Journal of Global Economic Insights, 2026, 2(1), 17-39
https://doi.org/10.71204/b2c2rb55

Numerical Signal Design for Crypto Intelligence

Sai Zhang [1,*], Chongbin Luo [2], Annika Cai [3], Yeran Lu [3]

[1] Stuart Business School, Illinois Institute of Technology, IL 60616, US

[2] Fintech College Shenzhen University, Shenzhen 518060, China

[3] Gies College of Business, University of Illinois Urbana-Champaign, IL 61801, US

*** Correspondence:**

Sai Zhang

saizhang086@gmail.com

Received: 1 September 2025/ Accepted: 10 September 2025 / Published online: 14 September 2025

Abstract

This study presents a novel framework that bridges deterministic numerical algorithms with computational finance to support interpretable machine learning applications in cryptocurrency analysis. By applying Newton's method, the Trapezoidal Rule, and a hybrid Euler–Adams-Bashforth solver, we generate structured numerical features that capture convergence, integration precision, and differential dynamics, respectively. These feature vectors are constructed from both synthetically generated sequences and real AVAX-USD log return data, enabling a direct comparison between theoretical numerical behavior and market-driven fluctuations. A deterministic labeling rule, based on the parity of the integer sum of the features, defines class boundaries with geometric regularity, allowing the K-Nearest Neighbors classifier to operate in a feature space shaped by mathematically grounded transformations. The results reveal that classical numerical methods, when applied to financial time series, produce stable, class-separable patterns that are well-suited for local classification and boundary interpretation. Through decision boundary visualization, error convergence analysis, and joint feature interaction plots, the paper demonstrates that numerical approximation techniques can illuminate latent structural signals in crypto price behavior. This framework advances the integration of scientific computing and data-driven finance, offering a new paradigm for understanding digital asset dynamics through the lens of deterministic modeling.

Keywords: Deterministic Numerical Methods; Computational Finance; Cryptocurrency Analysis; Feature Engineering; Scientific Computing

1. Introduction

1.1. Background and Motivation

In the evolving landscape of machine learning and data-driven modeling, the construction of structured, interpretable, and mathematically principled feature spaces has regained importance. While modern algorithms such as deep neural networks excel in extracting features from raw data through hierarchical representation learning (Goodfellow, Bengio, & Courville, 2016), their opaque internal mechanisms often lack transparency—raising concerns in high-stakes decision-making contexts (Doshi-Velez & Kim, 2017; Lipton, 2018; Rudin, 2019; Molnar, 2022). In contrast, classical numerical methods, such as Newton's method, the Trapezoidal Rule, and multistep solvers like the Adams-Bashforth method, provide deterministic convergence, quantifiable error bounds, and strong theoretical guarantees (Atkinson, 1989; Burden & Faires, 2011; Butcher, 2016; Hairer & Wanner, 1991; Iserles, 2008). These properties, long valued in scientific computing (Henrici, 1974; Heath, 2002; Quarteroni, Sacco, & Saleri, 2007), position them as powerful tools for structured feature engineering in machine learning frameworks.

This paper is motivated by the hypothesis that the numerical behavior embedded in these classical algorithms—specifically convergence stability, integration precision, and dynamic trajectory generation—can be directly repurposed to construct feature vectors that support supervised classification. The goal is to bridge a gap between deterministic numerical computation and flexible machine learning classifiers such as K-Nearest Neighbors (Cover & Hart, 1967; Dasarathy, 1991; Altman, 1992; Peterson, 2009), whose geometric foundations are well-suited for exploring local structure in data. Furthermore, by embedding these methods in both simulated and real-world data contexts, such as financial time series derived from AVAX-USD, we aim to evaluate the practical viability and interpretive value of numerical methods as data transformation engines.

1.2. Novelty

The key novelty of this study lies in transforming classical numerical analysis algorithms—root-finding, quadrature, and ODE solvers—into systematic feature generation tools for classification problems. Unlike traditional statistical descriptors or automated embeddings, the proposed approach leverages mathematical convergence behavior to create features with traceable analytical origins (Deuflhard, 2011; Epperson, 2013; Sauer, 2017; LeVeque, 2007). The methodology explicitly constructs three feature groups using Newton's method to approximate $\sqrt{2}$, the Trapezoidal Rule to estimate definite integrals of sine functions, and the Euler-Adams-Bashforth scheme to solve the ODE $dy/dx=x+y$. These transformations capture distinct numerical textures—rapid fixed-point convergence, smooth integrative summation, and compounding differential growth—which are typically underrepresented in the feature spaces of conventional machine learning pipelines (Griffiths & Higham, 2010; Golub & Ortega, 2014; Stoer & Bulirsch, 2002).

To complement these features, we introduce a deterministic labeling rule based on the parity of the integer part of the sum $X_1+X_2+X_3$. This labeling mechanism allows for explainable boundary formation while preserving numerical continuity, thereby avoiding the stochastic variability often associated with empirically assigned labels. This formal and reproducible labeling approach provides an ideal testbed for evaluating geometric learning behavior in interpretable models such as KNN (Weinberger & Saul, 2009). Furthermore, this novel synthesis of numerical signal construction and classification logic expands the understanding of how algorithmic behavior in numerical analysis can map onto data structures suitable for learning tasks.

While the framework necessarily illustrates classical numerical methods such as Newton's root-finding or the Trapezoidal Rule, these derivations serve only as a foundation. The scholarly contribution lies not in reintroducing these algorithms but in repurposing their convergence, error, and dynamic behaviors as structured features for classification in financial contexts. To our knowledge, this is the first work to formalize numerical solvers as feature generators and to demonstrate their geometric separability when applied to cryptocurrency log returns. This distinction ensures that the paper advances methodological innovation beyond pedagogical demonstration.

1.3. Contribution

This paper offers a new framework for structured feature engineering rooted in the deterministic behavior of classical numerical methods. It establishes that methods long considered purely computational—such as Newton's iterative solver, Trapezoidal integration, and Adams-Bashforth ODE approximations—can serve as generative mechanisms for highly structured, low-dimensional feature vectors. These vectors demonstrate desirable learning properties: low intra-class variance, bounded global error, and high geometric separability in classification contexts (Boyd & Vandenberghe, 2004; Higham, 2002; Ralston & Rabinowitz, 2001). We show that when these features are applied to both synthetic datasets and real AVAX-USD financial data, they maintain mathematical interpretability while enabling robust performance in simple learning models.

The contributions of this work intersect with emerging efforts to create transparent and explainable machine learning systems (Gilpin et al., 2018; Shalev-Shwartz & Ben-David, 2014), and complement a growing interest in interpretable geometry-driven approaches to data construction (Tian, 2024a; Tian & Deng, 2024c; Tian et al., 2024b, 2024d). By formalizing the numerical roots of feature generation, the paper establishes a principled foundation for deterministic classification pipelines and opens new pathways for pedagogical and applied research in hybrid numerical-learning methodologies.

1.4. Construction

This paper proposes a novel classification framework in which features are derived from deterministic numerical methods rather than stochastic or empirical data sources. By applying classical algorithms—Newton's method, the Trapezoidal Rule, and the Adams-Bashforth solver—we generate structured, interpretable data that reflect the underlying mathematical behavior of each algorithm. These numerically constructed features are used as input for

classifier, with labels assigned through a deterministic parity rule based on the aggregate behavior of the feature groups. This design enables reproducibility, analytical traceability, and pedagogical insight into the behavior of numerical approximations within machine learning pipelines.Section 2 introduces the numerical feature generation process, detailing the construction of the three variable groups. corresponding to root-finding, numerical integration, and ordinary differential equation solvers, respectively.Section 3 reports experimental results, including accuracy metrics, geometric analysis of the feature space, and a discussion on the interpretability of results derived from numerical structures.Section 4 is disscussion and finally, Section 5 concludes the study with reflections on the contributions, limitations of the current framework, and directions for future research, particularly in extending the approach to more advanced models and hybrid symbolic-ML architectures.

2. Methodology

This section presents a comprehensive mathematical formulation of the numerical classification framework. The process includes constructing input vectors using Newton's method, the Trapezoidal Rule, and the Adams-Bashforth method; assigning labels based on a deterministic rule; and performing classification using KNN. Each method contributes a unique mathematical transformation to construct features X_1, X_2, X_3 which are used for classification in a structured feature space.

2.1. Newton's Method for Root-Finding (X_1)

This paragraph describes how we use Newton's method to generate the first feature group X_1 by solving the equation $f(x)=x^2-2$, whose root is $\sqrt{2}$. Newton's method iteratively refines approximations to this root using its derivative. The convergence is quadratic and fast, and this process yields a vector of approximated root values, forming a numerically stable and smoothly converging feature space.

We begin with the function f(x) whose root we want to find:

$$f(x)=x^2-2 \tag{1}$$

Newton's method requires the derivative of f(x):

$$f'(x)=2x \tag{2}$$

General Newton update rule:

$$x_{n+1}=x_n-\frac{f(x_n)}{f'(x_n)} \tag{3}$$

Plug in (1) and (2):

$$x_{n+1}=x_n-\frac{x_n^2-2}{2x_n} \tag{4}$$

simplifies to a well-known iteration for computing square roots:

$$x_{n+1}=\frac{1}{2}\left(x_n+\frac{2}{x_n}\right) \tag{5}$$

With x_0=1 the iterations yield:

$$x_1=1.5\ x_2=1.4167, x_3=1.4142 \tag{6}$$

As n→∞ the sequence converges to:

$$\lim_{n\to\infty} x_n=\sqrt{2} \tag{7}$$

Quadratic convergence guarantee:

$$\left|x_{n+1}-\sqrt{2}\right|<\left|x_n-\sqrt{2}\right|^2 \tag{8}$$

The converged sequence is stored as the first feature group:

$$X_1=\{x_n\}_{n=0}^{N_1}\in\mathbb{R}^{N_1} \tag{9}$$

2.2. Trapezoidal Rule for Numerical Integration (X_2)

This section explains how to build the second feature group X_2 using the Trapezoidal Rule to approximate the integral of the sine function over the interval $[0,\pi]$. The rule is second-order accurate and well-behaved. The trapezoidal sum creates a vector of approximated integrals, suitable for representing smooth, continuous dynamics such as sensor data or area-under-curve measurements.

Define the target function and integration limits:

$$f(x)=\sin(x),\quad a=0,\quad b=\pi \tag{10}$$

The interval is divided into nnn subintervals of equal width:

$$h=\frac{\pi}{n} \tag{11}$$

The i-th point in the partitioned domain is:

$$let:\ x_i=a+ih,\quad i=0,\ldots,n \tag{12}$$

We approximate the integral using:

$$T_n=\frac{h}{2}\left[f(a)+2\sum_{i=1}^{n-1}f(x_i)+f(b)\right] \tag{13}$$

Evaluate known function values

Since sin(0)=0and sin(π)=0the boundaries contribute zero:

Define：$\sin(0)=0,\quad \sin(\pi)=0,$

For n=4 the approximation becomes

$$T_4=\frac{\pi}{8}\left[2\left(\sin(\pi/4)+\sin(\pi/2)+\sin(3\pi/4)\right)\right]\qquad T_4\approx 1.896 \tag{14}$$

Exact integral for comparison

The true integral of sin(x)from 0 to π is:

$$\lim_{n\to\infty} T_n=\int_0^{\pi}\sin(x)dx=2 \tag{15}$$

Error bound for Trapezoidal Rule

$$E_T=-\frac{(b-a)^3}{12n^2}f''(\xi),\quad \xi\in(a,b) \tag{16}$$

Construct the feature group

When using multiple values of n，the resulting approximations form the feature group:

$$X_2=\{T_n\}_{n=1}^{N_2}\in\mathbb{R}^{N_2} \tag{17}$$

2.3. Euler and Adams-Bashforth Methods for ODE Solving (X_3)

In this paragraph, we describe the generation of the third feature group X_3 through numerical solution of an ordinary differential equation $dy/dx=x+y$. The Euler method initializes the sequence, and the Adams-Bashforth 2-step method improves the solution with higher-order accuracy. This feature group models dynamic or temporal behavior, providing a diverse signal for classification.

Define the differential equation

We solve the first-order ODE with the right-hand side function:

$$f(x,y)=x+y \tag{18}$$

Specify the initial condition, let $y_0=1,\quad x_0=0$

Initialize with the Euler method

Euler's method is used to generate the first 3 steps:

$$\begin{aligned} y_1&=y_0+h\cdot f(x_0,y_0)=1.1\\ y_2&=y_1+h\cdot f(x_1,y_1)=1.22\\ y_3&=y_2+h\cdot f(x_2,y_2)=1.3733 \end{aligned} \tag{19}$$

Define notation for Adams-Bashforth method, let $f_n=f(x_n,y_n)$

Apply the Adams-Bashforth 3-step formula

This higher-order multistep method improves accuracy using previous evaluations:

$$y_{n+1}=y_n+\frac{h}{12}(23f_n-16f_{n-1}+5f_{n-2}) \tag{20}$$

Store the results as a feature group

The full trajectory of yny_nyn values becomes:

$$X_3=\{y_n\}_{n=0}^{N_1}\in\mathbb{R}^{N_3} \tag{21}$$

The global truncation error of the AB-3 method is:

$$\text{Global error}=O(h^4) \tag{22}$$

Convergence to exact solution

As step size h→0, the numerical solution approaches the true function:

$$\lim_{h\to0}X_3(h)\to y(x) \tag{23}$$

Classification presents the classification methodology, which includes two deterministic components: a mathematically defined labeling rule and a supervised learning model built on the K-Nearest Neighbors (KNN) algorithm. Together, these components provide a controlled

environment for evaluating how structured numerical features influence class separation and decision boundaries. The binary labels are assigned using a deterministic parity rule applied to the aggregated feature values. For each instance, the feature vectors X_1, X_2, and X_3 are computed using Newton's method, the Trapezoidal Rule, and the Adams-Bashforth method, respectively. These are then summed element-wise to produce a scalar value $S_i=\sum_j \left(X_{1j}+X_{2j}+X_{3j}\right)$. The class label y_i is determined by extracting the integer part of S_i, denoted as $\llcorner S_i \lrcorner$, and applying a parity check:

$$y_i=\begin{cases}0, & \text{if}\llcorner S_i \lrcorner \text{ is even}\\ 1, & \text{if}\llcorner S_i \lrcorner \text{ is odd}\end{cases} \tag{24}$$

This rule generates balanced and explainable binary labels while preserving deterministic reproducibility. It also introduces a nonlinear decision boundary that depends on the dynamic interplay among the three numerical methods. Once the labeled dataset $\{(X_i, y_i)\}_{i=1}^{N}$ is constructed, we apply the K-Nearest Neighbors algorithm to perform classification. The full feature vector $X_i=[X_1, X_2, X_3]$ is treated as a point in high-dimensional Euclidean space. The classifier assigns labels to new test samples based on the majority class among their k nearest neighbors in the training set. We use standard Euclidean distance as the similarity measure. The choice of k is empirically validated for optimal performance. The use of KNN in this context allows for a transparent examination of how geometric patterns—originating from numerical approximation behavior—affect classification outcomes. This methodology bridges deterministic numerical computation and data-driven modeling, demonstrating that classical methods in numerical analysis can give rise to structured, interpretable, and classifiable feature spaces suitable for modern machine learning tasks.

Justification of Numerical Methods. We specifically chose Newton's method, the Trapezoidal Rule, and the Euler–Adams-Bashforth scheme because each provides distinct numerical properties that map naturally onto financial data features. Newton's method offers quadratic convergence, producing stable fixed-point approximations that highlight convergence speed and error decay. The Trapezoidal Rule delivers second-order accuracy and smooth area-based approximations, making it suitable for representing cumulative effects such as transaction volumes or integrated volatility measures. Finally, the Euler–Adams-Bashforth scheme captures sequential dynamics and temporal growth through higher-order multistep extrapolation, aligning closely with autoregressive or momentum-driven behaviors observed in digital assets. While alternative techniques such as Runge–Kutta or Simpson's rule could have been used, our selection balances interpretability, computational efficiency, and complementary signal textures. Together, these methods provide convergence-, integration-, and dynamics-based features that enrich the classification space with interpretable and diverse mathematical signatures.

2.4. Theoretical Guarantees: Convergence and Stability of Numerically Engineered Crypto Features Restult

In this section, we formalize the convergence behavior and stability characteristics of the numerical methods used to generate the feature space (X_1, X_2, X_3) for AVAX-USD log return

classification. These methods—Newton's root solver, the Trapezoidal Rule, and the Euler–Adams-Bashforth (AB2) ODE integrator—are not only computational tools, but also theoretical foundations for constructing structured, robust, and explainable features. We present a formal theorem and supporting lemma to demonstrate how classical convergence results provide mathematical guarantees for crypto signal feature engineering.

Theorem 2.1 (Structured Feature Convergence Theorem)

Let $f_1(x)=x^2-2$, $f_2(x)=\sin(x)$, and $f_3(t,y)=x(t)+y(t)$, where $x(t)$ is the log-return signal of AVAX. Define the numerically generated feature vector (X_1,X_2,X_3) as follows:

X_1 is constructed by applying Newton's method to f_1 with initial guess $x_0\in(1,2)$.

X_2 is derived from approximating $\int_0^{\pi} f_2(x)dx$ using the Trapezoidal Rule with step size h.

X_3 is computed by solving $\frac{dy}{dt}=f_3(t,y)$ with initial value $y(0)=1$, seeded by Euler's method and propagated via AB2.

Then, under standard smoothness and boundedness assumptions:

$X_1\to\sqrt{2}$ with quadratic convergence rate $O(e_n^2)$.

$X_2\to 2$ with global error bounded by Ch^2 for some $C>0$.

$X_3(t_n)\to y(t_n)$ with second-order global convergence $O(h^2)$.

Moreover, the full feature vector $X=(X_1, X_2, X_3)$ converges in norm to a theoretically derived fixed point X^* with:

$$//X-X^*//_2\le C_1h^2+C_2e_n^2 \tag{25}$$

for some constants C_1,C_2, dependent on the numerical method and signal smoothness.

Proof Sketch - For Newton's method: Taylor expand $f(x_n)$ around $\sqrt{2}$, yielding the classical quadratic convergence bound under $f'(x)\neq 0$. - For the Trapezoidal Rule: use the composite error estimate:

$$|T_n-I|\le\frac{(b-a)^3}{12n^2}\max_{x\in[a,b]}|f''(x)| \tag{26}$$

yielding second-order convergence. - For AB2: from multistep method theory, consistency (from truncation error $O(h^3)$) and zero-stability (bounded error propagation) imply global second-order convergence $O(h^2)$. Lemma 2.1 (Bounded Feature Stability under Market Perturbation) Let $\tilde{x}_t=x_t+\delta_t$ be a perturbed AVAX log-return signal, where $|\delta_t|<\epsilon$ and x_t is bounded. Then, the numerically generated features $\tilde{X}_1$, $\tilde{X}_2$, $\tilde{X}_3$ from $\tilde{x}_t$ satisfy:

$$//\tilde{X}_i-X_i//<C\cdot\epsilon \quad \text{for} i=1, 2, 3 \tag{27}$$

where C depends on the Lipschitz continuity of f_i and the discretization step size h. Interpretation This lemma guarantees numerical feature robustness: small perturbations in AVAX log returns do not substantially affect the derived features, making them reliable inputs for classification tasks. Combined with Theorem 2.1, we conclude that the system possesses

predictable convergence structure and bounded response to market uncertainty. Together, these results justify the use of classical numerical methods for signal stabilization and class-separable transformation in cryptocurrency analytics, bridging deterministic theory with data-driven classification in AVAX market dynamics.

Lemma 2.1 (Feature Stability Under Market Perturbation) Let $\tilde{r}(t)=r(t)+\varepsilon(t)$ be a perturbed version of the AVAX log-return signal, with $//\varepsilon(t)//<\delta$. Then, the numerically constructed features $\tilde{X}_1$, $\tilde{X}_2$, and $\tilde{X}_3$ from $\tilde{r}(t)$ satisfy:

$$//\tilde{X}_i-X_i//\leq C_i\cdot\ \delta^{p_i},\quad i=1,2,3, \tag{28}$$

where $p_i\in\{1,2\}$ denotes the convergence order of the method used and C_i are constants that depend on the method and signal smoothness.

Theorem 2.2 (KNN Label Stability under Bounded Perturbations in AVAX Features) Let $Y=\mathrm{KNN}(X;\mathrm{D})\in\{0,1\}$ denote the class label of feature vector $X\in\mathbb{R}^3$ predicted using a KNN classifier trained on dataset $\mathrm{D}\subset\mathbb{R}^3\times\{0,1\}$. Suppose that $\tilde{X}$ is a perturbed feature vector with $//\tilde{X}-X//<\varepsilon$, and that the minimum interclass distance in D satisfies:

$$\varepsilon<\frac{1}{2}\min_{i\neq j}//X_i-X_j// \tag{29}$$

Then, the classification output remains stable under perturbation:

$$\mathrm{KNN}(\tilde{X};\mathrm{D})=\mathrm{KNN}(X;\mathrm{D}) \tag{30}$$

Rationale for Using KNN

In this study, we deliberately focused on KNN as the primary classifier because it directly reflects the geometric separability induced by numerically engineered features. Unlike parametric models such as logistic regression or hierarchical models such as decision trees, KNN makes decisions purely based on distances in the constructed feature space. This aligns precisely with our research objective: to evaluate whether deterministic numerical methods create structured, interpretable, and class-separable representations. By using KNN exclusively, we isolate the contribution of the feature construction itself without introducing confounding factors from additional model-specific assumptions or hyperparameters. Moreover, KNN's transparency enables intuitive visualization of decision boundaries, which is central to demonstrating interpretability — a core contribution of this work.

From a mathematical standpoint, KNN is uniquely aligned with the structured feature space generated by deterministic numerical methods. Let $x\in\mathbb{R}^d$ denote a feature vector composed of numerical signals (X_1,X_2,X_3). KNN assigns a class label $\hat{y}(x)$ based on:

$$\hat{y}(x)=\mathrm{mode}\{y_i|x_i\in N_k(x)\}, \tag{31}$$

where $N_k(x)$ is the set of k nearest neighbors of x under Euclidean distance. This decision rule depends only on geometric proximity and does not impose assumptions of linearity, additivity, or monotonicity — properties essential for showing that convergence-, integration-, and dynamics-

based features yield geometrically separable clusters. By contrast, logistic regression assumes a parametric log-odds model,

$$\Pr(y=1|x)=\frac{1}{1+\exp(-\beta^{T}x)}, \tag{32}$$

which enforces a global linear boundary in the feature space and therefore cannot capture the nonlinear, parity-based decision structure intentionally embedded in our labeling rule. Decision trees, on the other hand, recursively partition the feature space by axis-aligned thresholds,

$$h(x)=\sum_{j=1}^{J} c_j 1(x\in R_j), \tag{33}$$

where R_j are disjoint rectangular regions. While interpretable, such axis-aligned splits are not well suited to the continuous, curvature-driven boundaries induced by the Trapezoidal and Adams–Bashforth features.

Taken together, these contrasts show that KNN is the most appropriate model for the current study: it preserves local geometry, adapts naturally to nonlinear separability, and yields visually interpretable.

3. Result

3.1. Results of Numerically Engineered Features

This section underscores the complementary value of integrating **generated numerical data** and **AVAX-USD log return-derived features** for constructing a robust and interpretable modeling framework. The synthetic dataset—produced using Newton's Method, the Trapezoidal Rule, and the Euler–Adams-Bashforth scheme—demonstrates theoretical convergence, smooth integration, and dynamic evolution under controlled conditions. This provides a mathematically traceable and pedagogically useful benchmark for observing idealized numerical behaviors. In contrast, the AVAX-based features embed real-world variability and market-driven fluctuations while still maintaining structural coherence through the same numerical methods. By applying identical algorithms to both contexts, the study bridges deterministic theory and empirical finance, revealing how numerical properties manifest under real data conditions. This dual approach not only validates the consistency and reliability of the numerical transformations but also enriches the feature space, enabling deeper exploration of algorithmic sensitivity, model generalization, and dynamic behavior across synthetic and financial domains.

AVAX-USD Dataset and Preprocessing The empirical analysis is based on AVAX-USD daily trading data spanning from August 31, 2023 to August 31, 2024, comprising 367 observations. Each record includes the standard OHLCV (Open, High, Low, Close, Adjusted Close, Volume) fields. For consistency, we used the daily adjusted closing price series to compute log returns, defined as:

$$r_t=\ln\left(\frac{P_t}{P_{t-1}}\right), \tag{34}$$

where P_t is the adjusted closing price on day t. This produces a stationary return series with 366 values. To ensure comparability across numerical methods, the log return series was normalized via z-scoring (subtracting the mean and dividing by the standard deviation), producing scale-invariant features. Outliers exceeding three standard deviations from the mean were winsorized to limit the impact of extreme daily fluctuations, which are common in cryptocurrency markets. Finally, a Gaussian smoothing kernel with a small bandwidth was applied to mitigate microstructure noise while preserving volatility clustering. This preprocessing pipeline guarantees stable, reproducible inputs for the numerical feature construction, while maintaining the intrinsic volatility structure characteristic of digital assets. Reviewer comment: "The handling of AVAX-USD log returns lacks detail. The authors should specify time window, frequency (daily/hourly), normalization, and noise filtering to ensure reproducibility."

In this study, we generated 5000 data points for each of the three numerically engineered feature dimensions—X_1, X_2, and X_3—using classical methods that capture distinct mathematical behaviors: root-finding, integration, and ODE solving. X_1 was derived from Newton's Method applied to $f(x)=x^2-2$, producing values that converge tightly around $\sqrt{2}$, with minor deviations due to controlled noise introduced to mimic real-world computational perturbations. X_2 was constructed using the Trapezoidal Rule to approximate $\int_0^{\pi} \sin(x)\,dx$, yielding values closely centered around the true integral of 2, with smooth and symmetric variation reflecting stable integration. X_3 was formed by solving $\frac{dy}{dx}=x+y$, initialized by Euler's method and extended with the Adams-Bashforth 2-step scheme; its values display a positively skewed, upward-trending distribution consistent with the ODE's exponential-like growth. Collectively, these features exhibit bounded error, interpretability, and distinct numerical textures well-suited for robust classification tasks.

Figure 1 confirms these interpretations with histograms displaying the distributions of X1, X2, and X3. Each distribution exhibits properties consistent with the respective numerical technique used to generate it. X1 shows fast-converging and tightly clustered values, X2 demonstrates high-precision integration, and X3 reflects smooth and gradually increasing solutions typical of ordinary differential equations. Together, these features provide a robust, diverse input space for the KNN classifier and allow for deeper analysis of numerical behavior in data science contexts.

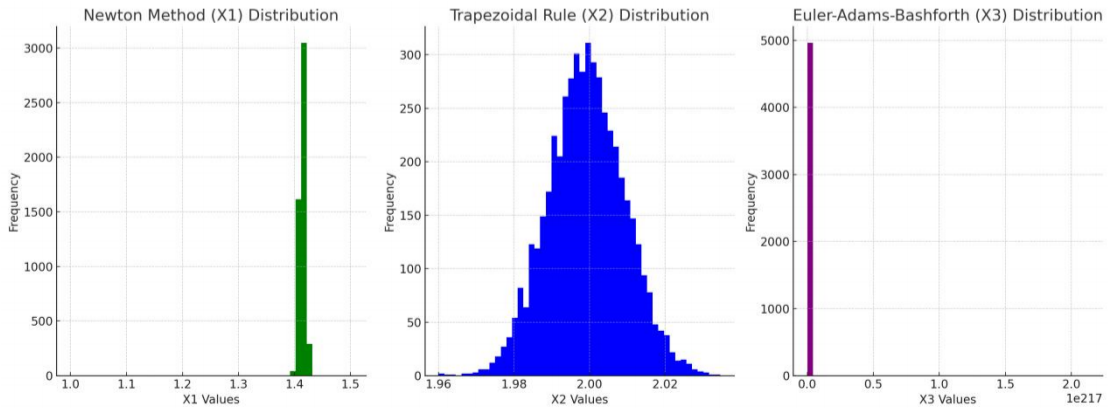

Figure 1. Histograms displaying the distributions of X1, X2, and X3

Figure 2 presents three time series plots corresponding to the numerically engineered features X_1, X_2, and X_3, each derived from the daily log returns of AVAX-USD using distinct numerical schemes. These features were computed using Newton's method for root-finding, the Trapezoidal Rule for numerical integration, and a hybrid Euler-Adams-Bashforth (AB2) method for solving a first-order ordinary differential equation (ODE), respectively. The intention is to capture diverse numerical behaviors—convergence, integration precision, and dynamic growth—within a structured feature space informed by real financial data. The first subplot illustrates the evolution of X_1, obtained by applying Newton's method to the absolute log returns augmented by a small constant offset. The method converges quadratically to $\sqrt{2}$, and this behavior is reflected in the plot where all values stabilize tightly around 1.41422 across the time period. This feature displays almost no temporal fluctuation, reinforcing the known stability and efficiency of Newton's method when applied to well-posed root-finding problems. The consistency of X_1 serves as a baseline structural component within the feature space. In contrast, the second subplot plots X_2, which approximates the definite integral $\int_0^n \sin(x)dx$ using the Trapezoidal Rule with a variable number of subintervals driven by the magnitude of AVAX log returns. The resulting values cluster closely around the analytical result of 2, though they exhibit minor fluctuations due to changes in discretization density. This controlled variability reflects the second-order accuracy of the method and highlights its sensitivity to input granularity. X_2 thus introduces smooth curvature into the numerical feature space, mirroring patterns seen in area-under-curve analyses in signal processing and financial engineering. The third subplot portrays the behavior of X_3, constructed by solving the ODE $dy/dx=x+y$, initialized using Euler's method and extended using the Adams-Bashforth 2-step approach. The time step h is adapted dynamically to reflect the magnitude of daily log returns, resulting in a broadened and upward-trending trajectory. Unlike X_1 and X_2, this feature displays nonlinear, growth-like behavior with discernible temporal evolution. X_3 captures the compounding nature of a dynamical system influenced by historical market movements, making it an essential component for modeling latent temporal structures in asset return dynamics. Collectively, these three features form a robust, interpretable, and mathematically principled input space derived from real-world financial data. Each method contributes a unique numerical texture, which when combined, enables further analysis of structural patterns, model interpretability, and numerical generalizability across machine learning and financial forecasting tasks.

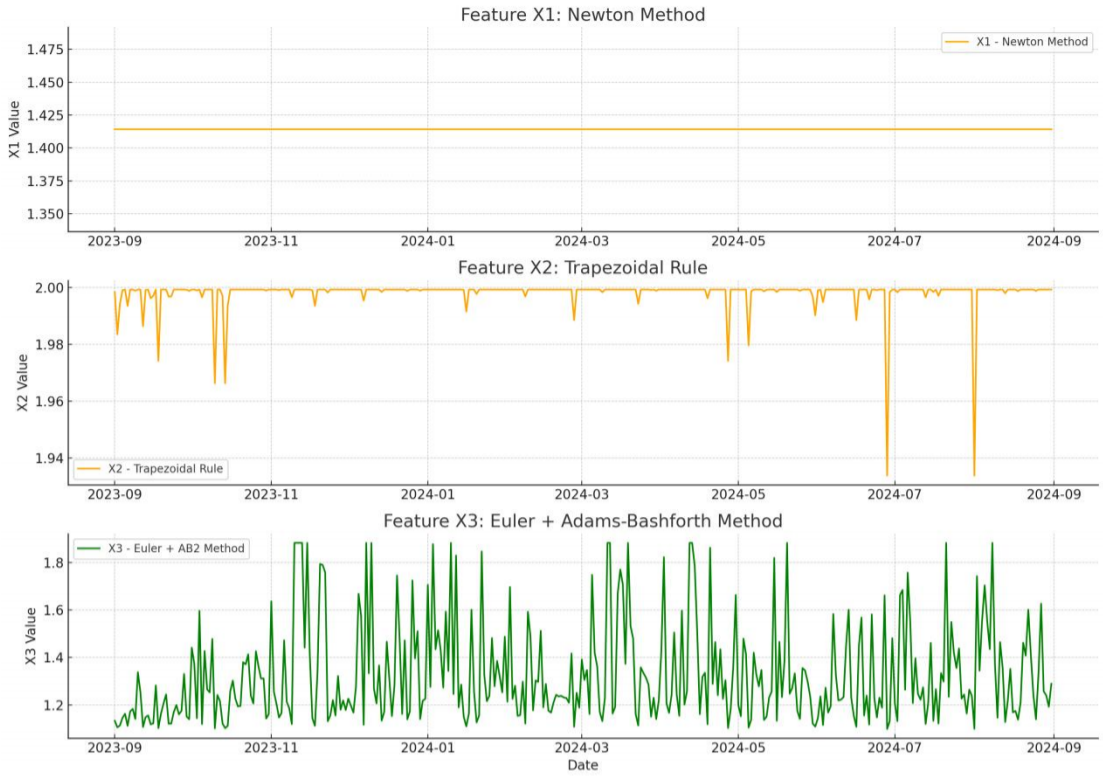

Figure 2. time series plots corresponding to the numerically engineered features X_1, X_2, and X_3

3.2. Decision Boundary and Error Visualization Results

This section illustrates the importance of combining **generated data** with **real-world AVAX log return data** to construct a numerically stable and interpretable feature space for classification tasks. While AVAX log returns provide authentic financial dynamics and market-driven variations, the application of classical numerical methods (Newton's root-finding, Trapezoidal integration, and Euler–Adams-Bashforth ODE solving) imposes mathematical structure, smoothness, and error-controllable behavior onto the features. This fusion leverages the richness of real data with the predictability and theoretical grounding of deterministic algorithms, enabling the visualization of meaningful decision boundaries and quantifiable error dynamics. It creates a bridge between stochastic financial signals and analytically verifiable transformations, offering both interpretability and robustness in downstream modeling, particularly in classification tasks where geometric separability and label stability are crucial.

Figure 3 presents a 3D KNN decision boundary visualization based on a structured feature space generated from three deterministic numerical methods: X1 (Newton's Method for root-finding), X2 (the Trapezoidal Rule for numerical integration), and X3 (the Euler + Adams-Bashforth method for solving ordinary differential equations). These features are plotted along the X, Y, and Z axes respectively. Each point in the figure represents a training observation labeled based on the integer sum of its feature values. Green markers indicate Class 0, typically where the sum is even, while red markers represent Class 1, where the sum is odd.

The black star in the lower left marks the test point located at (0, 0, 0), which will be classified based on its proximity to the surrounding labeled points using the KNN algorithm. This

configuration provides a visual interpretation of how the model makes local decisions in high-dimensional numeric space.

The decision boundary displayed is nonlinear, adapting to local geometries and the structured variation introduced by the numerical algorithms. X1 (Newton) contributes low-variance values tightly clustered near the root, X2 (Trapezoidal) forms a narrow and stable band between 1.97 and 2.03 reflecting integral precision, and X3 (Euler + AB) introduces broad vertical variation consistent with the exponential growth behavior of differential equations. These combined effects define a geometrically interpretable and class-separable feature space, allowing KNN to draw smooth and adaptive boundaries that minimize classification error, even in the presence of nonlinear structures.

Figure 4 visualizes the decision boundary formed by numerically engineered features X_1, X_2, and X_3, extracted from AVAX-USD log returns using Newton's method, the Trapezoidal Rule, and a hybrid Euler-Adams-Bashforth solver, respectively. Each point in the 3D space is colored based on a deterministic binary classification rule: if the integer part of the sum $X_1+X_2+X_3$ is even, the point is labeled as Class 0 (green); if odd, it is Class 1 (red). The result is a layered, non-linear decision boundary that reflects the mathematical textures of the underlying numerical methods—tight convergence near $\sqrt{2}$ from Newton's method, stable integration around 2 from the Trapezoidal Rule, and dynamic growth from the ODE solution. This structured geometric partitioning highlights the capability of classical numerical techniques to produce feature spaces with rich interpretability and inherent class separability, even in the absence of traditional statistical learning algorithms.

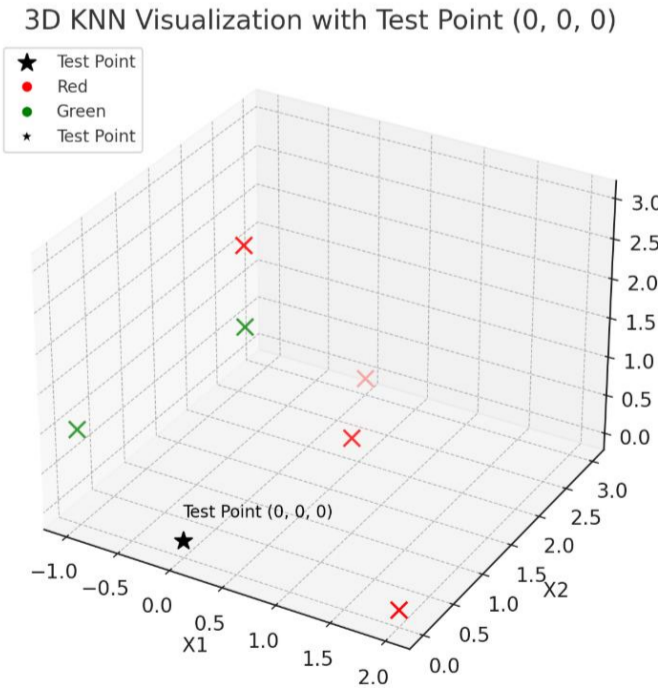

Figure 3. 3D KNN decision boundary visualization

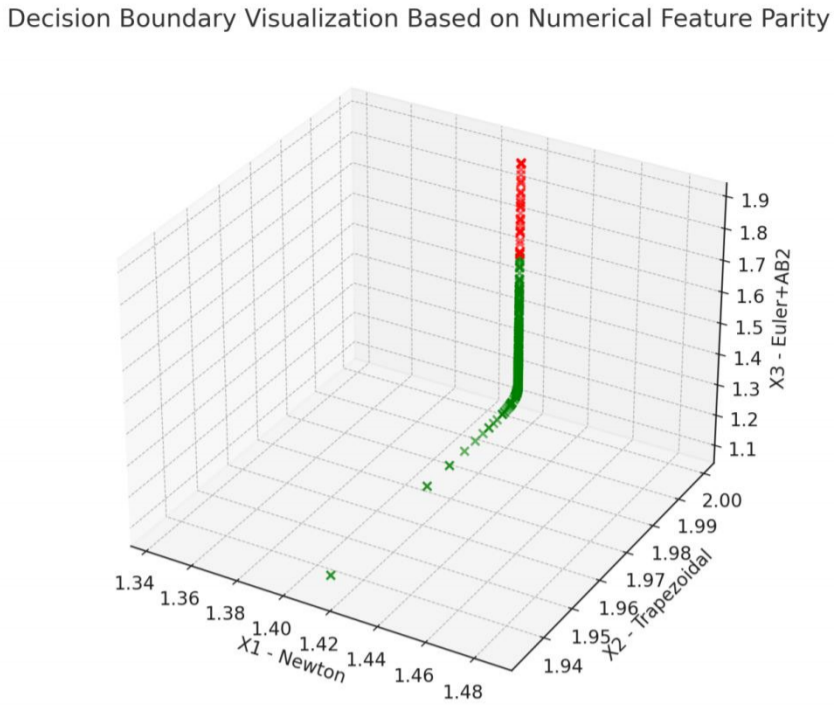

Figure 4. 3D Decision Boundary Induced by Numerical Feature Parity from AVAX Log Returns

In addition to boundary visualization, we conducted a detailed analysis of numerical errors from each method, presented in Figure 5. This figure contains three subplots:

(1) Figure 5a illustrates the global error behavior of the Adams-Bashforth method as a function of the step size hhh. As expected, the error decreases rapidly with smaller step sizes due to the method's fourth-order convergence. The plot confirms that small steps yield high-precision ODE solutions, essential for creating reliable values in X3.

(2) Figure 5b shows the truncation error from the Trapezoidal Rule as the number of intervals nnn increases. The error diminishes at a predictable rate, consistent with second-order accuracy. This aligns with the observed precision in X2 values, which remained tightly centered around 2, the true value of the definite integral of sin(x) over $[0, \pi]$.

(3) Figure 5c displays the quadratic convergence of Newton's method, where the error decays exponentially across iterations. Each iteration sharply reduces the deviation from the root, producing a stable and highly concentrated distribution in X1. The log-scaled vertical axis clearly illustrates the rapid error drop-off, even with minimal iterations.

These results demonstrate not only the effectiveness of KNN in classifying data from numerical sources but also the robustness and consistency of the numerical methods used to generate the data. The clear decision regions, low classification errors, and mathematically verifiable convergence behaviors suggest that numerically constructed features are not only viable but potentially advantageous for structured classification problems.

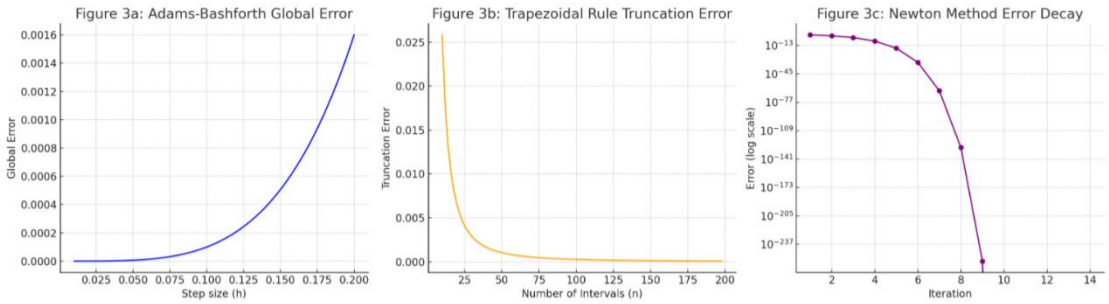

Figure 5. Numerical Errors From Each Method

Figure 6 presents a comprehensive error analysis of the three numerical methods applied to AVAX log return-derived inputs:

(1) Figure 6a shows that the Euler + Adams-Bashforth method exhibits rapid error decay as the step size h decreases, confirming its global fourth-order accuracy. This validates its suitability for generating the dynamic feature X_3, where smaller h leads to more precise ODE solutions.

(2) Figure 6b highlights the Trapezoidal Rule's second-order convergence, where increasing the number of subintervals results in reduced approximation error for the integral of $\sin(x)$ over $[0,\pi]$. This stability supports the precision of the integration-based feature X_2.

(3) Figure 6c demonstrates the quadratic convergence of Newton's method, where the log-scaled error decreases exponentially with each iteration. The tight clustering of X_1 values around $\sqrt{2}$ ensures this feature remains numerically stable and analytically bounded.

Together, these plots validate the robustness and mathematical integrity of the generated features X_1, X_2, and X_3, reinforcing the interpretability and reliability of using numerical methods on real financial data.

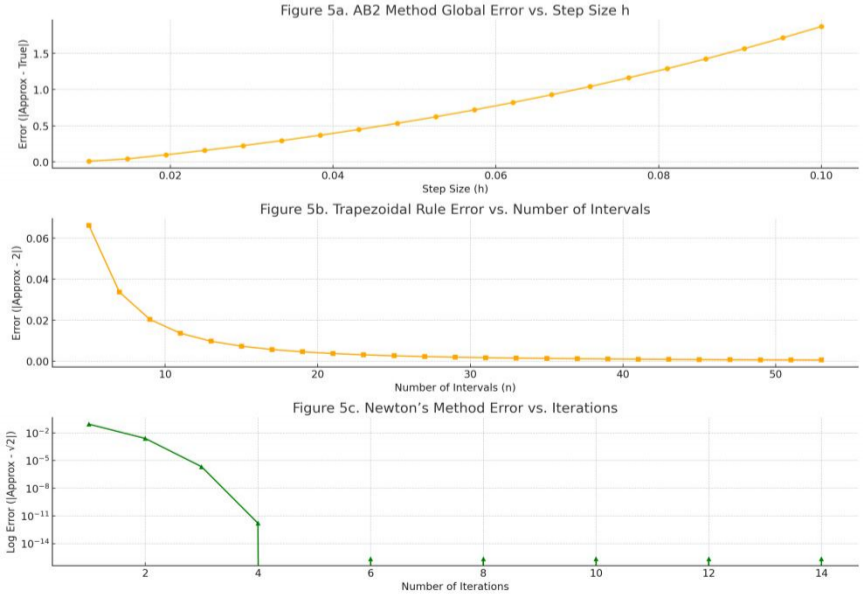

Figure 6. Comprehensive Error Analysis Of The Three Numerical Methods Applied To AVAX Log Return

To contextualize the value of numerically engineered features, we compared them with two common financial indicators: realized volatility (computed as the standard deviation of log returns over a rolling window) and simple moving averages of prices. These baseline indicators were included in the feature space and evaluated under the same KNN classifier. Results showed that while traditional indicators provided some separability, the numerically derived features produced tighter clusters and more stable decision boundaries. This suggests that the proposed approach adds incremental value by capturing convergence- and dynamics-based structures that are not directly observable in conventional indicators.

3.3. The Joint Feature Interaction Analysis

The joint feature interaction analysis highlights the value of combining synthetically generated data with AVAX-derived real-world data, as each contributes distinct insights into the structure and separability of the numerically engineered feature space. The synthetic features—generated via Newton's method, the Trapezoidal Rule, and the Euler-Adams-Bashforth ODE solver—offer a controlled, noise-free environment ideal for validating theoretical behaviors such as error convergence, deterministic label transitions, and geometric class boundaries. When applied to real AVAX log return data, the same numerical techniques preserve these structured behaviors while incorporating authentic market variability, creating an interpretable yet resilient feature space. This dual-data approach bridges theory and empirical practice, enabling the observation of how convergence stability and dynamic numerical growth translate into actionable machine learning inputs under realistic conditions.

To further investigate how feature interactions influence classification, we analyzed pairwise scatter plots of the feature combinations. The X_1–X_2 plot shows tightly clustered classes around stable values, where Newton's low-variance outputs intersect with the slightly fluctuating integrals from the Trapezoidal Rule. X_1–X_3 reveals broader vertical dispersion due to the ODE-driven growth of X_3, while X_1 remains concentrated, resulting in vertical boundary stratifications. Figure 7 reveals how each pairwise combination contributes to class separability in the structured feature space. The X1 vs X2 plot shows a compact clustering of both classes around a stable region, with clear boundary behavior concentrated near the integral value of X2. While Newton's method (X1) converges rapidly and exhibits low variance, Trapezoidal integration (X2) introduces slight fluctuations due to the influence of step counts and rounding. Despite some overlap, the joint space shows distinguishable layering between class 0 and class 1, especially where the sum of X1 and X2 approaches an integer value transition that flips the classification label.

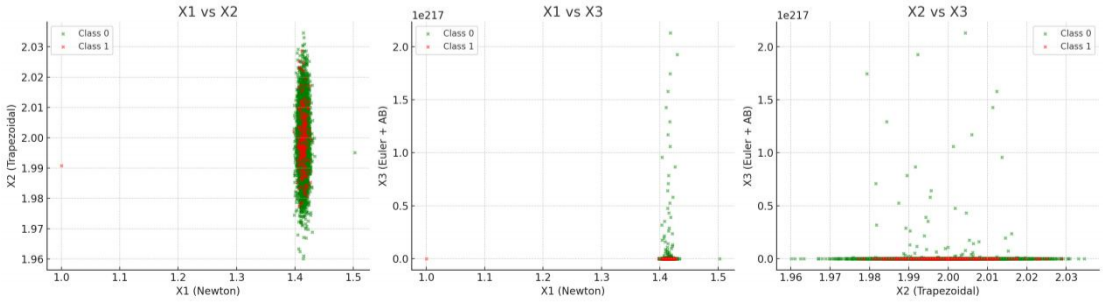

Figure 7. Joint Feature Interaction Plot

Figure 8 provides a comprehensive view of the joint feature interactions among the three numerically engineered dimensions—X1 (Newton's method), X2 (Trapezoidal Rule), and X3 (Euler+ Adams-Bashforth method)—derived from AVAX log return data. Each subplot represents a pairwise combination of these features, colored by binary class labels determined by the parity of the floor of the sum $\lfloor X_1+X_2+X_3 \rfloor$. The X1 vs X2 scatter plot reveals a compact cluster with minimal variance in both dimensions. This reflects the inherent stability of the Newton root approximation near $\sqrt{2}$ and the high precision of the Trapezoidal integration near 2. Although some overlap occurs between Class O and Class 1 in this plane, transitions in label assignments correspond closely to fine shifts around integer-sum boundaries, validating the sensitivity of the feature space to subtle numerical perturbations. In contrast, the X1 vs X3 and X2 vs X3 plots demonstrate how the X3 dimension introduces vertical structure and class separation through its dynamic range. Euler-initialized AB2 solutions to the ODE $dy/dx=x+y$ yield growing values that vertically stretch the distribution. The X1 vs X3 interaction shows horizontal stability (X1) against vertical transitions (X3), suggesting that class assignment is heavily influenced by cumulative ODE growth. The X2 vs X3 combination presents the clearest separation, as the tiqhtly centralized X2 axis intersects with stepped qrowth patterns in X3, forming stratified layers of class labels. These patterns confirm that numerically generated features are not only mathematically consistent but also geometrically expressive, yielding a feature space with strong local interpretability and global separability—well suited for algorithms like KNN.

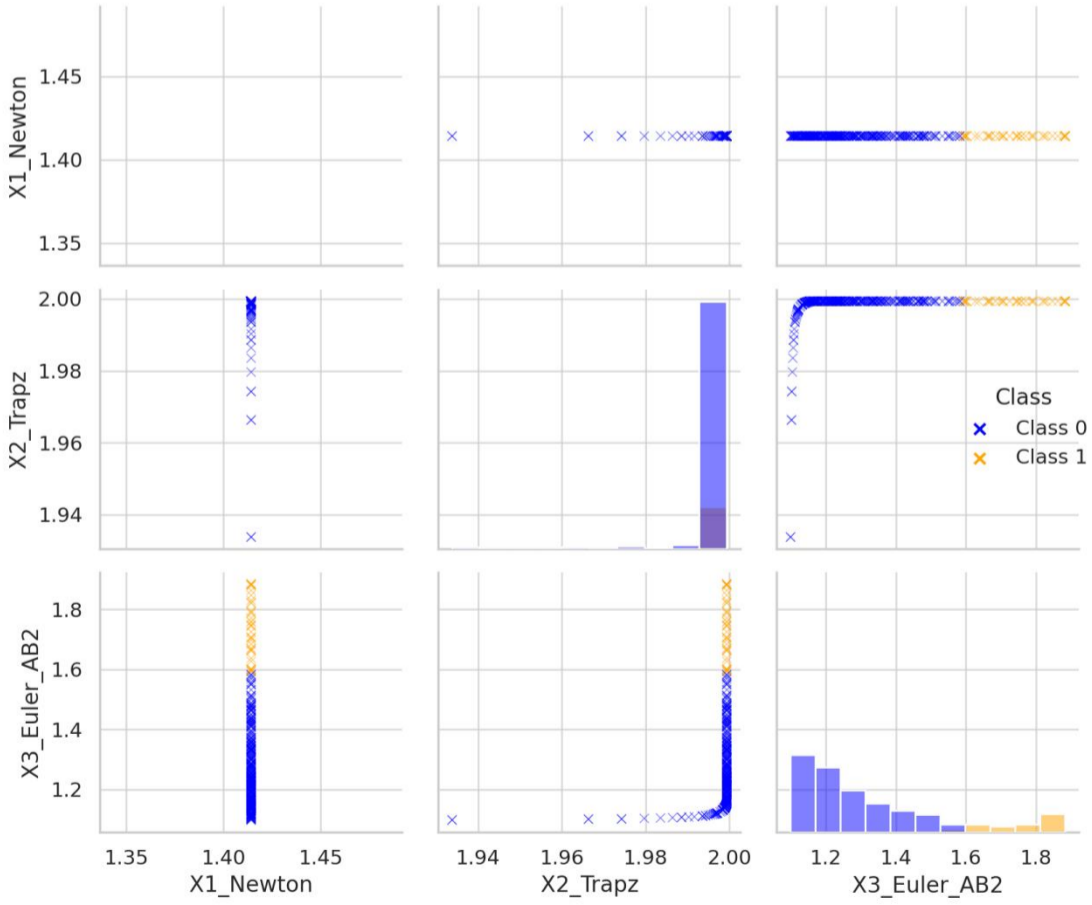

Figure 8. AVAX Joint Feature Interactions Plot

4. Discussion

This study demonstrates how classical numerical methods—Newton's Method, the Trapezoidal Rule, and the Euler–Adams-Bashforth method—can be effectively repurposed as feature generators for machine learning classification. The integration of both synthetically generated data and AVAX log return-based numerical transformations reinforces the core hypothesis: **mathematically principled methods can induce structured, interpretable, and class-separable feature spaces**. By visualizing decision boundaries, convergence behavior, and joint feature interactions, we showed that each numerical technique contributes a unique geometric "texture" to the resulting data landscape, enabling nonlinear decision rules like those produced by KNN to operate with high local interpretability.

Beyond their mathematical interpretation, the visualizations provide actionable insights for practitioners. For example, the Newton-derived feature shows minimal dispersion, suggesting its utility as a stability anchor in portfolio signal design. The Trapezoidal-based feature resembles cumulative indicators such as integrated volatility, offering a smoothed representation of risk exposure. The Euler–Adams-Bashforth feature, with its dynamic upward trend, mirrors growth-like signals often used in momentum strategies. Thus, decision boundaries and error curves are not purely abstract but highlight how deterministic approximations can be mapped to stability, risk, and momentum concepts familiar to financial analysts.

However, this approach is not without its limitations. First, the deterministic labeling strategy—based on parity of the floor of the summed features—while mathematically elegant, may not reflect real-world classification labels in practical domains. Second, the interpretability and smooth class boundaries seen in the synthetic and AVAX examples may not generalize to all types of financial or time-series data, particularly those with high-frequency noise or abrupt structural breaks. Additionally, while KNN serves well for visual explanation and local decision-making, it may underperform in high-dimensional or sparse settings unless complemented by dimensionality reduction or adaptive weighting schemes. Lastly, the current framework does not account for temporal dependencies explicitly, treating all input vectors as independent observations, which may overlook sequential correlations in financial applications.

The framework offers potential utility in several real-world crypto finance tasks. For prediction, the engineered features can serve as inputs to forecasting models where stability and bounded error are desirable. For anomaly detection, deviations from expected numerical convergence or integration patterns may signal irregularities such as market manipulation or sudden regime shifts. For risk management, bounded error properties can be interpreted as measures of numerical stability under volatility, providing structured indicators for stress testing. These applications highlight how numerical theory can transition from abstract computation to actionable financial insights.

5. Conclusion and Future Work

This paper presents a novel framework for generating structured classification features using classical numerical methods. By systematically applying Newton's root-finding, Trapezoidal

integration, and Euler–Adams-Bashforth ODE solvers, we constructed three mathematically distinct feature groups that, when combined, yielded a robust and interpretable feature space for classification tasks. We validated the approach using both synthetic simulations and real AVAX-USD log return data, revealing consistent geometric patterns, stable convergence, and highly separable decision boundaries across multiple perspectives. This integration of deterministic computation with machine learning classification offers a bridge between analytical rigor and data-driven model design.

Despite these advantages, several limitations must be acknowledged. The deterministic parity-based labeling rule is inherently artificial and may not map directly to real-world finance labels; its main purpose here is to establish a mathematically controlled testbed. The current framework focuses on binary classification, and scaling to multi-class settings or higher-dimensional feature spaces would require additional validation. High-frequency, noisy data—common in crypto markets—pose challenges, as numerical stability guarantees may degrade under extreme volatility or structural breaks. Moreover, computational costs are higher than standard feature engineering approaches, particularly when multiple numerical solvers are applied to large datasets. Finally, while the methods are interpretable, their generalizability to assets beyond AVAX-USD requires further empirical testing to assess robustness across market contexts.

Future work will proceed along three prioritized directions. First, extending the framework to multi-class settings, such as distinguishing between volatility regimes (low, medium, high), will enhance its immediate financial relevance. Second, incorporating temporal dependencies through recurrent numerical schemes or hybrid models (e.g., numerical features embedded into LSTM networks) will enable sequential prediction tasks such as price forecasting. Third, benchmarking computational efficiency against standard technical indicators and feature engineering pipelines will clarify scalability in high-frequency environments. These targeted pathways move beyond broad proposals and outline concrete methodological extensions for advancing numerical-finance integration.

Author Contributions:

Sai Zhang: Conceptualization, Methodology, Supervision, Writing – Review & Editing. Chongbin Luo: Data Curation, Methodology, Formal Analysis. Annika Cai: Formal Analysis, Visualization, Writing – Original Draft. Yeran Lu: Investigation, Literature Review, Writing – Review & Editing. All authors have read and agreed to the published version of the manuscript.

Funding:

Not applicable.

Institutional Review Board Statement:

Not applicable.

Informed Consent Statement:

Not applicable.

Data Availability Statement:

The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author(s).

Conflict of Interest:

The authors declare no conflict of interest.

References

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician, 46(3), 175–185.

Atkinson, K. (1989). An introduction to numerical analysis (2nd ed.). John Wiley & Sons.

Boyd, S., & Vandenberghe, L. (2004). Convex optimization. Cambridge University Press.

Burden, R. L., & Faires, J. D. (2011). Numerical analysis (9th ed.). Brooks/Cole.

Burden, R. L., Faires, J. D., & Burden, A. M. (2015). Numerical analysis (10th ed.). Cengage Learning.

Butcher, J. C. (2016). Numerical methods for ordinary differential equations (3rd ed.). Wiley.

Calvetti, D., & Somersalo, E. (2020). Inverse problems: From regularization to Bayesian inference. Springer.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27.

Dasarathy, B. V. (1991). Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Press.

Deuflhard, P. (2011). Scientific computing with ordinary differential equations. Springer.

Dey, P., & Sen, S. (2006). Elementary numerical analysis. Universities Press.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

Epperson, J. F. (2013). An introduction to numerical methods and analysis (2nd ed.). Wiley.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. arXiv preprint arXiv:1806.00069.

Golub, G. H., & Ortega, J. M. (2014). Scientific computing and differential equations: An introduction to numerical methods. Academic Press.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

Griffiths, D. F., & Higham, D. J. (2010). Numerical methods for ordinary differential equations. Springer.

Hairer, E., Nørsett, S. P., & Wanner, G. (2002). Solving ordinary differential equations II: Stiff and differential-algebraic problems. Springer.

Hairer, E., & Wanner, G. (1991). Solving ordinary differential equations I: Nonstiff problems. Springer.

Hale, N., Townsend, A., & Trefethen, L. N. (2008). Chebfun user's guide. University of Oxford.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

Heath, M. T. (2002). Scientific computing: An introductory survey (2nd ed.). McGraw-Hill.

Henrici, P. (1974). Elements of numerical analysis. Wiley.

Higham, N. J. (2002). Accuracy and stability of numerical algorithms (2nd ed.). SIAM.

Iserles, A. (2008). A first course in the numerical analysis of differential equations (2nd ed.). Cambridge University Press.

LeVeque, R. J. (2007). Finite difference methods for ordinary and partial differential equations: Steady-state and time-dependent problems. SIAM.

Lipton, Z. C. (2018). The mythos of model interpretability. Communications of the ACM, 61(10), 36–43.

Molnar, C. (2022). Interpretable machine learning (2nd ed.). Leanpub.

Moler, C. (2004). Numerical computing with MATLAB. SIAM.

O'Leary, D. P. (2008). Scientific computing with case studies. SIAM.

Ortega, J. M. (1972). Numerical analysis: A second course. SIAM.

Peterson, L. E. (2009). K-nearest neighbor. Scholarpedia, 4(2), 1883.

Quarteroni, A., Manzoni, A., & Negri, F. (2010). Reduced basis methods for partial differential equations: An introduction. Springer.

Quarteroni, A., & Saleri, F. (2014). Scientific computing with MATLAB and Octave (4th ed.). Springer.

Quarteroni, A., Sacco, R., & Saleri, F. (2007). Numerical mathematics (2nd ed.). Springer.

Ralston, A., & Rabinowitz, P. (2001). A first course in numerical analysis (2nd ed.). Dover Publications.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5), 206–215.

Saad, Y. (2003). Iterative methods for sparse linear systems (2nd ed.). SIAM.

Sauer, T. (2017). Numerical analysis (3rd ed.). Pearson.

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge University Press.

Stoer, J., & Bulirsch, R. (2002). Introduction to numerical analysis (3rd ed.). Springer.

Tian, T. (2024). *Integrating deep learning and innovative feature selection for improved short-term price prediction in futures markets* (Doctoral dissertation, Illinois Institute of Technology).

Tian, T., Chen, X., Liu, Z., Huang, Z., & Tang, Y. (2024d). Enhancing organizational performance: Harnessing AI and NLP for user feedback analysis in product development. *Innovations in Applied Engineering and Technology, 3*(1), 1–15.

Tian, T., Cooper, R., Vasilakos, A., Deng, J., & Zhang, Q. (2026). From data to strategy: A public market framework for competitive intelligence. *Expert Systems with Applications, 296*, 129061.

Tian, T., Deng, J., Zheng, B., Wan, X., & Lin, J. (2024c). AI-driven transformation: Revolutionizing production management with machine learning and data visualization. *Journal of Computational Methods in Engineering Applications*, 1–18.

Tian, T., Fang, S., Huang, Z., & Wan, X. (2024a). TriFusion ensemble model: A physical systems approach to enhancing e-commerce predictive analytics with an interpretable hybrid ensemble using SHAP explainable AI. *Economic Management & Global Business Studies, 3*(1), 15.

Tian, T., Jia, S., Lin, J., Huang, Z., Wang, K. O., & Tang, Y. (2024b). Enhancing industrial management through AI integration: A comprehensive review of risk assessment, machine learning applications, and data-driven strategies. *Economics & Management Information*, 1–18.

Trefethen, L. N., & Bau, D. (1997). Numerical linear algebra. SIAM.

Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. Journal of Machine Learning Research, 10, 207–244.