Predicting the FTSO consensus price

Filip Koprivec
filip.koprivec@ijs.si
JSI, FMF, AFLabs
Ljubljana, Slovenia

Tjaž Eržen
erzen.tjaz@gmail.com
AFLabs
Ljubljana, Slovenia

Urban Mežnar
urban.meznar@aflabs.si
AFLabs
Ljubljana, Slovenia

ABSTRACT

The paper presents a system for predicting cryptocurrency consensus prices within the Flare Time Series Oracle (FTSO), a decentralized oracle solution running on Flare blockchain. By leveraging a combination of smoothing techniques and machine learning methodologies, we detail and analyze the construction and performance of our own provider. This paper presents the FTSO mechanism, and basic information about the game theoretic background together with rewarding and submission protocol. Lastly, we present our provider's prediction accuracy for each coin.

KEYWORDS

FTSO, schelling point, machine learning, regression, smoothings

1 INTRODUCTION

The blockchain and decentralized finance (DeFi) sectors have seen significant growth, but they share a common challenge: securely accessing data not directly included in transaction signatures. This issue, known as the *oracle problem* [4], hinders the broader adoption of blockchain technologies as it's typically difficult to obtain reliable off-chain data. While various on-chain protocols offer solutions, each has its trade-offs concerning security, accuracy, and data reliability. Traditional centralized oracles present risks like data manipulation, whereas fully decentralized alternatives often suffer from latency and higher costs.

This paper examines the Flare Time Series Oracle, a decentralized oracle that uses a schelling point mechanism to aggregate data from multiple providers [12]. In FTSO, data providers submit price estimates for assets every three minutes, with the system price determined as a weighted median of these submissions. Given the inherent price variability across exchanges and the indeterminate nature of asset prices within a three-minute window, there isn't a singular "correct" price. Providers aim to select a price close to the final median, incentivized by the reward system. This competitive environment, involving around 100 data providers, has shown resilience against market anomalies and exchange issues.

Our study investigates machine learning techniques to predict this final median price using exchange data, considering factors like trading volumes and historical provider behavior. Given the dynamic nature of the competition, our prediction methods are designed for adaptability. We further detail the workings of FTSO and its role in the evolving landscape of decentralized finance.

2 RELATED WORK

While no literature precisely addresses the Flare FTSO, the general oracle problem has been extensively studied. Caldarelli [5] highlights the challenges of the blockchain oracle problem. Ellul [8] delves into its role in decentralized finance. Zohar and Eyal [16] provide a comprehensive study, while Caldarelli's subsequent work [3] offers an overview of oracle research. Liu et al. [15] survey various oracle implementation techniques. Notably, Alagha [1] introduces a reinforcement learning model to enhance oracle reliability [12].

The main oracle solution provider is Chainlink, which addresses the oracle problem with enhanced security and scalability in Chainlink 2.0 [6]. Zhang et al. [14] also detail their approach, providing insights for evolving projects like Flare FTSO in the oracle domain.

3 FTSO PROTOCOL

The Flare Time Series Oracle plays an important role in Flare Network's data accuracy and decentralization. The reward mechanism is not only meant to incentivize participation but also to guarantee that the data provided to the network remains consistent, reliable, and manipulation free.

The protocol works in a series of discrete steps to decrease the performance hit on the whole network. Every 3 minutes marks the beginning of a new *price epoch*. Providers are mandated to submit their price estimates in a timely manner, ensuring that their submissions are accepted by the network for the current epoch. To maintain confidentiality and prevent other providers from viewing or copying their predictions, providers initially submit a salted hash of their predicted value.

Only after the price epoch has ended, providers reveal the actual submitted values. This reveal must be done in the first 90 seconds of the next price epoch - the reveal period of an epoch overlaps with the first half of the next submit epoch, but the two do not interfere. Revealed values are validated to check, that they correspond to actual submitted values otherwise they are discarded. After the reveal epoch ends, all the revealed values are combined and a network-wide price is calculated. The network thus gets fresh asset prices every 3 minutes with some delay due to the reveal period. Such data granularity is not sufficient for high-frequency trading but has proven sufficient for many financial and future market applications.

Data providers are incentivized to submit *good* prices by the network-wide rewarding system. Participants whose estimates fall in the middle two quartiles (IQR range) of the final price are eligible for rewards. In the initial phase, rewards are distributed from global network inflation, but on-demand pricing models for more exotic assets are being developed. The exact amount of FLR tokens distributed as rewards is determined by the provider's vote power and other governance policies [2]. To prevent network congestion, rewards and vote power changes are calculated every 3.5 days and claimed by data providers on demand.

The network and community are explicitly against defining what a correct price is to remove the vulnerability of the definition relying on a specific price source. Assets are denominated in $ with 5 decimal points of precision. Since most of the exchanges

Information Society 2023, 9–13 October 2023, Ljubljana, Slovenia

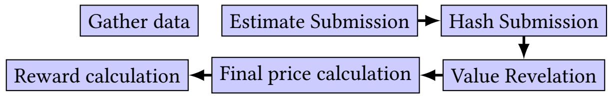

Figure 1: FTSO Submission and rewarding mechanism

quote a price that is accurate up to 3 decimal points, the configuration and no price explicit definition ensure, that submitted prices fall near the perceived fair market price, while still leaving room for competition on the last decimals.

One of the unique features of the Flare Network is the ability for token holders to delegate their votes to data providers. This means that even if a token holder does not actively participate in the estimation process, they can still earn FTSO rewards by delegating their voting power [9] and impact the price by selecting a specific data provider. It is important to note, however, that the voting power of a single data provider is limited to 2.5% to avoid too big of an individual impact.

The FTSO's reward mechanism is fostering decentralization and ensuring real-time data accuracy. Given that the core task revolves around predicting prices of other providers, participants not only need to make accurate predictions but also strategize to outperform others, making it a game of strategic decision-making. This challenge intriguingly sits at the crossroads of data science and game theory [7]. This article aims to delve deeper, exploring the multifaceted approaches and strategies employed to address this unique and complex problem.

4 DATA RETRIEVAL AND PREDICTION

4.1 Overview

The data retrieval process is a crucial step in our analysis. It involves collecting, processing, and preparing time series data, specifically price and timestamp pairs, for further analysis. This data is essential for understanding trends, making predictions, and deriving insights.

4.2 Data Retrieval Mechanism

The primary source of our data are the FTSO prices from previous epochs and current data from various cryptocurrency exchanges. However, the problem is multidimensional and layered. Not only do we need to decide on the specific time series models to employ, but we also face the intricate challenge of selecting which cryptocurrency exchanges to consider when training time series models for prediction. Each exchange has its own set of characteristics: trading volume, user base, regional influences, and even specific trading behaviors. Historical data shows, that providers are quick (on a sub-hour basis) to adapt to market opening and closing times and usually disregard after-hours trading prices on exchanges.

Furthermore, the reliability of data from each exchange can vary. Some exchanges might offer more consistent and clean data, while others might have gaps or anomalies. Deciding which exchanges to factor into our models and which to exclude becomes more than just a data-driven decision; it's a strategic choice that can significantly impact the accuracy and reliability of our predictions. This decision-making process requires a blend of quantitative analysis and domain expertise.

4.3 Data Processing and Smoothing Techniques

Once the data is retrieved, it undergoes several processing steps to ensure its quality and relevance for prediction. One of the primary challenges in time series forecasting is the inherent noise present in the data. Financial data is specifically prone to short-term spikes as low liquidity exchanges can experience large price deviations when market depth is limited. The spikes are quickly exploited by arbitragers, but price jumps - anomalies - are still available in the data and must be accounted for. To address this, we employ various smoothing techniques to filter out noise and highlight the underlying trends.

Exponential Moving Average (EMA): EMA is a type of weighted moving average that gives more weight to the most recent prices. The formula for EMA is:

$$EMA_t = \alpha \times P_t + (1 - \alpha) \times EMA_{t-1}$$

where α is the smoothing factor and P_t is the current price. In our system, the EMA vector and its alpha value are optimized using the `curve_fit` method from `scipy.optimize` library [11].

Savitzky-Golay Smoothing: This technique uses convolution to fit successive subsets of adjacent data points with a low-degree polynomial. It's effective in preserving the features of the distribution, such as heights and widths, making it suitable for our analysis [13].

Linear Interpolation: Linear interpolation is used to estimate values between two known values in a dataset. Our system employs a skew linear fit to interpolate missing or anomalous data points.

FFT Smoothing: The last smoothing method we've used is the Fast-Fourier smoothing.

Each of these methods has its own strengths and is chosen based on the specific characteristics of the data and the prediction requirements. So far, the only other smoothing method we've tried to incorporate is LOWESS (Locally Weighted Scatterplot Smoothing), which performed worse than the rest of the smoothing methods after training an overdetermined system on it (see 4.4). The mentioned methods were selected, as they are commonly used for smoothing the financial data [10], easily available in multiple scientific libraries, and offer good resilience against sudden spikes that are markets with low liquidity.

4.4 Prediction Mechanism

The core of our FTSO provider lies in its prediction mechanism. After smoothing the data using the techniques listed above, we adopt an overdetermined system approach for our predictions. This entails constructing a system of equations from the processed data and subsequently employing the least squares method to find the optimal prediction parameters.

Suppose we're training our time series over m epochs. Let $E \in \mathbb{R}^{m \times n}$ be a matrix where each column, denoted as $e_i \in \mathbb{R}^m$ (for $i \in \{1, ..., n\}$), represents the price vector for the i-th exchange across the m epochs. Define $\mathbf{v} \in \mathbb{R}^n$ as a vector that signifies the weights or contributions of each exchange to the forecasted price. Each entry, v_i in $\mathbf{v}$ corresponds to the significance of the i-th exchange, with the sum of all weights equating to 1.

Given the extensive epoch training data required for our model training and the limited availability of crypto exchanges (in the tens), we are dealing with an overdetermined system. In this context, we optimize the vector $\mathbf{v}$ using the least squares error method. The residual sum of squares evaluation function is optimized using the `fmin_cg` method from `scipy.optimize`, aiming to find the parameters that minimize the difference between the predicted values and the actual values in the training data.

Once the system is trained and the optimal parameters are found, these parameters are used to make predictions on new data. The final prediction is a dot product of the solution vector $\mathbf{v}$ of the overdetermined system equation and the vector of current prices on the chosen exchanges. More succinctly:

- For each exchange and for each smoothing method, we define a possible upper and lower range for the method's parameters and specify a step size.
- We then compute the cartesian product of all these sets, yielding all viable optimized parameter combinations in the form of a multidimensional grid.
- For each combination in this cartesian product, we:
  - Smooth the data using the methods described above.
  - Train the model and calculate the optimal solution vector, which tells us how much weight should each exchange hold
  - Evaluate its accuracy against the test data.
  - Identify the model configuration that delivers the best performance.

The overdetermined system was chosen due to a number of different factors. We preferred a simple model with the potential for an explanation or at least the possibility of quick access to information in which input parameters offer greater prediction power. Although not included in our numerical utility function, delegation and the social aspect of goodness of price are important for multiple reasons. Being less good, but providing reasonable prices attracts more delegations and provides more security and trust in the network. Therefore, the error of not predicting the price fully correctly versus being off by a lot due to an edge condition or overfitting a specific input parameter was much preferred. Furthermore, incoming network upgrades might force the providers to buy or sell assets on the price revealed (and not on market price) and this means that a large deviation from the correct price would also be financially problematic.

Lastly, the providers work in *bursts*. Most of the information-rich exchange data comes in just before the end of the epoch (last few seconds), so a longer evaluation time might mean we miss some information or be too late for the submission. Our internal analysis shows, that submission must be calculated at least 8-5 seconds before the end of each epoch to be reliably accepted by the network validators. (network latency usually requires a submission of the price a few seconds before the end of the epoch).

5 RESULT ANALYSIS

We evaluated the performance of our trained models by comparing them against three simpler prediction methods:

- The "Last Seen Value" Method: This approach predicts that the future value of a coin will be the same as its most recent price, determined by the FTSO protocol of that coin, observed before the prediction starts.
- The "Previous Epoch Value" Method: This method predicts the price of a coin for the current epoch based on the price of that same coin, determined by the FTSO protocol, from the previous epoch.
- Training an overdetermined system without smoothing the data.

Our calculation accuracy analysis spanned over a week, with new models trained every day on the previous 8-hour data (160 epochs). Following this, the model's success rate was then validated against the subsequent 8-hour dataset right after the training data. The success rate is the amount of times the predicted price would be in the interquartile range divided by the number of epochs the price was submitted for. This exactly corresponds to what price providers are financially incentivized to do.

The detailed results are presented in Figures 2 to 5. As anticipated, the **Last Seen Value Method** method yields modest outcomes, averaging averaging prediction success rate of 3.5% across all coins.

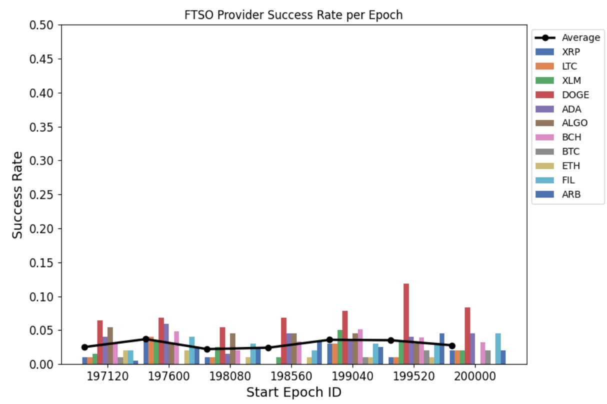

Figure 2: "Last Seen Value" prediction results

For the **Previous Epoch Value Method** approach, we set the prediction to match the price from the previous epoch. While this method outperformed the first, it still registered a low performance, averaging around 7% for all coins over the week. Notably, several coins like *ETH* or *FIL* had an average success rate close to 0%, while *DOGE* achieved an average of 15%. This goes to show that there's no one-size-fits-all approach when it comes to FTSO price predictions.

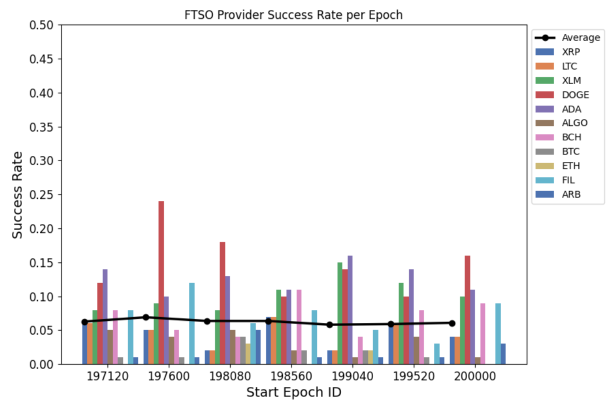

Figure 3: "Previous Epoch Value" prediction results

The method **Training an Overdetermined System Without Smoothing the Data** outperformed the first two, averaging around 10% success rate accross all coins during the testing week. Notably, the full prediction method that **Smooths the Data and Trains and Overdetermined System** outperformed all of the previous methods.

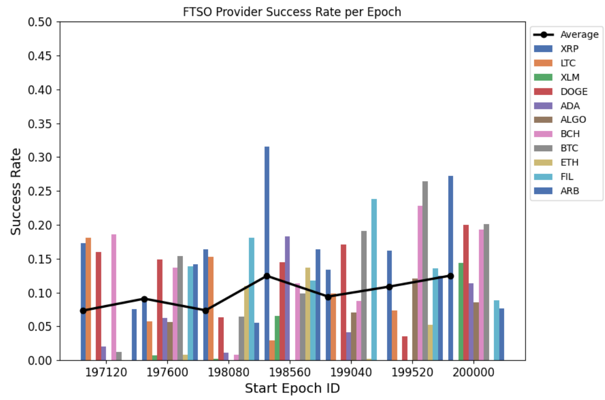

Figure 4: Overdeterminded system without data smoothing prediction results

The evaluation closely mirrored real-world conditions, due to changes in exchanges, fluctuations in vote powers, and inclusion of new data providers in the median calculation, models must be continuously retrained on an almost daily basis. Over the observed epochs, our FTSO provider demonstrated varied success rates across different cryptocurrencies. The success rates for *XRP*, *DOGE* and *BTC* generally ranged between 0.20 to 0.45, indicating moderate to high prediction accuracy. Meanwhile, coins like *XLM*, *ADA*, and *ARB* had lower success rates, often below 0.15, suggesting challenges in predicting their prices. Overall, the provider's performance fluctuated across epochs and coins, with some cryptocurrencies consistently achieving higher success rates than others. Overall, we were able to achieve moderate prediction success of around 0.22, currently ranking 26th among the 94 active FTSO providers.

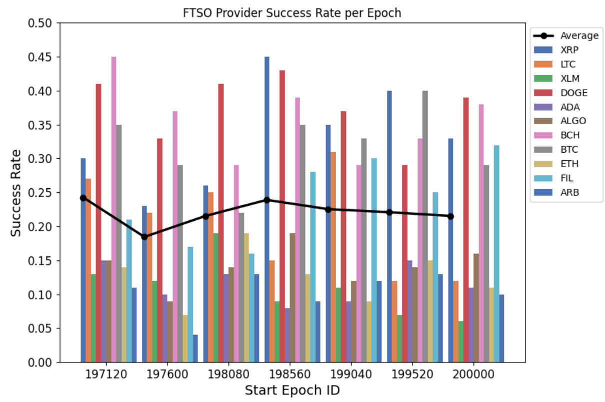

Figure 5: Overdeterminded system without with data smoothing prediction results

Because this method of smoothing and training an overdetermined system yielded better results than previous method of just training an overdetermined system, we can also be certain that smoothings in this case improve the result. This goes to show that without smoothing, our prediction model is highly influenced by noise and short-term fluctuations, making it challenging to capture the underlying trend in the time series data.

Over the observed epochs, our FTSO provider demonstrated varied success rates across different cryptocurrencies. The success rates for *XRP*, *DOGE* and *BTC* generally ranged between 0.20 to 0.45, indicating moderate to high prediction accuracy. Meanwhile, coins like *XLM*, *ADA*, and *ARB* had lower success rates, often below 0.15, suggesting challenges in predicting their prices. Overall, the provider's performance fluctuated across epochs and coins, with some cryptocurrencies consistently achieving higher success rates than others. Overall, we were able to achieve moderate prediction success of around 0.22, currently ranking 26th among the 94 active FTSO providers.

Because this method of smoothing and training an overdetermined system yielded better results than previous method of just training an overdetermined system, we can also be certain that smoothings in this case improve the result. This goes to show that without smoothing, our prediction model is highly influenced by noise and short-term fluctuations, making it challenging to capture the underlying trend in the time series data.

6 RMSE VALUES

Lastly, analyzed for each method and for each coin what is it's RSME (root mean squared error) to provide more insight into each method's accuracy. The results are depicted in 1. It's worth mentioning that since the prices of different coins vary, the RMSE values aren't comparable across the coins but only across the methods for one coin. For most coins, the *Last Seen Value* method generally yields the highest RMSE values, indicating the worst accuracy relative to other methods. Conversely, the *Overdetermined system with smoothing* method tends to produce the lowest RMSE values for most of the coins. The methods *Previous Epoch Value* and *Overdetermined system without smoothing* are ranked somewhere in between.

7 DISCUSSION AND FUTURE WORK

We have developed and assessed a functional provider solution to predict prices within the FTSO protocol. While we observed commendable performance for coins such as *XRP*, *DOGE*, and *BTC*, the results for other coins like *XLM*, *ADA*, and *ARB* were not as promising. Exploring additional smoothing techniques and incorporating multiple prediction methods would be beneficial. Notably, ensemble methods are renowned for reducing prediction variance, which in turn increases the probability of predictions falling within the median target range.

This paper has only focused on non-deep learning approaches to FTSO price prediction. A promising extension to the provider would be to explore time series prediction using various deep learning methods such as RNN or LSTM neural networks. These models have the potential to capture more subtle patterns in the data and adapt to the dynamic prices of the crypto coins. They might need to be modified to adapt to the specifics of the FTSO system and quick retraining times. Combining the more expensive inference of neural networks with presented overdetermined system together with error bounds on prediction results might also offer a more performant composite algorithm that would be able to use the fallback prediction in case of lateness of prediction by a stronger but more complicated model.

8 ACKNOWLEDGMENTS

The authors would like to thank AFLabs for the provision of exchange and FTSO data to be used during the development phase.

Coin	Last Seen	Prev. Ep	No smoth	Smooth
XRP	0.07412964	0.01536945	0.00542317	0.00398449
LTC	0.07412961	0.01536940	0.00735026	0.00401269
XLM	0.00010802	0.00025230	0.00090994	0.00025548
DOGE	0.00004626	0.00001359	0.00000733	0.00000641
ADA	0.00000201	0.00000395	0.00000183	0.00000174
ALGO	0.00011186	0.00000559	0.00000351	0.00000379
BCH	1.47382928	0.00013239	0.00000828	0.00000565
BTC	23.78687273	5.01065648	1.94068887	0.91171693
ETH	1.50008731	0.54618855	0.18091784	0.05930725
FIL	0.00360921	0.00079709	0.00039865	0.00040482
ARB	0.00098386	0.00025156	0.00015229	0.00014042

Table 1: RMSE for every method and coin

Coin	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
XRP	0.019	0.047	0.011	0.000	0.034	0.016	0.022
LTC	0.015	0.049	0.017	0.000	0.037	0.011	0.023
XLM	0.015	0.035	0.025	0.011	0.052	0.035	0.029
DOGE	0.064	0.069	0.054	0.069	0.079	0.119	0.084
ADA	0.049	0.059	0.015	0.045	0.035	0.042	0.045
ALGO	0.054	0.034	0.045	0.045	0.045	0.038	0.000
BCH	0.031	0.048	0.025	0.033	0.051	0.039	0.032
BTC	0.012	0.000	0.000	0.000	0.014	0.025	0.022
ETH	0.021	0.029	0.014	0.011	0.011	0.013	0.000
FIL	0.029	0.048	0.033	0.026	0.036	0.038	0.045
ARB	0.005	0.025	0.025	0.035	0.025	0.045	0.029

Table 2: The last seen value method

Coin	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
XRP	0.063	0.054	0.021	0.078	0.026	0.060	0.047
LTC	0.065	0.051	0.024	0.074	0.026	0.062	0.044
XLM	0.085	0.096	0.088	0.110	0.151	0.127	0.161
DOGE	0.125	0.249	0.187	0.154	0.146	0.167	0.161
ADA	0.142	0.174	0.137	0.114	0.164	0.149	0.115
ALGO	0.050	0.048	0.051	0.028	0.019	0.044	0.019
BCH	0.082	0.053	0.043	0.110	0.042	0.081	0.094
BTC	0.015	0.013	0.041	0.025	0.029	0.013	0.000
ETH	0.000	0.000	0.039	0.000	0.024	0.000	0.000
FIL	0.084	0.120	0.067	0.082	0.057	0.034	0.093
ARB	0.015	0.016	0.054	0.012	0.014	0.014	0.039

Table 3: The previous epoch value method

Coin	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
XRP	0.173	0.090	0.164	0.316	0.134	0.162	0.272
LTC	0.181	0.057	0.153	0.029	0.099	0.074	0.000
XLM	0.000	0.007	0.002	0.066	0.000	0.000	0.144
DOGE	0.160	0.149	0.063	0.145	0.171	0.035	0.200
ADA	0.020	0.062	0.011	0.183	0.041	0.000	0.114
ALGO	0.000	0.056	0.000	0.000	0.071	0.121	0.086
BCH	0.186	0.137	0.008	0.114	0.088	0.228	0.193
BTC	0.012	0.154	0.064	0.099	0.191	0.264	0.201
ETH	0.000	0.008	0.110	0.137	0.002	0.052	0.000
FIL	0.000	0.139	0.181	0.118	0.238	0.136	0.089
ARB	0.076	0.142	0.055	0.164	0.000	0.125	0.077

Table 4: Training an Overdetermined System Without Smoothing the Data

Coin	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
XRP	0.311	0.235	0.261	0.452	0.356	0.420	0.338
LTC	0.277	0.221	0.254	0.153	0.317	0.126	0.125
XLM	0.130	0.122	0.192	0.096	0.115	0.076	0.062
DOGE	0.413	0.335	0.412	0.438	0.376	0.296	0.396
ADA	0.158	0.198	0.138	0.080	0.099	0.157	0.112
ALGO	0.154	0.092	0.148	0.191	0.128	0.144	0.169
BCH	0.453	0.374	0.293	0.399	0.290	0.334	0.386
BTC	0.356	0.297	0.220	0.357	0.330	0.423	0.295
ETH	0.149	0.075	0.194	0.137	0.093	0.155	0.115
FIL	0.219	0.171	0.168	0.287	0.366	0.250	0.327
ARB	0.110	0.043	0.135	0.090	0.127	0.138	0.164

Table 5: Smoothing the data and training an overdetermined system

REFERENCES

[1] AlaghaA. "A reinforcement learning model for the reliability of blockchain oracles". In: *ScienceDirect* (2022). URL: https://www.sciencedirect.com/science/article/abs/pii/S0957417422021789.

[2] Ken Binmore. *Game Theory: A Very Short Introduction.* Oxford University Press, 2021. URL: https://books.google.si/books?hl=en&lr=&id=3KnuDwAAQBAJ&oi=fnd&pg=PT11&dq=game+theory&ots=mAN9J3v9-Q&sig=hvEPGmrg-JArHLvWIoNuoU5bxDE&redir_esc=y#v=onepage&q=game%20theory&f=false.

[3] Giulio Caldarelli. "Overview of Blockchain Oracle Research". In: *MDPI* 14.6 (2022), p. 175. URL: https://www.mdpi.com/1999-5903/14/6/175.

[4] Giulio Caldarelli. "Understanding the Blockchain Oracle Problem: A Call for Action". In: *Information* 11.11 (2020), p. 509. URL: https://www.mdpi.com/2078-2489/11/11/509.

[5] Giulio Caldarelli. "Understanding the Blockchain Oracle Problem: A Call for Action". In: 11.11 (2023), p. 509. URL: https://www.mdpi.com/2078-2489/11/11/509.

[6] Chainlink. *Chainlink 2.0 and the future of Decentralized Oracle Networks.* Accessed: 2023-09-05. 2023. URL: https://chain.link/whitepaper.

[7] Vasant Dhar. "Data Science and Prediction". In: *Communications of the ACM* 56.12 (2013), pp. 64–73. URL: https://dl.acm.org/doi/abs/10.1145/2500499.

[8] Joshua Ellul. "The Blockchain Oracle Problem in Decentralized Finance—A Multivocal Approach". In: 11.16 (2023), p. 7572. URL: https://www.mdpi.com/2076-3417/11/16/7572.

[9] Boi Faltings and Goran Radanovic. *Game Theory for Data Science: Eliciting Truthful Information.* Springer Nature, 2021. URL: https://books.google.si/books?hl=en&lr=&id=wYRyEAAAQBAJ&oi=fnd&pg=PP1&dq=game+theory+data+science&ots=mf1CNM9aP2&sig=2pJ_p-QB_NUeDm_Z23usBQSFw3M&redir_esc=y#v=onepage&q=game%20theory%20data%20science&f=false.

[10] James D. Hamilton. *Time Series Analysis.* Princeton University Press, 1994. URL: http://mayoral.iae-csic.org/timeseries2021/hamilton.pdf.

[11] A. J. Lawrance and P. A. W. Lewis. "An exponential moving-average sequence and point process (EMA1)". In: *Journal of Applied Probability* 14 (1 1977). Accessed: 2023-09-05, pp. 98–113. DOI: 10.2307/3213263. URL: https://www.cambridge.org/core/journals/journal-of-applied-probability/article/abs/an-exponential-movingaverage-sequence-and-point-process-ema1/7CFB5DE9313286DAB7C6EF3D540D6A29.

[12] Christopher Potts. "Interpretive Economy". In: *Semantics Archive* (2008). Accessed: 2023-09-05. URL: https://semanticsarchive.net/Archive/jExYWZlN/potts-interpretive-economy-mar08.pdf.

[13] William H. Press and Saul A. Teukolsky. "Savitzky-Golay Smoothing Filters". In: *Computers in Physics and IEEE Computational Science & Engineering* 4.6 (1990). Accessed: 2023-09-05, pp. 669–672. DOI: 10.1063/1.4822961. URL: https://pubs.aip.org/aip/cip/article-abstract/4/6/669/510032/Savitzky-Golay-Smoothing-Filters?redirectedFrom=PDF.

[14] Fan Zhang et al. "Decentralized Oracles: a Comprehensive Overview". In: *arXiv preprint arXiv:2004.07140* (2020). Accessed: 2023-09-05. URL: https://arxiv.org/abs/2004.07140.

[15] Yanzhao Zhang Zhiqiang Liu and Jiwu Jing. "Connect API with Blockchain: A Survey on Blockchain Oracle Implementation". In: *ACM* (2022). URL: https://dl.acm.org/doi/abs/10.1145/3567582.

[16] Aviv Zohar and Ittay Eyal. "A Study of Blockchain Oracles". In: *arXiv* (2020). URL: https://arxiv.org/pdf/2004.07140.