum. As for the bid side, the depth is quantized by collecting the tradable amounts of Bitcoin that are at or above a specific price and take their sum. We distinguish prices and volumes in each chart with the notation $pc_{\{ask,bid\}}(i)$ and $vc_{\{ask,bid\}}(p)$.

To represent the quantization of the depth chart, we consider a number of bins equal to N_{bins} for each chart. We can thus calculate the width of the bins in each chart:

$$w_{ask} = \frac{pc(D_{ask}) - p^*_{ask}}{N_{bins}} \tag{9}$$

$$w_{bid} = \frac{p^*_{bid} - pc(-D_{bid})}{N_{bins}} \tag{10}$$

Thus:

$$bin_{ask}(i) = \frac{1}{w_{ask}} \sum_{p=p^*_{ask}+(i-1)\cdot w_{ask}}^{p^*_{ask}+i\cdot w_{ask}} vc_{ask}(p) \tag{11}$$

$$bin_{bid}(i) = \frac{1}{w_{bid}} \sum_{p=p^*_{bid}-(i-1)\cdot w_{bid}}^{p^*_{bid}-i\cdot w_{bid}} vc_{bid}(p) \tag{12}$$

As a trade-off between resolution and dimensionality, N_{bins} was set to 10. Fig. 4 shows the relevant quantities used for the definition of the chart and for the quantization of the bins.

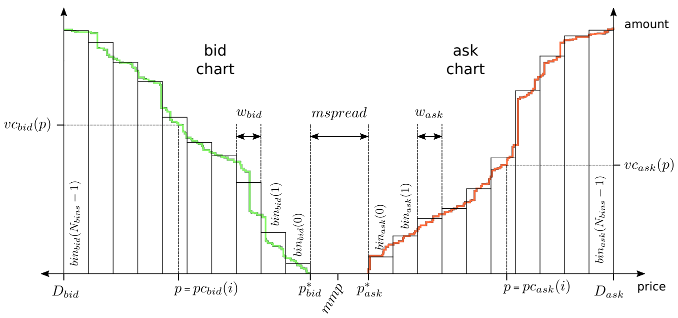

Fig. 4. Ask/Bid Depth Chart showing relevant quantities.

- Sell/Buy Count $count_{\{sell,buy\}}$: generally a trade is always considered to be aggressive, meaning that a specific trade would affect the movement of the price. Accordingly, and as the name implies, these features indicate the number of trades that have been generated by an aggressive sell or buy.
- Sell/Buy Traded Volume $traded\_volume_{\{sell,buy\}}$: given that the traded amount has an effect on liquidity, thus, the price movement, these features indicate the amount of Bitcoin that has been transacted due to an aggressive sell or buy.

Table I illustrates a number of selected features in terms of descriptive statistics in the considered time period, from August 20, 2018 to September 20, 2018.

B. Technical Indicators

Technical indicators can be identified as mathematical calculations, computed based on historical data of an asset, to predict the price movement. The following technical indicators were considered, and computed based on mmp;

TABLE I
SUMMARY STATISTICS FOR SELECTED FEATURES. VALUES ARE ROUNDED TO TWO DECIMAL PLACES.

Feature Name	Summary Statistics			
	Min	***Max***	***Mean***	***SD***
Ask Depth	9993.00	16000.00	13019.42	1527.40
Bid Depth	12951.00	18906.00	16249.11	1289.03
Ask Weighted Volume	3.14	33807.82	2853.07	3820.77
Bid Weighted Volume	21.60	36863.72	2263.84	2936.62
Sell Traded Volume	0.00	927.03	5.20	17.44
Buy Traded Volume	0.00	787.97	5.20	19.03
Market Spread	-94.97	-0.01	-0.06	0.79
Mid Market Price	6136.25	7402.01	6639.98	335.49

- Simple Moving Average sma: it indicates the arithmetic moving average of the price of an asset, that can be calculated in a specific time period. It is normally used to smooth out price fluctuations. Taken into account the frequency of our data, two sma's were calculated, namely $sma6$ and $sma16$, which indicate the moving averages of mmp at time periods 6 (every 12 minutes) and 16 (every 32 minutes) respectively. Given n, the time period considered, sma can be calculated by:

$$sma = \frac{mmp_1 + mmp_2 + ... + mmp_n}{n} \tag{13}$$

- Exponential Moving Average ema: it indicates the exponential weighted moving average of the price of an asset, by placing exponentially decreasing weights to the prices. Such weights represent higher weights to most recent prices, in the sense that recent prices have a higher significance in predicting future ones. Given n, the time period considered, ema can be calculated by:

$$ema_t = \alpha[mmp_1 + (1-\alpha)mmp_2 + ... + (1-\alpha)^{n-1}mmp_n] \tag{14}$$

where α represents the degree of weighting decrease at each point. Similarly, taken into account the frequency of our data, ema, ema_{12}, ema_{26} were calculated and indicate the exponential moving average at time periods 1 (every 2 minutes), 12 (every 24 minutes) and 26 (every 52 minutes) respectively.
- Moving Average Convergence Divergence $macd$: indicates the relationship between the previously calculated ema's, where it can be computed simply by finding the difference between ema_{26} and ema_{12}. This indicator helps in understanding bearish (prices expected to fall) and bullish (prices expected to rise) price movements.
- Bollinger Bands $band_{\{upper,lower\}}$: Bollinger Bands are lines that are associated to two standard deviations (sd) plotted away from the sma of the price of an asset. Usually, almost 90% of the original prices are contained within these two bands. Thus, breakouts that fall either under or above these bands indicate a major event that affected the price movement. Bollinger Bands are typically calculated by:

$$band_{upper} = sma_{21} + (sd_{20} * 2) \tag{15}$$

$$band_{lower} = sma_{21} - (sd_{20} * 2) \tag{16}$$

- Momentum $momentum$: it indicates the speed at which a price is changing. It it simply computed by subtracting 1 from mmp.

Fig. 5 shows some of the previously mentioned technical indicators for the last 500 data points, from September 19, 2018 at 07:22:00 a.m. to September 20, 2018 at 12:00:00 a.m.

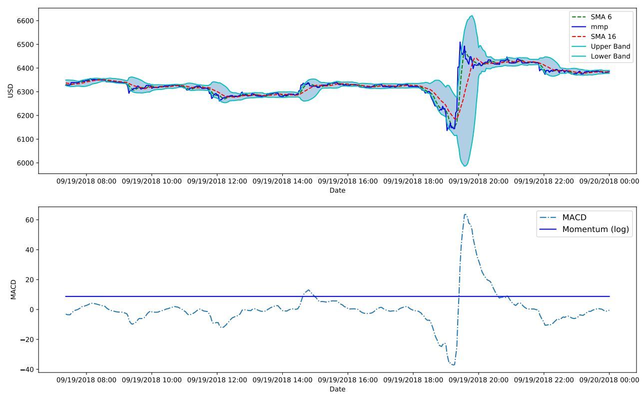

Fig. 5. Mid Market Price Technical Indicators.

VI. Data Modeling

Based on the set of features presented in section V, two modeling approaches were implemented, namely, descriptive modeling using HMM and predictive modeling using LSTM. This section describes the two models in details.

A. Descriptive Modeling with Hidden Markov Models

To better describe the movement of the prices and assess their evolution over time, we further explore our target variable mmp by modeling the two variables used to compute it, namely, best ask price (p_0) and best bid price (p_{-0}). For that purpose, we used HMM through which we create a feature by clustering the prices, with respect to a pre-defined number of hidden states.

Let y_t^i be the best price of each side i where $i = \{ask, bid\}$ at time t where $t = (1, 2, \ldots, T)$. We assume that the vectors Y^i are independent among each other, and each follows a Markov process, specified by the joint probability distribution, independently across i:

$$P(S^i_{1:T}, Y^i_{1:T}) = P(S^i_1)P(Y^i_1|S^i_1)\prod_{t=2}^{T} P(S^i_t|S^i_{t-1})P(S^i_t|Y^i_t) \tag{17}$$

Finally, given the nature of our data, we assume that each distribution $P(Y_1^i|S_1^i)$ is a multivariate Gaussian.

To construct our model, we start by addressing Problem 3, as explained in section III-A. Accordingly, using Baum-Welch algorithm, and given Y^i, the maximum likelihood estimates of the HMM parameters were computed using the Expectation Maximization (EM) algorithm [34], assuming a pre-defined number of hidden states M.

Once the parameters of our HMM are estimated, the likelihood of Y^i can be calculated using the Forward Backward algorithm, under the previously estimated parameters, followed by predicting the optimal sequence of the hidden states, which will represent a new feature $state$ added to the set of extracted features previously defined. Given the nature of the data, we assumed the number of hidden states to be 2, which in fact, proved to have the highest likelihood compared to other presumed number of states.

Table II shows the mean and variances of the prices per each of the hidden states. Moreover, fig. 6 illustrates the set of prices plotted with respect to the hidden state assigned by our HMM.

TABLE II
Descriptive Statistics of Hidden States. Values are rounded to three decimal places.

Hidden State	Summary Statistics for Prices per State			
	Best Ask Price		Best Bid Price	
	Mean	***Variance***	***Mean***	***Variance***
0	6395.035	9595.902	6394.976	9598.216
1	6985.890	53543.020	6985.834	53547.423

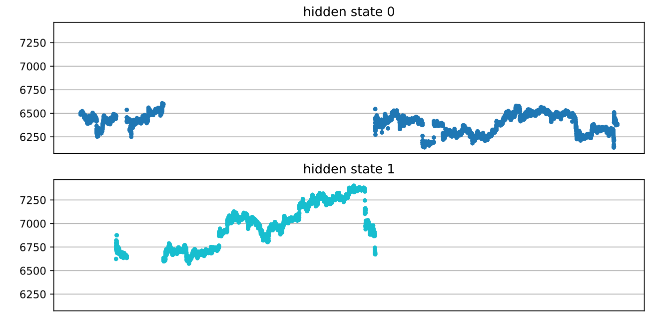

Fig. 6. Prices Plotted per Hidden State.

Specifically, fig. 7 illustrates the first 4500 data points of best ask price, clustered by the hidden states.

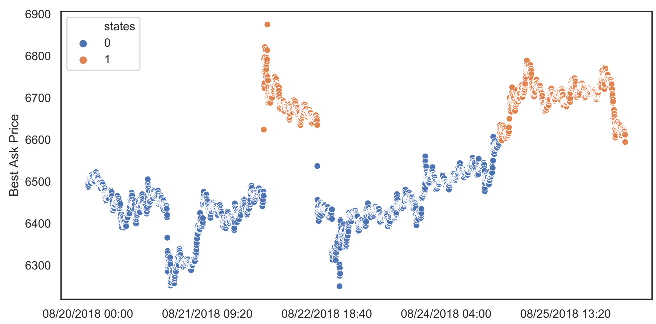

Fig. 7. Best Ask Price Clustered by HMM States.

B. Genetic Algorithm for LSTM Optimization

Following the logic explained in section III-B, and adapting the implementation from [35] and [36], we optimized a number of LSTM parameters before training it with the final set of features. The optimized parameters include: number of epochs (100), batch size (10), number of layers (3), number of neurons (100), dropout rate (0.2), optimizer (`adam`), loss (mean squared error) and evaluation metrics (mean absolute error).

C. Predictive Modeling with LSTM

Before training our optimized LSTM with the final set of features, a final step is needed: the introduction of the log returns ($log\_returns$). Given the nature of our data, as well as the target model, we compute the log returns of our target variable mmp by:

$$log\_return_{H(t)} = log(\frac{mmp(t)}{mmp(t-H)}) \tag{18}$$

where H is the prediction horizon; which defines how far ahead the model predicts in the future.

Accordingly, the final set of features is composed of $22,321$ data points and 52 features. 70% of the data has been used for training our LSTM (15610 data points), 20% has been used for validation (4683 data points), while the final 10% (2008 data points) was used for testing as an out of sample dataset. Additionally, dropout blocks were used between the hidden layers to avoid over-fitting, paired with an early stopping mechanism.

VII. Experimental Results

In this section we present the results of implementing the proposed model. It should be noted that the presented results are based on the out of sample dataset, unless stated otherwise, to show the true unbiased performance of our model.

A. Performance Evaluation Metrics

To evaluate the performance of our proposed model, the following metrics are considered and calculated as follows:

- Mean Squared Error (MSE):

$$MSE = \frac{1}{N_{sample}} \sum_{t}^{T} (\frac{mmp(t) - \widehat{mmp(t)}}{mmp(t)})^2 \tag{19}$$

- Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N_{sample}} \sum_{t}^{T} (\frac{mmp(t) - \widehat{mmp(t)}}{mmp(t)})^2} \tag{20}$$

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{N_{sample}} \sum_{t}^{T} |mmp(t) - \widehat{mmp(t)}| \tag{21}$$

Where N_{sample} is the size of the sample used to calculate these measurements.

Accordingly, these metrics are used to evaluate the performance of our model to assess the error rate. Additionally, in order to provide an unbiased sense of model's performance, these measurements are computed based on an out of sample (test) dataset that was not used to neither train nor fine-tune our model.

B. Description of results

Fig. 8 shows a comparison between the actual prices and 1-step ahead predictions where $H = 1$. As illustrated, the predictions of our proposed model are close to the actual ones and the movement of the prices is somewhat consistent. Moreover, to compare the performance of our model to more traditional time-series forecasting models, we implemented an ARIMA model and used our final set of features for training and predicting based on an out of sample dataset. Similarly, we implemented a Genetic Algorithm-optimized conventional LSTM to evaluate the importance of our proposed model.

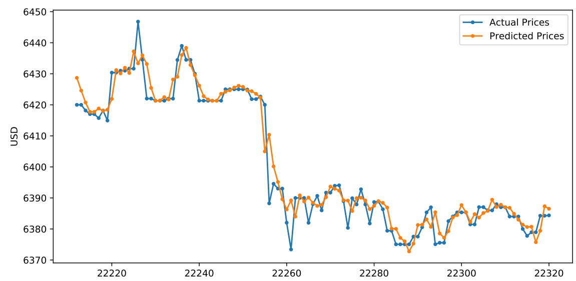

Fig. 8. Results for 1-Step Ahead Prices Prediction.

Accordingly, table III shows the performance of the three implemented models. As illustrated, our proposed model decreased the error rate significantly compared to ARIMA and the conventional LSTM, which proves the impact of HMM on enhancing the performance of a conventional LSTM.

TABLE III
Performance Evaluation for Implemented Models. Values are rounded to three decimal places.

Model Name	Performance Metrics		
	MSE	***RMSE***	***MAE***
ARIMA	20153.722	141.964	112.060
LSTM	49.089	7.006	2.652
HMM-LSTM	33.888	5.821	2.510

As a further step, we tested the performance of our model for multi-step prediction where $H = 2$. The results are illustrated in fig. 9. As expected, the performance was affected where MSE increased to 63.574. The rise is due to the iterative structure of LSTM where the prediction of one layer is passed to the next, thus, the error is accumulated to a larger number after two time steps ahead compared to only one step. However, looking back at the predicted results compared to the actual ones in fig. 9, it is safe to say that the predictions are realistic even with a relatively higher error rate.

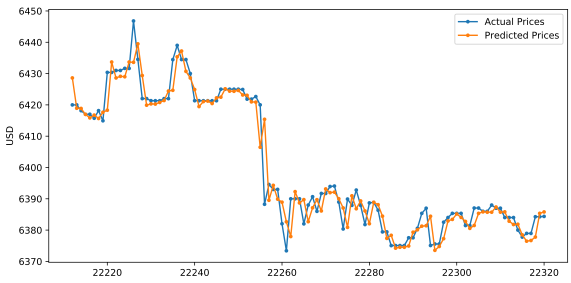

Fig. 9. Results for 2-Step Ahead Prices Prediction.

VIII. Conclusions

In this paper, we have proposed a new model that aims at predicting Bitcoin prices, based on Hidden Markov Models and Genetic Algorithm-optimized LSTM. The goal is to allow optimized M2M payments in the context of the Internet of Things domain. Our model addresses the dynamics of Bitcoin prices from a descriptive point of view, through Hidden Markov Models, by capturing insightful information that is hidden and cannot be directly seen nor extracted. Such valuable information is then integrated with LSTM, one of the typical methodologies of deep learning, to address the issue from a predictive point of view. GA was employed to optimize the parameters of LSTM as they can have a crucial effect on the performance of the model.

Experimental results showed that our proposed model has the lowest MSE, $RMSE$ and MAE between all three implemented models, which proves that our proposed model can be effective for Bitcoin prices prediction.

For future work, our model can be easily extended to consider additional features that can be extracted from the Blockchain to provide information about the internal details of Bitcoin transactions. Moreover, our model can be adapted to study the dynamics of different cryptocurrencies such as IOTA, that is specifically built to be used in the IoT domain.

References

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[2] K. Karimi and G. Atkinson, "What the Internet of Things (IoT) needs to become a reality," White Paper, FreeScale and ARM, 2013, pp. 1-16.
[3] D. Fakhri and K. Mutijarsa, "Secure IoT Communication using Blockchain Technology," 2018 International Symposium on Electronics and Smart Devices (ISESD), Bandung, 2018, pp. 1-6.
[4] K. Biswas and V. Muthukkumarasamy, "Securing Smart Cities Using Blockchain Technology," 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, NSW, 2016, pp. 1392-1393.
[5] M. Conoscenti, A. Vetrò and J. C. De Martin, "Blockchain for the Internet of Things: A systematic literature review," 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), Agadir, 2016, pp. 1-6.
[6] Y. Zhang and J. Wen, "The IoT Electric Business Model: Using Blockchain Technology for the Internet of Things," Peer-to-Peer Networking and Applications, 2017, pp. 983–994.
[7] A. Ensor, S. Schefer-Wenzl and I. Miladinovic, "Blockchains for IoT Payments: A Survey," 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 2018, pp. 1-6.
[8] P. Giudici and I. Abu-Hashish, "What determines bitcoin exchange prices? A network VAR approach," 2018, Finance Research Letters.
[9] P. Joon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," 2016.
[10] Z. Hao, R. Ji and Q. Li, "FastPay: A Secure Fast Payment Method for Edge-IoT Platforms using Blockchain," 2018 IEEE/ACM Symposium on Edge Computing (SEC), Seattle, WA, 2018, pp. 410-415.
[11] "The raiden network," https://raiden.network/, 2018.
[12] "Counterparty," https://counterparty.io/, 2018.
[13] "Omni Layer," https://www.omnilayer.org/, 2017.
[14] C. Wu, C. Lu, Y. Ma and R. Lu, "A New Forecasting Framework for Bitcoin Price with LSTM," 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, Singapore, 2018, pp. 168-175.
[15] E. S. Karakoyun and A. S. Cibikdiken, "Comparison of ARIMA Time Series Model and LSTM Deep Learning Algorithm for Bitcoin Price Forecasting," 2018 Multidisciplinary Academic Conference(MAC-EITAI), Prague, Czech Republic, 2018, pp. 171-179.
[16] S. Lahmiri and S. Bekiros, "Cryptocurrency Forecasting with Deep Learning Chaotic Neural Networks," Chaos, Solitons & Fractals: an interdisciplinary journal of nonlinear science, 2019, pp. 35-40.
[17] T. Guo, A. Bifet and N. Antulov-Fantulin, "Bitcoin Volatility Forecasting with a Glimpse into Buy and Sell Orders," 2018 IEEE International Conference on Data Mining (ICDM), Singapore, Singapore, 2018, pp. 989-994.
[18] J. Huisu, J. Lee, H. Ko, and W. Lee, "Predicting Bitcoin Prices by Using Rolling Window LSTM model," 2018 Data Science in FinTech (DSF), London, United Kingdom, 2018.
[19] T. Phaladisailoed and T. Numnonda, "Machine Learning Models Comparison for Bitcoin Price Prediction," 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), Kuta, 2018, pp. 506-511.
[20] S. McNally, J. Roche and S. Caton, "Predicting the Price of Bitcoin Using Machine Learning," 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Cambridge, 2018, pp. 339-343.
[21] D. Shah and K. Zhang, "Bayesian regression and Bitcoin," 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, 2014, pp. 409-414.
[22] A. Greaves and B. Au, "Using the bitcoin transaction graph to predict the price of bitcoin," 2015.
[23] I. Madan, S. Saluja and A. Zhao, "Automated bitcoin trading via machine learning algorithms," 2015.
[24] Z. Ghahramani, "An introduction to hidden Markov models and Bayesian networks," Hidden Markov models: applications in computer vision, 2001, pp. 9-41.
[25] M. Stamp, "A Revealing Introduction to hidden markov models," Department of Computer Science San Jose State University, 2004, pp. 26-56.
[26] P. Giudici and I. Abu Hashish, "A Hidden Markov Model to Detect Regime Changes in Cryptoasset Markets," Unpublished, 2018.
[27] J. Carr, "An Introduction to Genetic Algorithms," 2014.
[28] S. K. Pal and P.P. Wang, "Genetic Algorithms for Pattern Recognition," CRC Press: Boca Raton, FL, USA, 1996, pp. 336.
[29] K. E. Kinnear, "Advances in Genetic Programming," MIT Press: Cambridge, pp. 3-17.
[30] H. Chung and K. Shin, "Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction, " Sustainability, 2018, 10(10), pp.3765.
[31] C. Olah, "Understanding LSTM Networks," http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.
[32] "Top 100 Cryptocurrency Exchanges by Trade Volume," https://coinmarketcap.com/rankings/exchanges/, Retrieved: March 20^{th}2019.
[33] "Coinbase Pro Public API," https://docs.pro.coinbase.com/?r=1.
[34] T. P. Mann, "Numerically stable hidden Markov model implementation," An HMM scaling tutorial, 2006, pp. 1-8.
[35] M. Harvey, "Evolve a neural network with a genetic algorithm," https://github.com/harvitronix/neural-network-genetic-algorithm, 2017.
[36] A. Osipenko, "Nuero-Evolution," https://github.com/subpath/neuro-evolution, 2018.