Adaptive Fuzzy Level Set Algorithm for Bitcoin Realized Volatility Modeling and Forecasting*

Leandro Maciel[1][0000−0002−1900−7179], Guilherme Freitas[2][0009−0007−2581−3082], Vinicius Nazato[1][0009−0009−1270−6039], and Fernando Gomide[3][0000−0001−5716−4282]

[1] School of Economics, Business, Accounting and Actuary, University of São Paulo
leandromaciel@usp.br, vini_simoes@usp.br
[2] Institute of Matemathics and Statistics, University of São Paulo, Brazil
gv.freitas2003@usp.br
[3] School of Electrical and Computer Engineering, University of Campinas, Brazil
gomide@unicamp.br

Abstract. This paper addresses a novel approach to cryptocurrency risk management. An adaptive fuzzy model based on level sets is suggested to model and forecast the realized volatility of Bitcoin. The model, referred to as adaptive level set model (ALSM), is a rule-based fuzzy inference system that uses the concept of level sets to determine the model output in a data-driven and adaptive manner. One-step-ahead forecasts of realized volatility generated by the ALSM model are evaluated in terms of accuracy and compared to alternative machine learning models, an evolving fuzzy model, and the heterogeneous autoregressive (HAR) model. HAR serves as the baseline for realized volatility forecasting evaluation. The results indicate that the ALSM model achieves the highest accuracy among all competing approaches, highlighting its potential as a valuable tool to assist investors in forecasting risk in Bitcoin market.

Keywords: Adaptive Fuzzy Modeling · Realized Volatility · Level Set · Bitcoin · Forecasting.

1 Introduction

In January 2025, CoinMarketCap[4] reported that the cryptocurrency market has reached a total market capitalization of USD 3.5 trillion. Bitcoin (BTC) remains the dominant cryptocurrency in this market. During the same period, the share of Bitcoin in the market of cryptocurrencies was approximately 58%. These figures highlight the consolidation of cryptocurrencies in the global financial market as

* Supported by the Brazilian National Council for Scientific and Technological Development (CNPq) grant 302467/2019-0 and by the Ripple Impact Fund, a donor-advised Silicon Valley Community Foundation fund, for grant 2018-196450(5855), as part of the University Blockchain Research Initiative, UBRI.

[4] Source: https://coinmarketcap.com. Access on 28th January, 2025.

digital assets that have become integral parts of investment portfolios [16, 10, 19].

One of the main characteristics of the cryptocurrency market is its significant volatility compared to other markets such as stocks and commodities [1]. The volatility of a financial asset refers to the variability in its price returns. Higher volatility indicates a greater likelihood of both large positive and negative returns. In finance, volatility is a crucial factor as it measures one of the key elements in decision making: the risk-return trade-off of assets. In addition, volatility is widely used in risk management, portfolio construction, and asset pricing [8, 33, 25].

In the cryptocurrency market, volatility is influenced by technological advances, market sentiment, and regulatory changes [34]. Among these factors, market sentiment plays a central role [20]. For example, the price of Bitcoin dropped by 12% after the announcement that Tesla would no longer accept it as a payment. Thus, the volatility of cryptocurrencies is affected by shocks, noise, news, changes in investor expectations, and other highly uncertain factors [28].

Volatility is an unobservable variable that must, therefore, be estimated using models. Approaches to volatility modeling include [11]:

1. Historical methods to estimate volatility via the standard deviation of past returns data using exponential smoothing methods such as the exponential weighted moving average (EWMA) model [17];
2. Financial econometrics approaches such as generalized autoregressive heteroskedasticity (GARCH) time series models, which assumes a generating process for volatility to capture stylized facts observed in financial return series such as high kurtosis and volatility clustering [23];
3. Stochastic volatility methods that asset price dynamics using stochastic differential equations [12];
4. Implied volatility approaches that extract return variance information from the prices of market-traded assets such as option contracts [7];
5. Realized volatility-based (RV) mechanisms to calculate return variability from the intraday prices using heterogeneous autoregressive (HAR) regression framework [6].

Recent studies in the literature suggest that realized volatility-based models give important information to estimate risk measures, especially due to their ability to capture intraday fluctuations [13, 21, 5]. In the case of highly volatile dynamics of cryptocurrencies intraday variability becomes even more critical and essential for risk analysis and asset pricing.

Many researchers have addressed cryptocurrency volatility modeling and forecasting. For instance, the relevance of volatility spillover effects among cryptocurrencies to forecast realized volatility of Bitcoin is investigated in [26]. The results indicate that Bitcoin volatility models incorporating these effects have greater explanatory power in-sample, and significantly enhance the accuracy of short-term forecasts. Incorporation of external predictive information from other cryptocurrency markets to forecast realized volatility of Bitcoin is pursued in [32]. Using six multivariate approaches, the paper reports that scaled principal component analysis (SPCA) significantly improves the predictive accuracy of the standard HAR model.

Various GARCH models and HAR Bitcoin volatility models are evaluated in [5]. The results indicate that exponential GARCH (EGARCH) and asymmetric power GARCH (APARCH) outperform other GARCH models, while HAR models based on realized variance consistently surpass GARCH models that rely on daily data, especially for short-term volatility forecasts. [14] compares Bitcoin volatility forecasting with traditional econometric and machine learning models. Deep learning models outperform GARCH models across different forecasting horizons. Furthermore, long-short-term memory (LSTM) outperforms the heterogeneous autoregressive (HAR) model. Neural networks generally exhibit better performance when compared against classic statistic-based methods.

A recent work [4] analyzes the role of daily indices in forecasting Bitcoin volatility compared with monthly and weekly indices. Daily indices of economic policy uncertainty (EPU) and geopolitical risk (GPR) outperformed monthly indices in explanatory power and predictive accuracy. Combining indices with different frequencies significantly improved predictive performance.

This paper advances the current state of the art by introducing a new algorithmic approach to model and forecast the realized volatility of Bitcoin, namely, the adaptive level set fuzzy modeling (ALSM) and forecasting method and algorithm. Originally outlined in [22], the method consists of a rule-based fuzzy model whose outputs, differently from the linguistic and functional fuzzy rule-based models, are computed using output functions associated with each rule of the rule base. The ALSM model uses the concept of level sets – first introduced in [30] – and uses data to develop a function that maps activation levels of the rules into values in the output space [22]. The approach is adaptive, processing the data sequentially, which makes it particularly suitable to model time-varying dynamic systems, such as financial asset volatility market dynamics. In fact, [24] demonstrates the predictive power of the ALSM model to forecast future asset prices. This paper evaluates the ALSM model in forecasting the realized volatility of Bitcoin, offering an innovative approach in the volatility literature, and demonstrating the usefulness of fuzzy models in offering effective solutions in highly uncertain and complex environments. Using intraday Bitcoin data with a 5-minute frequency, the paper computes one-step-ahead forecasts of realized volatility. The results of the ALSM model are compared with HAR as the baseline model, and with alternative machine learning and evolving fuzzy forecasters.

The paper is organized as follows. Section 2 addresses the ALSM model. Section 4 presents the methodology, including data, alternative predictive models, and evaluation metrics of predictive quality. The results are addressed in Section 4.2. Section 5 concludes the paper summarizing its contributions and suggesting issues for future investigation.

2 Fuzzy Adaptive Level Set Modeling

This section briefly reviews the conceptual idea of fuzzy level-set-based modeling and an adaptive algorithm to develop fuzzy rule-based models from data.

2.1 Fuzzy Level Set Modeling

The level set method uses fuzzy rule-based models of the form

$$\mathcal{R}_i: \text{ if } \mathbf{x} \text{ is } \mathcal{A}_i \text{ then } y \; is \; \mathcal{B}_i, \tag{1}$$

where $i = 1, 2, \ldots, N$, and $\mathcal{A}_i$ and $\mathcal{B}_i$ are convex fuzzy sets with membership functions $\mathcal{A}_i(\mathbf{x}) : \mathcal{X} \rightarrow [0, 1]$ and $\mathcal{B}_i(y) : \mathcal{Y} \rightarrow [0, 1]$, respectively.

The development of fuzzy rule-based models needs the specification of the number of rules N, the membership functions for the antecedent variables $\mathcal{A}_i(\mathbf{x})$, and the membership functions $\mathcal{B}_i(y)$ of the consequent variables of each rule $\mathcal{R}_i$. Knowledge-based, grid, hierarchical, or clustering procedures can be used to granulate the input-output space and to determine the membership functions. Once the membership functions $\mathcal{A}_i$ and $\mathcal{B}_i$ are given, the fuzzy level set modeling method proceeds as follows [31].

Given input data $\mathbf{x} \in \mathcal{X}$:

1. Compute the activation level τ_i of each rule $\mathcal{R}_i$
$$\tau_i = \mathcal{A}_i(\mathbf{x}). \tag{2}$$
2. For each level τ_i, find the corresponding level set $\mathcal{B}_{\tau_i}$
$$\mathcal{B}_{\tau_i} = \{y | \tau_i \leq \mathcal{B}_i(y)\} = [y_{il}, y_{iu}]. \tag{3}$$
where y_{il} and y_{iu} are the bounds of the interval for which $\mathcal{B}_i(y)$ is greater than or equal to τ_i.
3. Compute the midpoint of the level set
$$m_i(\tau_i) = \frac{y_{il} + y_{iu}}{2}. \tag{4}$$
4. Compute the model output $\hat{y}$ as
$$\hat{y}(\tau) = \frac{\sum_{i=1}^{N} \tau_i m_i(\tau_i)}{\sum_{i=1}^{N} \tau_i}. \tag{5}$$

Figure 1 summarizes the idea. The expression (5) is a particular case of a general mapping $\mathcal{F} : [0, 1] \rightarrow \mathcal{Y}$ in which $\mathcal{F}_i(\tau_i) = m_i(\tau_i)$, called the output function, is the midpoint of the interval $\mathcal{B}_{\tau_i} = [y_{il}, y_{iu}]$. In the example of Figure 1, if m_i is computed for all $x \in \mathcal{X}$, then the output function is the affine function

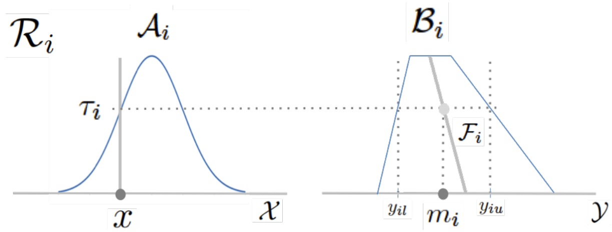

Fig. 1. Fuzzy level set modeling.

$\mathcal{F}_i$ in $[0, 1] \times \mathcal{Y}$. Other choices are possible, especially in data-driven modeling frameworks. The reason to adopt the midpoint is discussed in [18]. In general, level set fuzzy models use output functions to produce outputs $\hat{y}(\tau)$

$$\hat{y(\tau)} = \frac{\sum_{i=1}^{N} \tau_i \mathcal{F}_i(\tau_i)}{\sum_{i=1}^{N} \tau_i} = \mathcal{F}(\tau). \tag{6}$$

Note that $\mathcal{F}$ maps membership degrees to elements of the output domain $\mathcal{Y}$, which is very different from what is done by linguistic and functional fuzzy models.

Data-driven level set fuzzy modeling [17] relies on a data set $\mathcal{D} = \{(\mathbf{x}^k; y^k)\}$, $\mathbf{x}^k \in R^p$, $y^k \in R$ such that $y^k = f(\mathbf{x}^k)$, for $k = 1, 2, \ldots, K$, to develop a fuzzy model $\mathcal{F}$ that approximates the function f in $\mathcal{D}$. There is no need to specify $\mathcal{B}_i$ because the output functions $\mathcal{F}_i$ in (6) can be estimated from $\mathcal{D}$. This paper assumes affine output functions

$$\mathcal{F}_i(\tau_i) = v_i \tau_i + w_i. \tag{7}$$

The coefficients v_i and w_i can be estimated using any proper procedure, for example, the recursive correntropy-based least squares algorithm.

2.2 Recursive Correntropy Least Squares Algorithm

The algorithm adopted in this paper employs a correntropy-based recursive least squares algorithm [27] [24] to compute the coefficients of the output functions of the rules. The details of the algorithm are as follows.

Let $\mathbf{x}^k$ be an input data and $\tau_i^k = \mathcal{A}_i(\mathbf{x}^k)$ the activation level of i-th rule in step k. Therefore, from (6) and (7), the output of the model $\hat{y}^k$ at step k is expressed as

$$\hat{y}^k = \frac{\tau_1^k(v_1^k \tau_1^k + w_1^k)}{s^k} + \ldots + \frac{\tau_N^k(v_N^k \tau_N^k + w_N^k)}{s^k}, \tag{8}$$

with the normalization factor $s^k = \sum_{i=1}^{N} \tau_i^k$.

Let $\boldsymbol{\theta}^k = [v_1^k, w_1^k, \ldots, v_N^k, w_N^k]^T$ be the vector with coefficients of the output functions of all N rules, and

$$\mathbf{a}^k = [(\tau_1^k)^2/s^k, \tau_1^k/s^k, \ldots, (\tau_N^k)^2/s^k, \tau_N^k/s^k], \tag{9}$$

denote the vector of normalized activation levels of the N rules. The recursive least squares algorithm based on correntropy proceeds as follows:

Initialize: $\boldsymbol{\theta}^0 = 0$, $\mathbf{P}^0 = \alpha\mathbf{I}$
For each step $k = 0, \ldots K$:

1. Read the data pair $(\mathbf{x}^k; y^k)$.
2. Compute the estimation error $e^k = \mathbf{a}^k\boldsymbol{\theta}^k - y^k$.
3. Set:

$$\psi^k = \frac{1}{\sqrt{2\pi}\zeta^3}\exp\left(-\frac{|e^k|^2}{2\zeta^2}\right), \tag{10}$$

   in which ζ is a user-selected value that determines the width of the kernel.
4. Compute the activation levels $\tau_i^k = \mathcal{A}_i(\mathbf{x}^k)$, $i = 1, \ldots, N$, and form the activation vector $\mathbf{a}^k$, as in (9).
5. Compute the gain matrix:

$$\mathbf{P}^k = \frac{1}{\lambda}\left(\mathbf{P}^{k-1} - \frac{\psi^k\mathbf{P}^{k-1}(\mathbf{a}^k)^T\mathbf{a}^k\mathbf{P}^{k-1}}{\lambda + \psi^k\mathbf{a}^k\mathbf{P}^{\mathbf{k-1}}(\mathbf{a}^k)^T}\right), \tag{11}$$

   where $\lambda \in [0, 1]$ is a forgetting factor.
6. Update the coefficients of the output functions:

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \psi^k\mathbf{P}^k(\mathbf{a}^k)^T(y^k - \mathbf{a}^k\boldsymbol{\theta}^k). \tag{12}$$

7. Compute the model output $\hat{y}^{k+1}$ using the expression

$$\hat{y}^{k+1} = \mathbf{a}^k\boldsymbol{\theta}^{k+1}. \tag{13}$$

The procedure requires initial estimates for the coefficients $\boldsymbol{\theta}^0$, and the initial gain matrix $\mathbf{P}^0$. Usually $\boldsymbol{\theta}^0 = \mathbf{0}$, and $\mathbf{P}^0 = \alpha\mathbf{I}$, where α is a sufficiently large constant, e.g., 10^3. The parameter λ is the forgetting factor that weighs the data instances in the sequence. A smaller λ places more emphasis on recent data, allowing the algorithm to track time-varying behaviors more [15]. Membership functions and their parameters can be chosen using fuzzy clustering or domain knowledge.

3 Volatility Modeling and Forecasting Models

3.1 Volatility modeling

This study analyzes the modeling and forecasting of Bitcoin's realized volatility. The realized volatility on day t, RV_t, is defined as [2]:

$$RV_t = \sqrt{\sum_{j=1}^{n} r_{t,j}^2}, \tag{14}$$

where $r_{t,j} = \ln(p_{t,j}) - \ln(p_{t,j-1})$, $j = 1, \ldots, n$, are the intraday log-returns, $p_{t,j}$ is the price on day t at time j, and n represents the number of intraday returns on day t.

Intraday data at a 5-minute frequency are used. Each day's data is discretized for the trading hours, which start at 5:00 AM and end at 12:00 PM. Thus, there are a total of 288 prices in a single day, implying that $n = 288$.

3.2 Heterogeneous Autoregressive Regression

The baseline forecast approach considered is the HAR model of [9]. The HAR method is an additive cascade model of volatility components defined over different time periods. This model effectively captures key empirical features of financial returns – long memory, fat tails, and self-similarity – in a parsimonious manner. Interestingly, the literature has highlighted the strong forecasting performance of HAR [6, 29, 9].

The HAR model is defined by [9]:

$$RV_{t+1} = \beta_0 + \beta_1 RV_t + \beta_2 RV_t^{(w)} + \beta_3 RV_t^{(m)} + \epsilon_t, \tag{15}$$

where β_0, β_1, β_2, and β_3 are parameters estimated using ordinary least squares; $RV_t^{(w)}$ and $RV_t^{(m)}$ represent the weekly and monthly realized volatilities, respectively; and ϵ_t is a white noise error term.

$RV_t^{(w)}$ and $RV_t^{(m)}$ are calculated as, respectively:

$$RV_t^{(w)} = \frac{1}{5}\left(RV_t + RV_{t-1} + RV_{t-2} + RV_{t-3} + RV_{t-4}\right), \tag{16}$$

$$RV_t^{(m)} = \frac{1}{22}\left(RV_t + RV_{t-1} + RV_{t-2} + RV_{t-3} + \ldots + RV_{t-21}\right), \tag{17}$$

By considering daily, weekly and monthly realized volatility, this approach accounts for realized volatility over time horizons longer than one day. The aim is to recognize the heterogeneity among traders. It identifies three primary volatility components: short-term traders with daily or higher trading frequency, medium-term investors who typically rebalance their positions weekly, and long-term agents with a characteristic time horizon of one month or more [9].

3.3 Machine Learning and Evolving Fuzzy Models

In addition to the HAR model, used as a baseline, and the ALSM model, which is the approach suggested in this paper, the following methods are also considered: Multilayer Perceptron (MLP) neural network, Support Vector Regression (SVR), and the evolving Takagi-Sugeno model (eTS+) [3]. SVR and MLP are traditional machine learning techniques, eTS+ is an adaptive fuzzy rule-based model that processes data sequentially, and continuously updates its structure and functionality as new data are input.

Similar to the HAR approach, one-step-ahead forecasts from SVR, MLP, eTS+, and ALSM are generated as:

$$RV_{t+1} = f\left(RV_t, RV_t^{(w)}, RV_t^{(m)}\right), \tag{18}$$

where $f(\cdot)$ represents the respective mapping function for each model (SVR, MLP, eTS+, and ALSM).

The form of (18) follows the same structure as the HAR approach. Alternatively, for SVR, MLP, eTS+, and ALSM, we also assess the following form:

$$RV_{t+1} = f\left(RV_t, RV_{t-1}, \ldots, RV_{t-p}\right), \tag{19}$$

where p is the number of lagged realized volatilities values considered.

4 Computational Results

4.1 Data and Error Measures

The data used in this paper consist of intraday Bitcoin price quotes at a 5-minute frequency. They were extracted from the Binance API (Application Programming Interface)[5], one of the leading cryptocurrency trading platforms worldwide. The dataset spans from 1/1/2018 to 11/30/2024, totaling 725,877 intraday quotes. This period was selected based on data availability at the time of extraction.

The one-step-ahead forecasts produced by the models are compared using the root mean squared error (RMSE) and mean absolute error (MAE) measures

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(RV_t - \hat{RV}_t\right)^2} \qquad \text{MAE} = \frac{1}{T}\sum_{t=1}^{T}\left|RV_t - \hat{RV}_t\right|$$

where RV_t is the actual realized volatility at time t and $\hat{RV}_t$ is the corresponding predicted value produced by the forecasters; T denotes the sample size.

4.2 Forecasting Performance Evaluation

The data were split into in-sample and out-of-sample sets. The in-sample period spans from January 1, 2018, to September 26, 2020, comprising a total of 1,000 observations, was used to train the models. The out-of-sample period, from September 27, 2020, to November 30, 2024, consisting of 1,526 observations, was used to evaluate the forecasting performance of the models. The data split rate follows the recommendation of [9] for developing HAR models, i.e., using 1,000 observations for training. One-step-ahead forecasts were obtained

[5] Source: https://binance-docs.github.io/apidocs/spot/en/#general-api-information. Access on December 4th, 2024.

sequentially by re-estimating the model parameters each day while maintaining a fixed rolling window of 1,000 observations. It is important to highlight that this approach was only applied to the HAR, SVR, and MLP models. In contrast, the eTS+ and ALSM models are inherently adaptive, meaning that they update their parameters whenever a new datum is input.

The models are developed considering two approaches. The first approach uses the same input variables as the HAR model, as in (18), while the second considers lags of the realized volatility itself as input, as in (19).

The SVR model uses the same input variables as the HAR (SVR3), with a regularization parameter value of 50, a penalty of 0.001 in the loss function, and a radial basis function (RBF) kernel. This same parametrization was used for the SVR model considering two-lag inputs, referred to as SVR2.

Similarly, the MLP uses the same inputs as the HAR model, MLP3, and also considers two-lag realized volatilities as input, denoted by MLP2. In both cases, the neural networks employ a single hidden layer with 10 neurons and ReLU activation function, and a linear output layer. The training was conducted using the backpropagation algorithm.

The eTS+ with the same three inputs as HAR, denoted eTS+3, uses a hyperparameter value of 0.05 which is the threshold needed to evaluate the quality of the clustering structure. In the two-lags ($p = 2$) input, denoted eTS+2 it uses $p = 2$ and a hyperparameter value of 0.07.

The ALSM with the same three inputs as HAR, denoted ALSM3, uses two rules with Gaussian membership functions, with a forgetting factor value of 0.99. The ALSM with two-lags input, denoted ALSM2, also uses two rules with Gaussian membership functions and a forgetting factor of 0.99.

Table 1 summarizes the forecasting performance of the methods in terms of error measures. The bold values indicate the best-performing model, the one with the lowest error measure. Among the models that follow the same structure as the HAR model – i.e., those that use past daily, weekly, and monthly realized volatility as inputs (SVR3, MLP3, eTS+3, and ALSM3) – the adaptive level set fuzzy model exhibited the lowest RMSE error metric (see Table 1). Based on the same metric, only the eTS+3 and ALSM3 models outperform the baseline HAR model. In terms of MAE, SVR3 achieves the best performance, and all proposed approaches surpass the HAR model.

Table 1. Forecasting performance evaluation

Method	HAR	SVR2	SVR3	MLP2	MLP3	eTS+2	eTS+3	ALSM2	ALSM3
RMSE	0.0372	0.0433	0.0431	0.0447	0.0389	0.0362	0.0361	**0.0346**	0.0353
MAE	0.0123	0.0115	**0.0113**	0.0149	0.0121	0.0119	0.0117	0.0115	0.0115

When considering models that use lagged daily realized volatility as inputs, the ALSM model remains the most accurate approach, yielding the lowest RMSE value (see Table 1). Notably, ALSM2 is the most precise among the alternative strategies in terms of RMSE. Again, based on this metric, the fuzzy models (eTS+2 and ALSM2) outperform the HAR model. Regarding MAE, SVR2 and ALSM2 exhibit the lowest error values.

Overall, the adaptive fuzzy approach appears as an effective method to forecast the realized volatility of Bitcoin. Figure 2 shows the daily forecasts generated by the two-lag ALSM model, highlighting its potential for risk management.

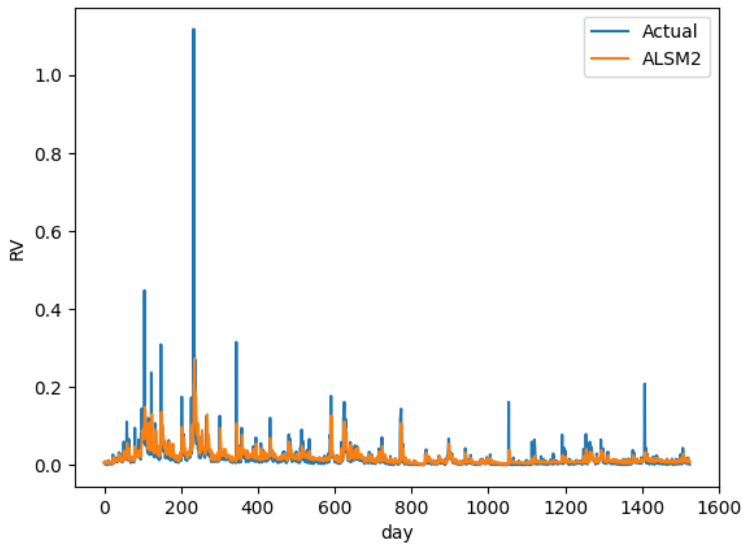

Fig. 2. ALSM daily forecasts for the two-lag inputs model (ASLM2).

5 Conclusion

The volatility of financial asset returns is a crucial risk measure in finance. Developing accurate models for volatility modeling and forecasting has been a persistent challenge for researchers, particularly for highly volatile, riskier assets such as cryptocurrencies. This paper has addressed a novel approach to forecast Bitcoin volatility using an adaptive fuzzy model based on the concept of level sets. The model is a fuzzy rule-based inference system whose outputs are computed using output functions that map the activation levels of the inputs into values in the output space.

Computational experiments focused on short-term, one-step-ahead, forecasting of the realized volatility of Bitcoin using intraday data. The performance of the adaptive level set model was compared with machine learning techniques and an evolving fuzzy model. The baseline model adopted was the heterogeneous autoregressive regression model – a widely used model in the realized volatility literature. The results suggest that the level set-based algorithmic approach achieves the highest performance. Future work should explore the use of the level set-based algorithm in other markets and asset classes. The use market sentiment variables in adaptive fuzzy modeling is also an issue to be investigated.

References

1. Ahmed, M.S., El-Masry, A.A., Al-Maghyereh, A.I., Kumar, S.: Cryptocurrency volatility: A review, synthesis, and research agenda. Research in International Business and Finance **71**, 102472 (2024)
2. Andersen, T.G., Bollerslev, T., Diebold, F.X., Labys, P.: The distribution of realized exchange rate volatility. Journal of the American Statistical Association **96**(453), 42–55 (2001)
3. Angelov, P.: Evolving Takagi-Sugeno fuzzy systems from data streams (eTS+). In: Angelov, P., Filev, D., Kasabov, N. (eds.) Evolving intelligent systems: Methodology and applications, pp. 21–50. Wiley & IEEE Press, Hoboken, NJ, USA (2010)
4. Aras, S., Özdemir, M.O., Çılgın, C.: Can joint modelling of external variables sampled at different frequencies enhance long-term bitcoin volatility forecasts? Finance Research Letters **73**, 106679 (2025)
5. Øverland Bergsli, L., Lind, A.F., Molnár, P., Polasik, M.: Forecasting volatility of bitcoin. Research in International Business and Finance **59**, 101540 (2022)
6. Branco, R.R., Rubesam, A., Zevallos, M.: Forecasting realized volatility: Does anything beat linear models? Journal of Empirical Finance **78**, 101524 (2024)
7. Chen, T., Deng, J., Nie, J.: Implied volatility slopes and jumps in bitcoin options market. Operations Research Letters **55**, 107135 (2024)
8. Chun, D., Cho, H., Ryu, D.: Volatility forecasting and volatility-timing strategies: A machine learning approach. Research in International Business and Finance **75**, 102723 (2025)
9. Corsi, F.: A simple approximate long-memory model of realized volatility. Journal of Financial Econometrics **7**(2), 174–196 (2009)
10. Cui, T., Ding, S., Jin, H., Zhang, Y.: Portfolio constructions in cryptocurrency market: A cvar-based deep reinforcement learning approach. Economic Modelling **119**, 106078 (2023)
11. Degiannakis, S., Floros, C.: Methods of Volatility Estimation and Forecasting, pp. 58–109. Palgrave Macmillan UK, London (2015)
12. Harasheh, M., Bouteska, A.: Volatility estimation through stochastic processes: Evidence from cryptocurrencies. The North American Journal of Economics and Finance **75**, 102320 (2025)
13. Hu, N., Yin, X., Yao, Y.: A novel har-type realized volatility forecasting model using graph neural network. International Review of Financial Analysis **98**, 103881 (2025)
14. Huang, Z.C., Sangiorgi, I., Urquhart, A.: Forecasting bitcoin volatility using machine learning techniques. Journal of International Financial Markets, Institutions and Money **97**, 102064 (2024)
15. Jung, L.: System identification: Theory for the user. Pearson, Upper Saddle River (1999)

16. Ko, H., Son, B., Lee, J.: Portfolio insurance strategy in the cryptocurrency market. Research in International Business and Finance **67**, 102135 (2024)
17. Lahmiri, S.: Modeling and predicting historical volatility in exchange rate markets. Physica A: Statistical Mechanics and its Applications **471**, 387–395 (2017)
18. Leite, D., Gomide, F., Yager, R.: Data driven fuzzy modeling using level sets. Proceedings of the IEEE International Conference on Fuzzy Systems, Padova, Italy (2022)
19. Li, D., Shi, Y., Xu, L., Xu, Y., Zhao, Y.: Dynamic asymmetric dependence and portfolio management in cryptocurrency markets. Finance Research Letters **48**, 102829 (2022)
20. Lin, X., Meng, Y., Zhu, H.: How connected is the crypto market risk to investor sentiment? Finance Research Letters **56**, 104177 (2023)
21. Lin, Y., Lin, Z., Liao, Y., Li, Y., Xu, J., Yan, Y.: Forecasting the realized volatility of stock price index: A hybrid model integrating ceemdan and lstm. Expert Systems with Applications **206**, 117736 (2022)
22. Maciel, L., Ballini, R., Gomide, F.: Adaptive level set fuzzy modeling. In: NAFIPS International Conference on Fuzzy Systems, Soft Computing, and Explainable AI, South Padre Island, TX, USA, in press. (2024)
23. Maciel, L.: Cryptocurrencies value-at-risk and expected shortfall: Do regime-switching volatility models improve forecasting? International Journal of Finance & Economics **26**(3), 4840–4855 (2021)
24. Maciel, L., Ballini, R., Gomide, F.: Adaptive fuzzy modeling and forecasting of financial time series. In: 2024 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr). pp. 1–7 (2024)
25. Oh, D.H., Park, Y.H.: Garch option pricing with volatility derivatives. Journal of Banking & Finance **146**, 106718 (2023)
26. Qiu, Y., Wang, Y., Xie, T.: Forecasting bitcoin realized volatility by measuring the spillover effect among cryptocurrencies. Economics Letters **208**, 110092 (2021)
27. Rong, H., Yang, Z., Wong, P.: Robust and noise insensitive recursive maximum correntropy-based evolving fuzzy system. IEEE Transactions on Fuzzy Systems **28**(9), 2277–2284 (2020)
28. Sakariyahu, R., Lawal, R., Adigun, R., Paterson, A., Johan, S.: One crash, too many: Global uncertainty, sentiment factors and cryptocurrency market. Journal of International Financial Markets, Institutions and Money **94**, 102028 (2024)
29. Xiao, J., Wen, F., Zhao, Y., Wang, X.: The role of us implied volatility index in forecasting chinese stock market volatility: Evidence from har models. International Review of Economics & Finance **74**, 311–333 (2021)
30. Yager, R.: An alternative procedure for the calculation of fuzzy logic controller values. Journal of the Japanese Soc. for Fuzzy Theory and Systems **3**, 736–746 (1991)
31. Yager, R.: An alternative procedure for the calculation of fuzzy logic controller values. Journal of Japan Society for Fuzzy Theory and Systems **3**(4), 736–746 (1991)
32. Yi, Y., He, M., Zhang, Y.: Out-of-sample prediction of bitcoin realized volatility: Do other cryptocurrencies help? The North American Journal of Economics and Finance **62**, 101731 (2022)
33. Zhang, Y., He, M., Wang, Y., Wen, D.: Model specification for volatility forecasting benchmark. International Review of Financial Analysis **97**, 103850 (2025)
34. Zhang, Z., Zhao, R.: Good volatility, bad volatility, and the cross section of cryptocurrency returns. International Review of Financial Analysis **89**, 102712 (2023)