

Article

Not peer-reviewed version

Neural Network-Informed Lotka-Volterra Dynamics for Cryptocurrency Market Analysis

[Dimitris Kastoris](#) , [Dimitris Papadopoulos](#) ^{*} , [Konstantinos Giotopoulos](#)

Posted Date: 27 June 2025

doi: 10.20944/preprints202506.2227.v1

Keywords: Lotka-Volterra; Deep Neural Network; MATLAB; Runge-Kutta method; Cryptocurrency; Forecasting





Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Neural Network-Informed Lotka-Volterra Dynamics for Cryptocurrency Market Analysis

Dimitris Kastoris ^{1,†}, Dimitris Papadopoulos ^{1,†,*}  and Kostas Giotopoulos ^{1,†} 

Department of Management Science and Technology, University of Patras, 26504 Patras, Greece

* Correspondence: dimfpap@upatras.gr

† These authors contributed equally to this work.

Abstract: Mathematical modeling plays a crucial role in supporting decision-making across a wide range of scientific disciplines. These models often involve multiple parameters, the estimation of which is critical to assessing their reliability and predictive power. Recent advancements in artificial intelligence have made it possible to efficiently estimate such parameters with high accuracy. In this study, we focus on modeling the dynamics of cryptocurrency market shares by employing a Lotka-Volterra system. We introduce a methodology based on a deep neural network (DNN) to estimate the parameters of the Lotka-Volterra model, which are subsequently used to numerically solve the system using a fourth-order Runge-Kutta method. The proposed approach, when applied to real-world market share data for Bitcoin, Ethereum, and alternative cryptocurrencies, demonstrates excellent alignment with empirical observations. Moreover, our method outperforms ARIMA models in terms of accuracy, showcasing its effectiveness for crypto market forecasting. The entire framework, including neural network training and Runge-Kutta integration, was implemented in MATLAB.

Keywords: Lotka-Volterra; deep neural network; MATLAB; Runge-Kutta method; cryptocurrency; forecasting

1. Introduction

Over the past decade, blockchain technology has rapidly emerged as a transformative force, with many experts predicting it could reshape various aspects of modern society [1]. At its core, a blockchain is a decentralized, publicly accessible digital ledger [2]. While it is most commonly associated with cryptocurrencies-digital assets designed to function as a medium of exchange without the need for centralized oversight-it has far-reaching applications beyond that.Bitcoin, introduced in 2008 [2] under the pseudonym Satoshi Nakamoto (whose identity remains unknown and may represent either an individual or a group), stands as the pioneer and most widely recognized cryptocurrency. By December 2024, Bitcoin’s market capitalization soared to around 1.8 trillion euros [3], and it continues to maintain a strong market presence, currently valued above 1.4 trillion euros.The revolutionary concept introduced by Nakamoto not only sparked a dramatic surge in Bitcoin’s financial value but also inspired the creation of thousands of alternative cryptocurrencies. Today, over 17,000 digital currencies exist, either directly based on Bitcoin’s original blockchain or built on modified versions. This number continues to grow steadily. Beyond cryptocurrencies, blockchain technology has found applications across various sectors [4], with Decentralized Finance (DeFi) emerging as one of the most prominent.

A significant part of what has driven the success of cryptocurrencies is their strong focus on security and user privacy-features [5] that traditional financial systems often struggle to match. By utilizing advanced cryptographic protocols, digital currencies ensure the integrity of transactions and protect the anonymity of users. Additionally, the decentralized architecture behind most cryptocur-rencies removes the need for central governing bodies [6], which not only reduces transaction costs but also improves efficiency and accessibility. Ethereum, one of the most prominent cryptocurrencies, has taken these innovations even further. It introduced smart contracts-self-executing agreements

[6] coded directly onto the blockchain. These contracts automatically enforce their terms, offering greater transparency and eliminating the need for intermediaries. This breakthrough has significantly expanded the use cases for blockchain technology, pushing it beyond simple peer-to-peer payments into areas like decentralized finance (DeFi), gaming, and various Web3 applications. As of April 8, 2025, Ethereum's market capitalization stands at approximately 167 billion euros [7]. This marks a significant decline compared to December 2024, when its market cap was valued at around 401 billion euros. The distinct benefits of blockchain—particularly its security, transparency, and decentralized control—continue to fuel the rise of cryptocurrencies like Ethereum. As the technology matures and adoption spreads, it's likely to play an even more critical role in shaping the future of finance. Given Ethereum's open-source nature, decentralized framework, and history of high returns, it has become a prime focus for investors and researchers alike, sparking growing interest in forecasting its price dynamics and understanding its long-term potential.

Cryptocurrencies have steadily gained prominence and are now playing an increasingly significant role in the global financial landscape. At their peak, the total market capitalization of digital assets approached 3 trillion euros, rivaling the valuations of some of the world's largest publicly traded companies. What once began as an experimental domain for a niche group of tech-savvy enthusiasts has evolved into a mainstream asset class. The launch of Bitcoin futures in 2017 marked a turning point, and the approval of spot Bitcoin ETFs in January 2024 further solidified crypto's presence in traditional financial markets. Initially driven by retail interest, the space has since attracted major institutional players [8], with many more expected to follow suit.

That said, the rise of crypto has not been without skepticism. Several traditional financial institutions remain wary, viewing cryptocurrencies as speculative bubbles rather than genuine innovations. Common criticisms include extreme price fluctuations, substantial energy demands, and relatively slow transaction speeds that hinder scalability and liquidity [9]. Some argue that neither Bitcoin nor blockchain has yet offered clear solutions to significant real-world problems. Nonetheless, regardless of whether these digital assets ultimately live up to their revolutionary promise, their current influence in the financial world is undeniable. As of today, the total market capitalization of all cryptocurrencies hovers around 3 trillion euros. A 2024 survey reported approximately 560 million active cryptocurrency users [10], highlighting the scale and reach of this emerging ecosystem. Unsurprisingly, the scientific community has taken a growing interest in cryptocurrencies, particularly through the lenses of complex systems and network theory. A considerable body of work has focused specifically on Bitcoin—examining its transactional network [11,12], price dynamics, and forecasting models [13,14]. However, only a limited number of studies have addressed the broader crypto market as a whole or attempted to model its complex behavior at the systemic level [15,16].

This study explores the behavior of cryptocurrency market shares by applying the Lotka-Volterra model to Bitcoin (BTC), Ethereum (ETH), and a combined group of alternative coins (ALTs). We compare BTC, ETH, and ALTs because they represent the three most significant and distinct segments of the cryptocurrency market. BTC (Bitcoin) is the original and dominant cryptocurrency, ETH (Ethereum) is the leading smart contract platform, and ALTs (altcoins) collectively capture all other cryptocurrencies, reflecting broader market trends and diversification. Given the dominance of a few major cryptocurrencies in total market capitalization, significant barriers to new entrants, and interdependent strategic behavior, the cryptocurrency market exhibits key characteristics of an oligopolistic structure. This justifies the application of competitive dynamics models, such as the Lotka-Volterra framework, to analyze the evolution of market shares and concentration. To determine the nonlinear interaction parameters within the system, we integrate a neural network trained on actual market data. This combined method effectively captures the competitive and cooperative dynamics between major digital assets. The estimated parameters from the neural network are then used to simulate the system's evolution over time. Overall, the approach offers valuable insights into the structural behavior and shifts within the crypto market. The primary contribution of this study lies in analyzing the evolution of market structure and concentration by utilizing concepts

from the evolutionary theory of population biology and population dynamics. Specifically, this research forecasts and models market evolution by employing the Lotka–Volterra framework, which is traditionally used to describe the competitive interactions between species vying for a shared, finite resource [17]. The primary objective of the proposed methodology is to offer an alternative approach for estimating the concentration level of a market with significant entry barriers, such as the cryptocurrency market. The Lotka–Volterra model utilized in this study has been applied successfully in various fields beyond biology, providing reliable estimates of the dynamics [18] within the processes it describes. Furthermore, the methodology proposed here can be integrated with existing frameworks or even used as a benchmark to validate their evaluation outcomes in the context of cryptocurrency markets.

Achieving the objectives outlined above would make significant contributions to both academic research and practical applications. It would offer novel approaches for estimating market concentration and provide an additional tool for strategic planning. When integrated with the recommendations presented in the conclusion, these efforts would lead to the creation of a comprehensive framework capable of capturing the various elements and factors that drive diffusion processes, particularly in the context of market competition and concentration levels.

2. Dynamics of Interacting Populations Under Competition

The hypothesis regarding population variation suggests that the rate of change is directly proportional to the current population size. The most widely accepted method for modeling species population growth in the absence of competition [19,20] is represented by the following equation:

$$\frac{dN(t)}{dt} = rN(t) \left(1 - \frac{N(t)}{K} \right) \quad (1)$$

Here, $N(t)$ denotes the population size at time t , r represents the intrinsic growth rate, and K is the environmental carrying capacity—the maximum population the environment can sustain. This logistic model forms the basis of many demand estimation and forecasting techniques, including the logistic growth family [21] and Gompertz models [22].

When multiple species coexist in a shared ecosystem, they inevitably compete for limited resources. Definitions and explanations of such interspecies competition are found in [19] and [23], and can be summarized as follows: "Competition arises when two or more species experience diminished fitness-reduced growth or saturation levels-due to their simultaneous presence."

In such a scenario, each species' access to resources diminishes due to the presence of others, resulting in decreased growth rates and reduced carrying capacities. More specific types of species [17] interaction have been categorized as:

- **Predator-prey:** One population's growth decreases while the other's increases.
- **Competition:** Growth rates of both populations decrease.
- **Mutualism (symbiosis):** Growth rates of both populations increase.

In the context of a closed and established cryptocurrency market, firms interact similarly: competition over limited market share can reduce the share of each participant, especially if they aim to maximize both market dominance and profits. This aligns with the second interaction type (competition), making it the most suitable analogy.

The simplest way to reflect this competition in a growth model is by introducing terms that quantify mutual interference. The resulting formulation is the Lotka–Volterra model, developed by Lotka and Volterra. This model has been thoroughly explored in the literature [17,20,23] for scenarios involving two or more interacting species.

For m competing participants (or entities) in a market [24,25], the system dynamics of Equation (1) can be described by the following set of first-order nonlinear differential equations:

$$\frac{dN_i}{dt} = N_i \left(a_i - \sum_{j=1}^m a_{ij} N_j \right), \quad i = 1, 2, \dots, m \quad (2)$$

In this expression, $\frac{dN_i}{dt}$ represents the rate of change of the i -th entity, a_i is its individual growth coefficient, and a_{ij} measures the competitive effect of species j on species i (interspecies when $i \neq j$, intraspecies when $i = j$). Note that these coefficients generally differ.

Each equation in this system can be derived by transforming the logistic growth equation as follows:

$$\frac{dN(t)}{dt} = rN(t) \left(1 - \frac{N(t)}{K} \right) = N(t) \left(r - \frac{r}{K} N(t) \right) = N(t)(r - aN(t)) \quad (3)$$

Additional terms are then included to represent the competitive interactions among entities in the market.

For the case where $m = 3$, the generalized Lotka–Volterra system described in Equation (2) is expressed as:

$$\begin{aligned} \frac{dx}{dt} &= x(a_{10} + a_{11}x + a_{12}y + a_{13}z) \\ \frac{dy}{dt} &= y(a_{20} + a_{21}x + a_{22}y + a_{23}z) \\ \frac{dz}{dt} &= z(a_{30} + a_{31}x + a_{32}y + a_{33}z) \end{aligned} \quad (4)$$

Here x, y , and z represent the three competing entities. The Lotka–Volterra formulation is particularly well-suited to capturing the nonlinear competitive dynamics among firms in a market. Its ability to incorporate interaction terms directly into the growth equations allows for more realistic modeling [26] of interdependencies, which are often nonlinear and time-varying. In contrast, traditional techniques such as linear regression are limited by their assumption of fixed relationships and fail to encapsulate these dynamic competitive effects. While agent-based models offer a more detailed view by simulating individual behaviors, they require significant computational power and detailed datasets, which may not always be feasible. Likewise, equilibrium-based models tend to ignore temporal changes, assuming static market conditions.

For these reasons, the Lotka–Volterra system stands out as a balanced and computationally efficient option that can both describe and predict the evolving nature of market competition. Similar modeling strategies have previously been applied to other sectors with promising results [27,28]. This system captures competition at a macro level and includes external influences—such as marketing variables—only implicitly. The key assumption is that such external factors remain constant throughout the study period. These factors may include global economic conditions (e.g., inflation, interest rates, recession fears), regulatory changes, investor sentiment, technological upgrades, and institutional adoption, all of which can significantly affect market dynamics. While these are not directly analyzed in this paper, their future incorporation would enable a more holistic model. In this paper, we develop a hybrid neural-network–Lotka–Volterra model as a generalized framework for forecasting cryptocurrency market dynamics and characterizing the competitive interactions and equilibrium among multiple assets. Future work will integrate exogenous drivers to capture both direct and indirect environmental influences.

3. Deep Neural Network Model for Crypto Dynamics

3.1. Feed Forward Neural Network Architecture

A feed forward neural network (NN) with three hidden layers as on Figure 1 was constructed to learn the temporal dynamics of the system. The architecture is described as follows:

- **Input Layer:** Accepts a single input feature corresponding to time, denoted as t .
- **Hidden Layers:** Three fully connected hidden layers, whith 10,6 and 9 neurons each and using the hyperbolic tangent (\tanh) activation function. To promote stable learning, weights were initialized to small values and biases were set to zero.
- **Output Layer:** A fully connected layer with three outputs, representing the estimated values of the variables $x(t)$, $y(t)$, and $z(t)$. Each output is connected to a regression layer, which calculates the loss based on the deviation between predicted and actual values.

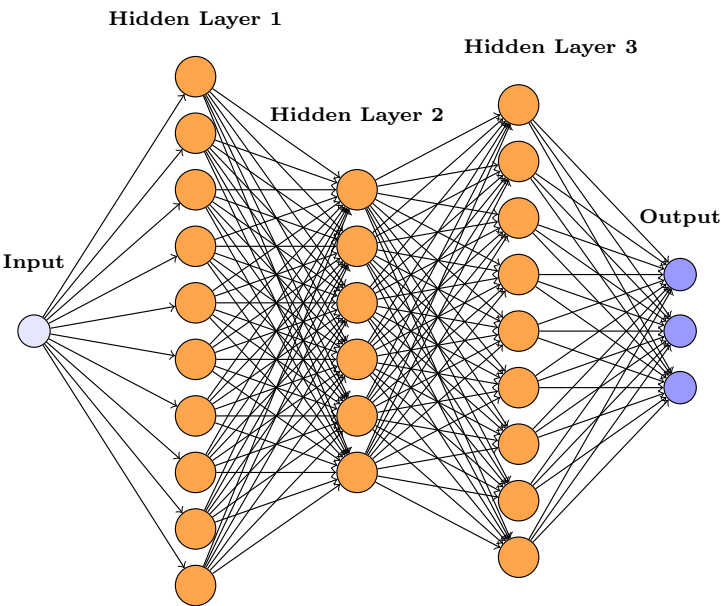


Figure 1. An example of a deep neural network with one input neuron, three output neurons and three hidden layers, with 10,6 and 9 neurons.

Function of the Neural Network:

1. **Time-Series Prediction:** The neural network is trained on datasets (x, y, z) to learn and predict their evolution over time. Once trained, it produces predictions for the three variables based solely on time input.
2. **Modeling Underlying Relationships:** By learning from data, the NN approximates the hidden relationships between the variables and time, capturing complex and nonlinear behaviors that traditional models might overlook.
3. **Preparing for Optimization:** After prediction, the outputs obtain continuous estimates of the state variables. These are then used to compute numerical derivatives $\frac{dx}{dt}$, $\frac{dy}{dt}$, and $\frac{dz}{dt}$. These derivatives are essential for the next phase of analysis, where model parameters are estimated via optimization using `lsqnonlin`.

In essence, the neural network plays a crucial role in approximating the dynamics of the system, bridging the gap between observed data and the differential equations required for parameter identification. The neural network architecture with three hidden layers consisting of 10, 6, and 9 neurons respectively was selected based on bootstrap validation performance. Using moving block bootstrap resampling to simulate multiple training and testing scenarios, we compared several candidate archi-

tructures. The 10-6-9 configuration consistently achieved the lowest average test error and exhibited minimal performance variance across bootstrap samples, indicating strong generalization capability and robustness to data perturbations. The network's depth allowed it to model nonlinear interactions among crypto market variables, while its compact structure minimized overfitting.

3.2. Feed-Forward Propagation

A feedforward neural network processes data points through a sequence of operations, as outlined in [29]. Given an input data point $t^{(m)}$, the forward propagation begins by initializing the first layer activation as $r_1^{[0]} = t^{(m)}$. To obtain the activations of layer l , denoted \mathbf{r}^l , from those of the previous layer \mathbf{r}^{l-1} , two main steps are followed.

First, an intermediate vector \mathbf{c}^l is calculated by multiplying the weight matrix W^l with \mathbf{r}^{l-1} , then adding the corresponding bias vector \mathbf{b}^l :

$$c_j^l = \sum_{k=1}^{h^{l-1}} W_{jk}^l r_k^{l-1} + b_j^l. \quad (5)$$

This operation can be compactly written in matrix form as:

$$\mathbf{c}^l = W^l \mathbf{r}^{l-1} + \mathbf{b}^l. \quad (6)$$

The second step applies the activation function σ^l element-wise to \mathbf{c}^l , producing the output activations \mathbf{r}^l :

$$\mathbf{r}^l = \sigma^l(\mathbf{c}^l) = \sigma^l(W^l \mathbf{r}^{l-1} + \mathbf{b}^l). \quad (7)$$

Thus, for each input $t^{(m)}$, both the pre-activation values \mathbf{c}^l and the activated outputs \mathbf{r}^l are computed using Equations (6) and (7). To highlight the dependence on the input sample $t^{(m)}$, we can denote these layer-wise computations as $\mathbf{r}^{l(m)} = \sigma^l(\mathbf{c}^{l(m)})$. In matrix notation, the forward propagation for the entire dataset is summarized as:

$$\begin{aligned} T &= [t_1, \dots, t_N], \\ \mathbf{c}^l &= [\mathbf{c}^{l(1)}, \dots, \mathbf{c}^{l(N)}], \\ \mathbf{A}^l &= [\mathbf{r}^{l(1)}, \dots, \mathbf{r}^{l(N)}]. \end{aligned} \quad (8)$$

This leads to a recursive formulation for the forward pass:

$$\begin{aligned} \mathbf{A}^0 &= T, \\ \mathbf{c}^l &= W^l \mathbf{A}^{l-1} + \mathbf{b}^l, \\ \mathbf{A}^l &= \sigma^l(\mathbf{c}^l), \quad \text{for } l = 1, \dots, L. \end{aligned} \quad (9)$$

The final network output, represented by \mathbf{A}^L , is a function of the input and the learned parameters:

$$N(T, W^L, \mathbf{b}^L) = \mathbf{A}^L. \quad (10)$$

Alternatively, we may express the output in component form as:

$$N_j(t^{(m)}, W^L, \mathbf{b}^L) = \pi_j(\mathbf{A}^L) = A_j^L(t^{(m)}, W, \mathbf{b}), \quad (11)$$

where π_j in (11) denotes the extraction of the j -th component from the final output vector \mathbf{A}^L .

4. Hybrid Neural Network–ODE Framework for Parameter Estimation and Forecasting

This section outlines the methodology developed for estimating parameters in dynamic systems governed by nonlinear ordinary differential equations (ODEs). Our approach utilizes a feedforward neural network in combination with optimization techniques to infer the unknown parameters. To demonstrate its practical utility, we apply the method to a Lotka-Volterra model describing the interactions among three major segments of the cryptocurrency market: Bitcoin, Ethereum, and altcoins. Time-series data for each segment is used, and the results highlight the model’s ability to accurately capture the system’s dynamic behavior.

Accurate parameter estimation in nonlinear dynamic systems is a key challenge across various domains, including financial forecasting. Traditional techniques often rely on simplifications such as linearization or require detailed prior knowledge of the system’s internal structure-assumptions that may not hold in complex financial ecosystems. In this work, we introduce a data-driven approach that leverages the flexibility of neural networks to model intricate dependencies, while employing numerical optimization to estimate the parameters of a nonlinear model. To provide clarity on the implementation, we also present the core steps of the algorithm used in our framework.

Algorithm 1 Proposed Method

- 1: **Input:** Crypto market share data from Excel file, time vector $t_data = [0, 1, \dots, N - 1]^T$
 - 2: Read and preprocess data $[BTC, ETH, ALT]$ into $data$
 - 3: Split data into 70% training ($train_data, train_time$) and 30% testing ($test_data, test_time$)
 - 4: Define a Feedforward Neural Network with 3 hidden layers (tanh activation) and linear output
 - 5: Initialize best weights and minimum validation cost
 - 6: **for** each bootstrap iteration $b = 1$ to 5 **do**
 - 7: Generate moving block bootstrap indices (block length = 10) for training data
 - 8: Select resampled training data $train_data^{(b)}, train_time^{(b)}$
 - 9: Train DNN to predict \dot{U} , i.e., the derivatives of market shares ($\dot{U} = \frac{dU}{dt}$), on $train_time^{(b)}, train_data^{(b)}$
 - 10: Integrate \dot{U}_{pred} to obtain U_{pred} , i.e., reconstruct predicted market shares from predicted derivatives
 - 11: Evaluate on out-of-bag (OOB) data to compute validation cost
 - 12: **if** OOB cost is minimum **then**
 - 13: Store best weights
 - 14: **end if**
 - 15: **end for**
 - 16: Use best weights to compute U_{pred} and derivatives $\frac{du_1}{dt}, \frac{du_2}{dt}, \frac{du_3}{dt}$ analytically
 - 17: Define ODE system: $\dot{u}_i = u_i \cdot (p_{i1} + p_{i2}u_1 + p_{i3}u_2 + p_{i4}u_3), i = 1, 2, 3$
 - 18: Use nonlinear least squares to estimate parameters p_1 to p_{12}
 - 19: Print estimated parameters
 - 20: Numerical solution fo the Lotka-Volterra equations over the $test_time$ interval using the Runge-Kutta method
-

4.1. Mathematical Formulation

As we see in the Algorithm 1 the neural network is trained to approximate the time evolution of the system by learning the mapping from time to the observed data [30]. After predicting the trajectories, we numerically compute the derivatives and fit them to a generalized Lotka–Volterra

(GLV) model using nonlinear least squares optimization. The system of ODEs used to model the interactions among the three components (e.g., BTC, ETH, and ALTs) is given by:

$$\begin{aligned}\frac{du_1}{dt} &= u_1(p_1 + p_2u_1 + p_3u_2 + p_4u_3), \\ \frac{du_2}{dt} &= u_2(p_5 + p_6u_1 + p_7u_2 + p_8u_3), \\ \frac{du_3}{dt} &= u_3(p_9 + p_{10}u_1 + p_{11}u_2 + p_{12}u_3).\end{aligned}\tag{12}$$

To estimate the parameters p_1, \dots, p_{12} , we define a residual function that quantifies the mismatch between the model's predicted derivatives and the ones obtained from the neural network output. This residual function is given by:

$$r(p) = \begin{bmatrix} \frac{du_1}{dt} - u_1(p_1 + p_2u_1 + p_3u_2 + p_4u_3) \\ \frac{du_2}{dt} - u_2(p_5 + p_6u_1 + p_7u_2 + p_8u_3) \\ \frac{du_3}{dt} - u_3(p_9 + p_{10}u_1 + p_{11}u_2 + p_{12}u_3) \end{bmatrix}\tag{13}$$

In Equation (13) the derivatives are obtained from the neural network. Because the network is trained to output $u_i(t)$ only, each time derivative $\frac{du_i}{dt}$ must be computed by analytically differentiating the trained network with respect to t (using the chain rule through every hidden layer). Likewise, when forming $r(p)$ for the `lsqnonlin` call, we should state explicitly that the input " $\frac{du_i}{dt}$ " is this analytic network derivative rather than a numerical finite difference. The optimal parameters are obtained by minimizing the sum of squared residuals:

$$\min_p \|r(p)\|_2^2.\tag{14}$$

This optimization aligns the dynamics of the GLV model with the data-driven derivatives, effectively capturing the underlying interactions among the system components.

4.2. Numerical Integration via Runge–Kutta

Numerous numerical methods have been developed for solving ordinary differential equations. Among the most widely used for the integration of first- and second-order ODEs are the Runge-Kutta (-Nyström) schemes and linear multistep methods (e.g., see [31–38]).

The obtained coefficients are integrated into the Lotka-Volterra system (12) which is solved over the original time span using the classical 4th-order Runge-Kutta method [31]. The simulated trajectories are then compared to the observed data, and RMSE is computed to quantify fit quality.

$$\frac{dy}{dt} = f(t, y), \quad y(t_0) = y_0\tag{15}$$

The Runge-Kutta method approximates the solution to (15) through an iterative process defined by:

$$\begin{aligned}y_{n+1} &= y_n + h \sum_{i=1}^s b_i k_i, \\ k_i &= f\left(t_n + c_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j\right), \quad \text{for } i = 1, 2, \dots, s.\end{aligned}\tag{16}$$

where
 h represents the time step,
 s denotes the number of stages in the method,
 k_i are the intermediate stage values,
 a_{ij}, b_i, c_i are coefficients specifying the particular Runge-Kutta scheme.

From the general Runge-Kutta framework (16), various schemes of different algebraic orders can be derived. In this study, we utilize a fourth-order Runge-Kutta method with four stages [31].

Given an initial condition $y(t_0) = y_0$, the solution at the subsequent time step $t_{n+1} = t_n + h$ is calculated as:

$$\begin{aligned} k_1 &= f(t_n, y_n), \\ k_2 &= f(t_n + c_2h, y_n + (a_{21}k_1)h), \\ k_3 &= f(t_n + c_3h, y_n + (a_{31}k_1 + a_{32}k_2)h), \end{aligned} \quad (17)$$

$$\begin{aligned} k_4 &= f(t_n + c_4h, y_n + (a_{41}k_1 + a_{42}k_2 + a_{43}k_3)h), \\ y_{n+1} &= y_n + h(b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4). \end{aligned} \quad (18)$$

For the classical RK4 method, the coefficients are:

$$\begin{aligned} c_2 &= \frac{1}{2}, \quad c_3 = \frac{1}{2}, \quad c_4 = 1, \\ a_{21} &= \frac{1}{2}, \quad a_{31} = 0, \quad a_{32} = \frac{1}{2}, \quad a_{41} = 0, \quad a_{42} = 0, \quad a_{43} = 1, \\ b_1 &= \frac{1}{6}, \quad b_2 = \frac{1}{3}, \quad b_3 = \frac{1}{3}, \quad b_4 = \frac{1}{6}. \end{aligned}$$

This iterative procedure is applied at each time step, generating an approximate solution to the differential equation described by (15).

4.3. Model Evaluation

The neural network was trained using the Adam optimizer and epochs with initial learning rate were selected after experimenting with various values. To identify the optimal architecture, three different configurations of hidden layers were evaluated. In addition, multiple activation functions were tested, but the hyperbolic tangent (tanh) proved most effective, delivering the most stable and accurate results.

For performance evaluation, a moving block bootstrap validation approach was employed. This method resamples blocks of the original time series to generate multiple training and testing datasets, preserving temporal dependencies within each block. Unlike traditional cross-validation or walk-forward methods, bootstrap validation is particularly useful when dealing with limited data or non-stationary time series [39]. It allows for robust statistical inference and a more comprehensive understanding of the model's generalization performance across different subsets of the data. We chose a block length of 10 time steps to span roughly one calendar month of share data, under the assumption that correlations persist over that window. We ran 5 bootstrap replicates, which we found sufficient to stabilize the out-of-bag (OOB) cost minima in preliminary tests. For each replicate, the OOB indices consist of all training points not selected by any block; the OOB cost is then computed on those held-out points to estimate generalization error. The replicate producing the lowest OOB cost is retained for the final neural-network weights. By applying this resampling technique, we obtained a realistic and statistically grounded estimate of the network's prediction accuracy under data variability.

During training, a dynamic plot was used to monitor convergence and track loss reduction in real time. Once training was complete, the model was used to predict the time-dependent variables $x(t)$, $y(t)$, and $z(t)$ across the full time horizon. The performance of the proposed method is highly influenced by both the choice of activation function and the strategy used for initializing weights

and biases. The tanh function was chosen over sigmoid and ReLU because it centers outputs around zero, aiding in faster convergence and improving gradient propagation. Unlike sigmoid, tanh offers stronger gradients and mitigates the vanishing gradient issue, while also avoiding the “dead neuron” problem often associated with ReLU. Initial weights were drawn from a distribution of small values to keep the activation functions within their linear regions during early training, which promotes more stable learning dynamics. Similarly, initializing biases with small values prevents early saturation and preserves symmetry breaking across neurons. These combined practices contribute to improved stability, faster convergence, and higher accuracy, making the proposed framework well-suited for estimating parameters in complex nonlinear systems such as those modeling cryptocurrency market dynamics.

Algorithm 2 Selection of Best Neural Network Architecture

```
1: Input: Crypto market share data; Search over hidden layer sizes  $h_1, h_2, h_3 = 5 \dots 10$ ; Learning rates  $\{0.00005, 0.0001\}$ ; Epochs  $\{100, 200\}$ 
2: for each combination of  $(h_1, h_2, h_3)$ , learning rate, and epoch count do
3:   for each bootstrap replicate  $b = 1$  to 5 do
4:     Train the FFNN on a moving-block bootstrap sample of training data
5:     Evaluate out-of-bag (OOB) validation cost for this replicate
6:     Save weights of the replicate with lowest OOB cost
7:   end for
8:   Estimate ODE parameters via least-squares using NN output; simulate with RK4; compute RMSE for BTC, ETH, ALTs
9:   Store architecture configuration and RMSE
10: end for
11: Sort all architectures by mean RMSE (RK4 simulation); select the top 10
```

From Algorithm 2 we see that the DNN searched over all $6^3 = 216$ possible 3-hidden-layer DNN architectures (from 5 to 10 neurons per layer), for each learning rate and epoch combination.

From Table 1, we select the best architecture based on the mean RMSE.

Table 1. Per-asset RMSEs (BTC, ETH, ALTs) and mean RMSE (MRMSE) for various NN architectures and hyperparameters.

h_1	h_2	h_3	BTC	ETH	ALTs	MRMSE	LR	Epochs
10	6	9	0.056355	0.097497	0.096556	0.083469	0.0001	200
10	6	9	0.063268	0.092336	0.097758	0.084454	0.0001	100
8	10	10	0.068767	0.097675	0.101360	0.089267	0.0001	200
7	7	6	0.062612	0.100960	0.102480	0.088684	0.0001	200
7	10	10	0.071430	0.096462	0.102860	0.090251	0.0001	200
6	5	10	0.063437	0.101870	0.102940	0.089416	0.0001	200
9	10	5	0.071015	0.099051	0.103100	0.091055	0.00005	200
9	9	10	0.068600	0.099730	0.104540	0.090957	0.0001	200
10	8	10	0.058409	0.114040	0.105150	0.092533	0.0001	200
9	10	5	0.072399	0.100790	0.105270	0.092820	0.0001	100

RMSE evaluates the accuracy of predictions by measuring the average error between observed and predicted value, based on validation procedure:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

(19)

Then, we extract the coefficients for all three and solve the system using the 4th-order Runge-Kutta method. Based on the comparison of the mean RMSE, we choose the best one, which we then include it in the graphs.

5. Numerical Experiment

5.1. Case Study Description

This study examines the historical market penetration and future prospects of the three crypto competitors. Monthly market share data for Bitcoin (BTC), Ethereum (ETH), and altcoins (ALTS) were sourced from CoinMarketCap historical snapshots [40], covering August 2015 to May 2023 as it is presented on Figure 2, a period of 93 months or almost 8 years.

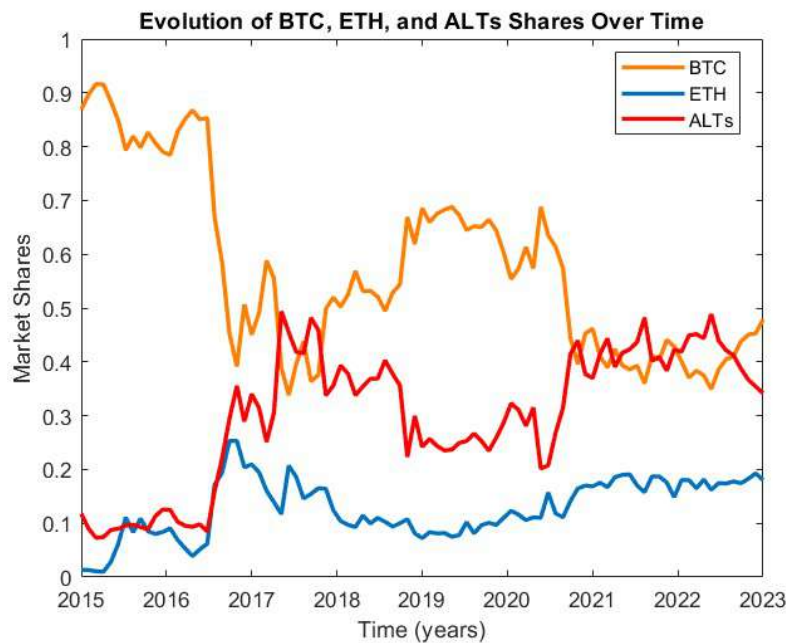


Figure 2. Evolution of cryptocurrency market from August 2015 to May 2023.

The market share of alternative cryptocurrencies (ALTS) is derived by subtracting the shares of Bitcoin (BTC) and Ethereum (ETH) from 1, ensuring that the total market share across all segments sums to unity.

The initial step in assessing the performance of the proposed framework involves estimating the parameters of the Lotka-Volterra system defined in Equation (2). While traditional approaches typically rely on predefined assumptions derived from the data, this study adopts a data-driven strategy. Specifically, a deep neural network is employed to infer the system parameters by effectively “training” the model based on time-series data of cryptocurrency market shares. This neural network-based estimation replaces heuristic techniques and enables the model to capture the underlying dynamics through optimization and learning from observed behavior. The results of the application of DNN for the case studied provided the following values for the corresponding parameters:

$$\begin{aligned}\frac{dx}{dt} &= x(0.5358 - 0.4521x + 0.7681y - 1.3624z), \\ \frac{dy}{dt} &= y(1.4664 - 1.2830x - 1.6997y - 1.7748z), \\ \frac{dz}{dt} &= z(0.1941 - 0.1602x - 0.5119y - 0.1015z).\end{aligned}\tag{20}$$

The estimated parameters, derived from neural network-assisted system identification, provide detailed insights into the mutual influences and self-regulatory mechanisms among Bitcoin (BTC, x), Ethereum (ETH, y), and Altcoins (ALTS, z) in the cryptocurrency market.

Intrinsic Growth Rates:

- $p_1 = 0.5358$ (BTC), $p_3 = 0.7681$ (ALTs), and $p_5 = 1.4664$ (ETH) are all positive, indicating that each asset class would grow in isolation absent any competition or inhibition.

Intraspecies (Self-Interaction) Terms:

- $p_2 = -0.4521$ (BTC) and $p_7 = -1.6997$ (ETH) are negative, reflecting strong self-limitation (diminishing returns) as market share increases.
- $p_{12} = -0.1015$ (ALTs) is also negative, indicating that, unlike before, the Altcoin sector exhibits some self-inhibitory effects at high share levels.

Interspecies Interactions:

- **BTC Equation:**
 - $p_4 = -1.3624$: A strong negative influence of ALTs on BTC, suggesting that as Altcoin share rises, BTC's growth is sharply suppressed.
 - $p_3 = 0.7681$: A substantial positive effect from ETH on BTC, implying that Ethereum growth tends to bolster Bitcoin's market share.
- **ETH Equation:**
 - $p_6 = -1.2830$: A strong negative effect of BTC on ETH, reversing the previous supportive role—here Bitcoin dominance inhibits Ethereum.
 - $p_8 = -1.7748$: A very strong inhibitory influence of ALTs on ETH, indicating fierce competition from the broader Altcoin market.
- **ALT Equation:**
 - $p_{10} = -0.1602$: Bitcoin slightly inhibits Altcoins.
 - $p_{11} = -0.5119$: Ethereum exerts a moderate negative effect on Altcoins.
 - $p_{12} = -0.1015$: Confirms Altcoins also self-inhibit at high share.

Summary:

All three asset classes have positive intrinsic growth, but strong self-limitation and mutual inhibition dominate:

- BTC faces sharp suppression from Altcoins ($p_4 < 0$) and diminishing returns on its own share ($p_2 < 0$). - ETH is heavily inhibited by both BTC ($p_6 < 0$) and ALTs ($p_8 < 0$), despite its own positive growth rate. - ALTs, while intrinsically growing, experience self-inhibition ($p_{12} < 0$) and competitive pressures from both BTC and ETH ($p_{10}, p_{11} < 0$).

These parameter signs suggest an intensely competitive market where no one asset can expand without encountering strong inhibitory forces from itself and its peers.

5.2. System Stability Testing

To investigate the local dynamics of the crypto market share model, we first estimate the parameters p_i and p_{ij} from observed market data using a neural network combined with optimization techniques. These parameters define the nonlinear Lotka–Volterra system (12), where u_i represent the market shares of BTC, ETH, and ALTs, respectively.

By setting the time derivatives to zero,

$$\frac{du_i}{dt} = 0, \quad (21)$$

we solve the resulting algebraic system and find that it will eventually settle to an equilibrium of approximately 41% for BTC, 20% for ETH, and 39% for ALTs:

$$u^* = (0.4150, 0.1953, 0.3897).$$

This represents the steady-state market shares predicted by the model under the estimated parameters. To understand how the system behaves near this equilibrium, we linearize the nonlinear system (12) around u^* . Denoting deviations from equilibrium as

$$U = u_1 - u_1^*, \quad V = u_2 - u_2^*, \quad W = u_3 - u_3^*, \quad (22)$$

the linearized dynamics satisfy

$$\frac{d}{dt} \begin{pmatrix} U \\ V \\ W \end{pmatrix} = J(u^*) \begin{pmatrix} U \\ V \\ W \end{pmatrix}, \quad (23)$$

where $J(u^*)$ is the Jacobian matrix of partial derivatives of the right-hand side of (12) evaluated at the equilibrium.

The general solution near equilibrium is

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = c_1 \mathbf{v}_1 e^{\lambda_1 t} + c_2 \mathbf{v}_2 e^{\lambda_2 t} + c_3 \mathbf{v}_3 e^{\lambda_3 t}, \quad (24)$$

where λ_i and \mathbf{v}_i are the real eigenvalues and eigenvectors of $J(u^*)$, and the coefficients c_i are obtained by projecting the initial deviation onto the eigenvector basis:

$$\begin{aligned} c_1 &= 0.1583, \\ c_2 &= 0.1082, \\ c_3 &= -0.5636. \end{aligned} \quad (25)$$

Analyzing the eigenvalues of $J(u^*)$, we find

$$\lambda_1 = -0.4850, \quad \lambda_2 = -0.2072, \quad \lambda_3 = -0.0013, \quad (26)$$

all of which are strictly negative. This confirms that u^* is a locally asymptotically stable node. The absence of any imaginary parts indicates purely monotonic convergence to equilibrium, without oscillatory transients.

The phase portrait shown in Figure 3 displays trajectories initiated from diverse market share configurations. All trajectories converge smoothly to the equilibrium u^* , supporting the model's predictive validity and its resilience to small perturbations.

This comprehensive approach—combining parameter estimation with Lotka–Volterra equilibrium analysis and linear stability theory—follows the methodology in [28], where the system's long-term dynamics are characterized by a unique stable equilibrium. Hence, barring major external shocks, the model robustly predicts that cryptocurrency market shares will stabilize at u^* .

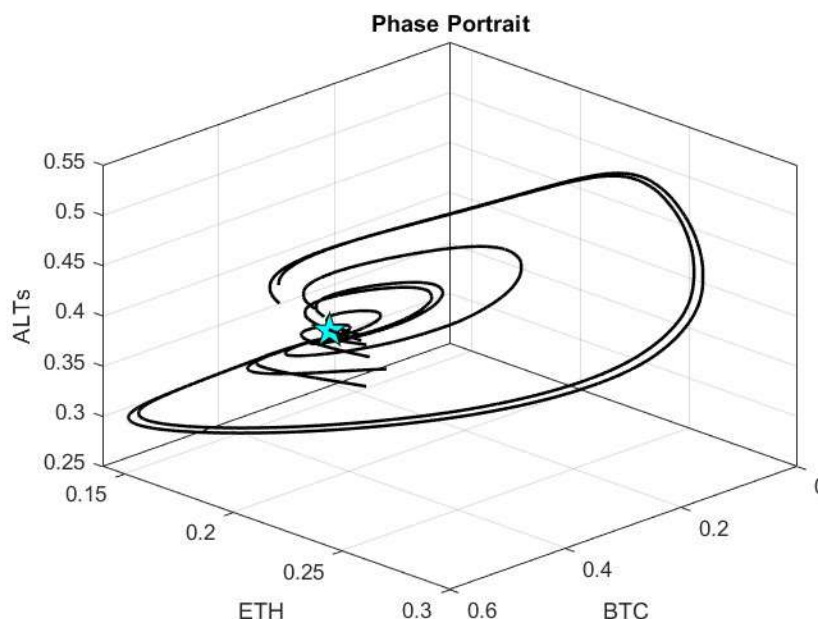


Figure 3. Trajectories from various initial market shares converging to the unique stable equilibrium u^* .

5.3. Case Study Results

In this section, we compare the effectiveness of our hybrid DNN-RK4 method against a range of ARIMA models for predicting the market shares of BTC, ETH, and ALTs. The DNN-RK4 approach demonstrates the closest alignment to the real-world market share data, consistently producing the lowest prediction errors across the tested period.

The ARIMA (AutoRegressive Integrated Moving Average) model is a widely used statistical method for analyzing and forecasting time-series data. It combines three components: the autoregressive (AR) part models the relationship between current and past values; the integrated (I) part handles non-stationarity by differencing the data; and the moving average (MA) part captures dependencies between residual errors. By adjusting its parameters (p , d , q), ARIMA can effectively model and forecast trends and patterns in sequential financial data, including cryptocurrency prices. In the $ARIMA(p, d, q)$ model, p denotes the order of the autoregressive component (i.e., the number of lagged values of the series included), d represents the degree of differencing applied to make the series stationary (i.e., the number of times the series is replaced by the difference between consecutive observations), and q specifies the order of the moving-average component (i.e., the number of lagged forecast errors incorporated). Together, these three parameters allow ARIMA models to capture both short-term autocorrelation (through p and q) and nonstationarity (through d) in time series data.

The various ARIMA configurations [42], while capable of capturing some of the underlying temporal patterns, generally yield higher Root Mean Squared Error (RMSE) values compared to the DNN-RK4 method. To facilitate an objective assessment, all models are evaluated using the RMSE metric, and results are summarized both in graphical form and in a results table.

This analysis provides strong evidence that the DNN-RK4 method offers superior predictive power for modeling the evolution of cryptocurrency market shares when compared to traditional ARIMA time series approaches.

To evaluate the forecasting accuracy of the proposed DNN-RK4 approach, grounded in a nonlinear dynamical system framework, we compared its performance with that of classical time series models, specifically ARIMA. We evaluated multiple $ARIMA(p, d, q)$ configurations, including $(2, 1, 2)$, $(2, 1, 1)$, and $(1, 0, 1)$. For each candidate model, we fitted the parameters on the training set and computed the RMSE on the validation set. The $ARIMA(2, 1, 2)$ configuration produced the lowest RMSE, and was therefore chosen for comparison against our proposed method. The comparison focuses on the three

main components of the market: Bitcoin (BTC), Ethereum (ETH), and alternative cryptocurrencies (ALTs).

The DNN-RK4 model, driven by parameters estimated through a neural network, exhibited superior predictive performance across all market segments. For BTC, the RK4 model achieved an RMSE of 0.06865, significantly outperforming the ARIMA model’s RMSE of 0.14742 as we notice on Figure 4. In the case of ETH, we see on Figure 5 that the differential was similarly notable-0.02682 (RK4) compared to 0.03665 (ARIMA). For the ALT category, the DNN-RK4 model again showed improved accuracy, reducing the error from 0.11890 (ARIMA) to 0.05581 as shown in Figure 6. These results highlight the advantage of using a model based on competitive dynamics to capture nonlinear relationships inherent in the cryptocurrency ecosystem, where investor behavior, technological updates, and market sentiment evolve rapidly.

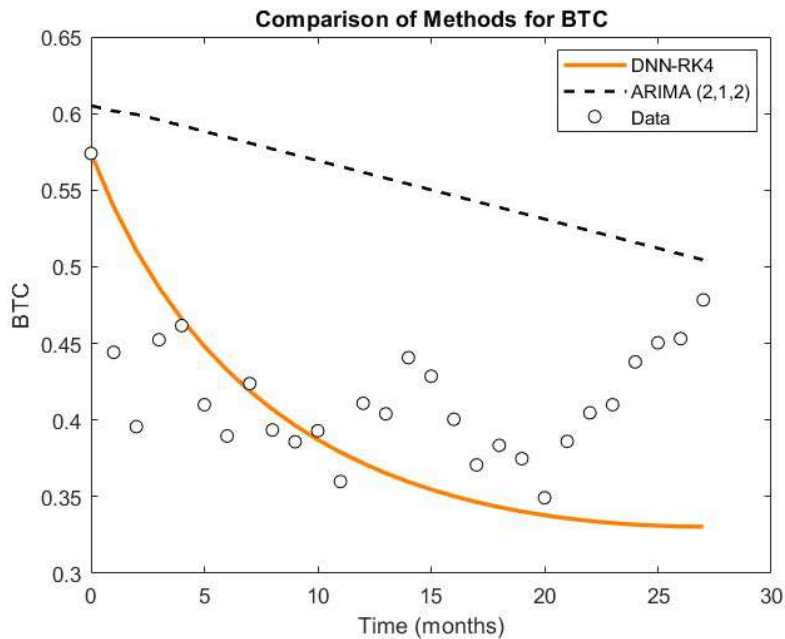


Figure 4. Comparison of BTC.

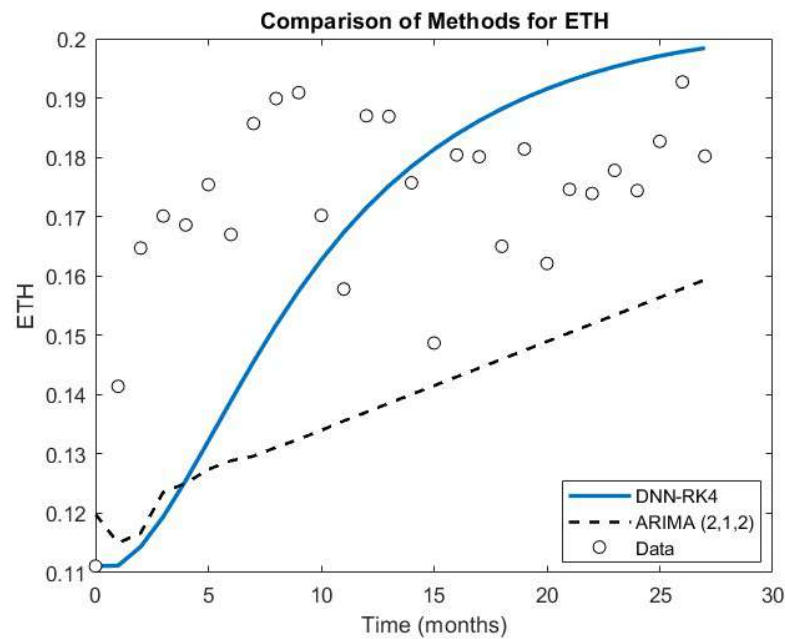


Figure 5. Comparison of ETH.

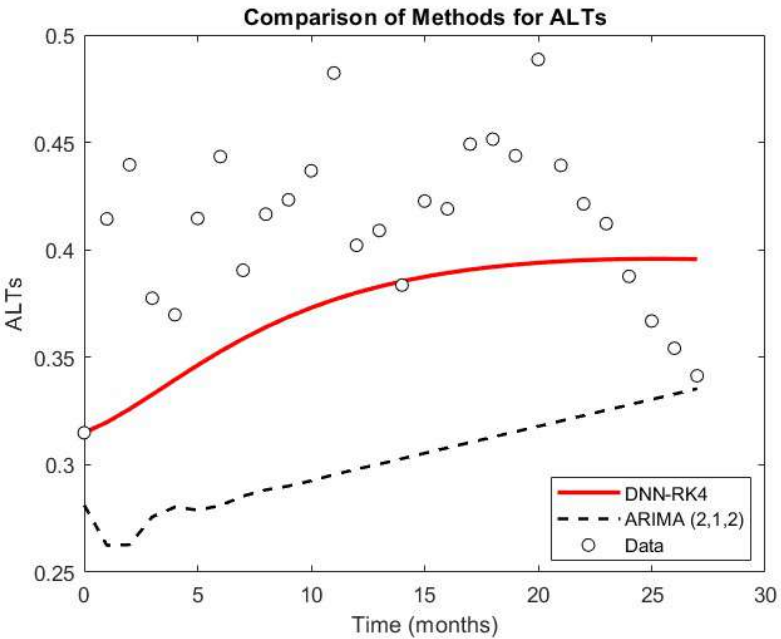


Figure 6. Comparison of ALTs.

The cryptocurrency market is characterized by high volatility, interdependence between major assets, and continuous entry of new tokens. These features make traditional linear models less effective in capturing complex feedback mechanisms and competitive interactions. The DNN-RK4 method, supported by neural network-driven parameter learning, demonstrates strong adaptability in forecasting such nonlinear systems. It is particularly useful in capturing turning points and saturation effects, which are common in speculative and innovation-driven markets like crypto.

Table 2. Root Mean Squared Error (RMSE) values for different methods and components BTC, ETH and ALTs.

Method	RMSE for BTC	RMSE for ETH	RMSE for ALTs
DNN-RK4	0.06865	0.02682	0.05581
ARIMA (2,1,2)	0.14742	0.03665	0.11890

Further supporting these observations, Table 2 compares RMSE values across multiple modeling strategies. The DNN-RK4 model, in particular, demonstrated excellent forecasting capabilities, with RMSE values of 0.06865, 0.02682, and 0.05581 for BTC, ETH, and ALTs, respectively. Compared to more traditional techniques-including ARIMA model approaches-the DNN-RK4 consistently provided more accurate results, affirming its effectiveness in modeling the dynamic interactions of market share evolution in a decentralized digital asset environment.

6. Discussion

To further improve the robustness and accuracy of cryptocurrency market modeling, future research will investigate a range of advanced machine learning and hybrid approaches tailored to the unique challenges of crypto dynamics. One promising direction involves Temporal Convolutional Networks (TCNs) [43], which can outperform recurrent models in sequence modeling tasks while offering parallel processing advantages and stable gradients over long sequences-a critical factor when analyzing extensive crypto market histories.

In addition, the use of Graph Neural Networks (GNNs) will be explored to represent the complex interdependencies between different cryptocurrencies [44]. Crypto assets often exhibit co-movements, influenced by trading volumes, blockchain correlations, or investor sentiment. Modeling these rela-

tionships as a graph enables more informed parameter estimation and can uncover hidden structures in the market.

Another area of potential is Neural ODEs, which integrate the strengths of neural networks with differential equations. This approach is particularly suitable for learning continuous-time representations, offering a natural fit for systems like Lotka–Volterra that evolve over time. Neural ODEs can adaptively model temporal changes in market competition and volatility with fewer parameters and better interpretability [45].

The integration of Sentiment Analysis from social media and blockchain news will also be considered. Since investor behavior in crypto is highly reactive to news, incorporating textual data through models like BERT or domain-specific transformer variants (e.g., FinBERT) could improve forecasting accuracy by providing context-aware inputs to dynamic models [46]. Finally, online learning techniques will be examined for their ability to continuously update model parameters as new data arrives. This is critical for maintaining prediction accuracy in a market characterized by rapid innovation, regulatory shifts, and speculative bubbles. Something important to note here is that, because our Lotka–Volterra model assumes constant coefficients, it cannot capture one-time shocks such as Bitcoin’s May 2020 halving or regulatory crackdowns in 2021. As a result, the estimated parameters may partially reflect these exogenous events rather than pure “peer competition.” Future work will incorporate time-varying coefficients, exogenous covariates, and the previously discussed methodologies to address this limitation, aiming to develop a more comprehensive, adaptive, and robust framework for modeling and forecasting market share evolution in the cryptocurrency domain.

7. Conclusions

The proposed framework demonstrates that integrating deep neural networks with Lotka–Volterra competitive dynamics provides a robust, data-driven means of capturing and forecasting the evolving structure of cryptocurrency market shares. By estimating interaction coefficients directly from observed BTC, ETH, and ALT time series and then simulating their joint evolution via a fourth-order Runge–Kutta scheme, the model not only outperforms traditional ARIMA benchmarks but also yields interpretable insights into mutual competitive pressures (e.g., the asymmetric influence of BTC on ALTs and ETH on BTC). Moreover, local stability analysis confirms that the estimated equilibrium remains locally asymptotically stable, explaining why any transient deviations decay in the absence of exogenous shocks. These findings both validate the Lotka–Volterra–NN hybrid approach in a highly nonlinear, volatile domain and furnish a principled basis for strategic decision-making. Looking ahead, extending the model to accommodate time-varying coefficients, exogenous covariates, or graph-based interactions among a broader set of tokens could further enhance its adaptability and predictive power, ensuring that it remains responsive to the rapid innovation and intermittent shocks characteristic of global crypto markets.

Author Contributions: D.K., D.P. and K.G. conceived of the idea, designed and performed the experiments, analyzed the results, drafted the initial manuscript and revised the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Grant (81845) from the Research Committee of the University of Patras via “C. CARATHEODORI” program.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data used in this study are publicly available online at <https://coinmarketcap.com/historical/>. The specific dataset used for analysis is also provided in the supplementary material.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. White GRT. Future applications of blockchain in business and management: A Delphi study . *Strategic Change* ,**2017**, 26, 439–451.
2. Bitcoin.org. Available online: <https://bitcoin.org/bitcoin.pdf>
3. Tradingview.com .Available online: <https://www.tradingview.com/symbols/BTC/>
4. Taleb, N. et al.Prospective applications of blockchain and bitcoin cryptocurrency technology . *TEM J* ,**2019**, 8, 48–55.
5. Navamani, T. M. A Review on Cryptocurrencies Security . *Journal of Applied Security Research* ,**2021**, 18(1), 49–69.
6. T. Nandy, U. Verma, P. Srivastava, D. Rongara, A. Gupta and B. Sharma.The Evaluation of Cryptocurrency: Overview, Opportunities, and Future Directions . *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)* ,**2023**, 1421–1426.
7. Tradingview.com .Available online: <https://www.tradingview.com/symbols/ETH/>
8. Forbes.com .Available online: <https://www.forbes.com/sites/lawrencewintermeyer/2021/08/12/institutional-money-is-pouring-into-the-crypto-market-and-its-only-going-to-grow/?sh=1a5424261459>
9. Nicholas Taleb, N.Bitcoin, currencies, and fragility . *Quantitative Finance* ,**2021**, 21(8) , 1249–1255.
10. VISUAL CAPITALIST. Available online: <https://www.visualcapitalist.com/crypto-ownership-growth-by-region/>
11. Javarone, M. A. and Wright, C. S.From bitcoin to bitcoin cash: A network analysis . *In Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems* ,**2018**, 77–81.
12. Donet, J. A. D., Pérez-Sola, C. and Herrera-Joancomartí, J.The bitcoin p2p network . *In International conference on financial cryptography and data security* ,**2014**, 87–102.
13. Kjærland, F., Khazal, A., Krogstad, E. A., Nordstrøm, F. B. and Oust. An analysis of bitcoin's price dynamics . *J. Risk Financ. Manag* ,**2018**,11, 63.
14. Velankar, S., Valecha, S. and Maji, S.Bitcoin price prediction using machine learning . *In 2018 20th International Conference on Advanced Communication Technology (ICACT)* ,**2018**, 144–147.
15. ElBahrawy Abeer, Alessandretti Laura, Kandler Anne, Pastor-Satorras Romualdo and Baronchelli Andrea.2017Evolutionary dynamics of the cryptocurrency markeT . *R. Soc. Open Sci* ,**2017**, 4:170623.
16. Wu, K., Wheatley, S. and Sornette .Classification of cryptocurrency coins and tokens by the dynamics of their market capitalizations . *Soc. Open Sci* ,**2018**, 5, 180381 .
17. J.D. Murray.Mathematical Biology. *Springer-Verlag* ,**1993**, New York .
18. A. W. Wijeratnea, F. Yi, and J. Wei. "Bifurcation analysis in the diffusive Lotka-Volterra system: An application to market economy. *Chaos,Solitons Fractals* , **2009**, vol. 40 , 902–911 .
19. D. Neal.Introduction to Population Biology. *New York: Cambridge University Press* , **2004**.
20. W. E. Boyce and R. C. DiPrima .Elementary Differential Equations and Boundary Value Problems, . *Hoboken, NJ: Wiley* , **2004**, 8th ed.
21. J. C. Fisher and R. H. Pry .“A simple substitution model of technological change,” . *Technol. Forecast. Soc. Change* , **1971**, vol. 3, 75–88.
22. L. P. Rai.“Appropriate models for technology substitution” . *J. Sci. Ind. Res.* ,**1999**, vol. 58 , 14–18.
23. M. Begon, C. Townsend, and J. Harper. Ecology: From Individuals to Ecosystems . *Oxford, U.K.: Blackwell* ,**2006**, 4th ed.
24. T. H. Fay and J. C. Greeff. “A three species competition model as a decision support tool” . *Ecol. Modell* ,**2008**, vol. 211,, 142–152.
25. P. G. L. Leach and J. Miritzis.“Analytic behaviour of competition among three species” . *J. Nonlinear Math. Phys* ,**2006**, vol. 13, 535–548.
26. Olivença DV, Davis JD and Voit EO. Inference of dynamic interaction networks: A comparison between Lotka-Volterra and multivariate autoregressive models . *Front. Bioinform* ,**2022**, Volume 2 .
27. Kastoris, D.; Giotopoulos, K.; Papadopoulos, D. Neural Network-Based Parameter Estimation in Dynamical Systems . *Information* ,**2024**, vol. 15, 809.
28. C. Michalakelis, T.S. Sphicopoulos, D. Varoutas.Modelling competition in the telecommunications market based on the concepts of population biology.*Transactions on Systems, Man and Cybernetics. Part C: Applications and Reviews* 4 **2011**, 200–210.
29. G. Bebis and M. Georgiopoulos. Feed-forward neural networks . *In IEEE Potentials* ,**1994**, Vol 13 (4), 27–31.

30. L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu and L. Shao. Normalization Techniques in Training DNNs: Methodology, Analysis and Application . in *IEEE Transactions on Pattern Analysis and Machine Intelligence* ,**2023**, vol. 45, no. 8, 10173–10196.
31. J. C. Butcher. Numerical Methods for Ordinary Differential Equations . 3 ,**2016**, 10, 143–331.
32. Tan Delin , Chen Zheng. "On A General Formula of Fourth Order Runge-Kutta Method" . *Journal of Mathematical Science & Mathematics Education* , **2012**, 7(2), 1–10.
33. J. R. Dormand, M. E. A. El-Mikkawy, P. J. Prince. Families of Runge-Kutta-Nyström formulae . *IMA J. Numer* ,**1987**, 7, 235–250.
34. D. F. Papadopoulos, T. E. Simos. A new methodology for the construction of optimized Runge-Kutta- Nyström methods . *International Journal of Modern Physics C* ,**2011**, 22 (6), 623–634.
35. D. F. Papadopoulos, T. E. Simos. The use of phase lag and amplification error derivatives for the construction of a modified Runge-Kutta-Nyström method. *Abstract and Applied Analysis* ,**2013**, 910624.
36. Papadopoulos, D.F., Anastassi, Z.A., Simos, D.F. The use of phase-lag and amplification error derivatives in the numerical integration of ODEs with oscillating solutions. *AIP Conference Proceedings* , 2009.
37. Papadopoulos, D.F., Anastassi, Z.A., Simos, D.F. A zero dispersion RKN method for the numerical integration of initial value problems with oscillating solutions. *AIP Conference Proceedings* , 2009.
38. Papadopoulos, D.F. A Parametric Six-Step Method for Second-Order IVPs with Oscillating Solutions. *Mathematics* ,**2024**, vol. 12, 3824.
39. Steyerberg, Ewout W, Harrell, Frank E, Jr , Borsboom, Gerard J.J.M, Eijkemans, M.J.C, Vergouwe, Yvonne, Habbema, J. Dik F. Internal validation of predictive models: Efficiency of some procedures for logistic regression analysis . *Journal of Clinical Epidemiology* ,**2001**, 54 (8), 774–781.
40. coinmarketcap.com . Available online: <https://coinmarketcap.com/historical/>
41. Schulz, A. W. Equilibrium modeling in economics: a design-based defense. *Journal of Economic Methodology* ,**2024**, 31(1), 36–53.
42. Kontopoulou, V.I.; Panagopoulos, A.D.; Kakkos, I.; Matsopoulos, G.K. A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks . *Future Internet* ,**2023**, 15, 255.
43. Ying Liu, Xiaohua Huang, Liwei Xiong, Ruyi Chang, Wenjing Wang, Long Chen. Stock price prediction with attentive temporal convolution-based generative adversarial network . *Array* ,**2025**, 25.
44. Zening Zhao, Jinsong Wang, Jiajia Wei. Graph-Neural-Network-Based Transaction Link Prediction Method for Public Blockchain in Heterogeneous Information Networks . *Blockchain: Research and Applications* ,**2025**.
45. Coelho, C.; da Costa, M.F.P.; Ferrás, L.L. XNODE: A XAI Suite to Understand Neural Ordinary Differential Equations . *AI* ,**2025**, 6, 105.
46. Vincent Gurgul, Stefan Lessmann, Wolfgang Karl Härdle. Deep learning and NLP in cryptocurrency forecasting: Integrating financial, blockchain, and social media data . *International Journal of Forecasting* ,**2025**.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.