Financial Strategy and Management Reviews, 2025, 1(2), 11-31
https://doi.org/10.71204/f1dbs667

Optimizing Precision in Volatile Crypto Markets with E²-Fuse: An Energy-Based Ensemble for Bitcoin Forecasting

Sai Zhang [1,*], Yeran Lu [2], Chongbin Luo [3], Qifan Wei [2], Xunyi Liu [2], Yiyun Zheng [2]

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

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

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

*** Correspondence:**

Sai Zhang

saizhang086@gmail.com

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

Abstract

Bitcoin (BTC) has become a major financial asset, attracting significant attention from both institutional and individual investors. However, its extreme price volatility — driven by macroeconomic factors, regulatory changes, and investor sentiment—makes accurate forecasting challenging, as traditional models and machine learning approaches struggle to effectively capture its complex dynamics. Addressing these limitations, we introduce E²-Fuse, a novel energy-minimizing ensemble framework specifically designed for Bitcoin price prediction. E²-Fuse conceptualizes each base model's mean squared error (MSE) as individual "energy" components and employs gradient-based optimization techniques to minimize the total system energy, thereby enhancing predictive accuracy. This physics-inspired methodology systematically integrates multiple advanced ML predictors, ensuring that the ensemble adapts dynamically to BTC's volatile market conditions. Our empirical evaluations demonstrate that E²-Fuse achieves a Normalized RMSE (Accuracy Score) of 0.9969, indicating near-perfect prediction accuracy, and consistently outperforms single models and traditional ensemble methods by achieving lower prediction errors and enhanced robustness against market fluctuations. The framework's model-agnostic nature allows for seamless incorporation of diverse algorithms, including neural networks and gradient boosting methods, broadening its applicability across various high-volatility domains beyond finance. By formalizing ensemble weight determination through energy minimization, E²-Fuse not only advances the state-of-the-art in BTC forecasting but also offers a versatile and theoretically grounded approach for optimizing predictive performance in complex, dynamic environments. This study underscores the potential of integrating advanced ML techniques within an optimization-driven ensemble framework, paving the way for more reliable and adaptive financial forecasting models.

Keywords: Bitcoin Price Prediction; Ensemble Learning; Machine Learning; Energy Minimization; Gradient-Based Optimization; Financial Forecasting

1. Introduction

Bitcoin (BTC) has rapidly evolved from a niche digital currency to a prominent financial asset, attracting both institutional and individual investors. Its price is notoriously volatile, shaped by macroeconomic factors, regulatory actions, and investor sentiment—making BTC forecasting a formidable task. Traditional time-series models (e.g., ARIMA, GARCH) often struggle with the market's non-stationary nature (Tsay, 2005), while machine learning (ML) approaches such as Random Forest (RF) and Support Vector Machines (SVM) have demonstrated promise in handling nonlinearities and capturing complex patterns (Goodfellow et al., 2016; Vapnik, 1995). However, no single algorithm can account for all the intricate drivers of BTC price variations, and many ensemble methods rely on simple averaging or stacking mechanisms that do not explicitly optimize each model's contribution (Dietterich, 2000).

Recent advancements have begun to address these gaps. For instance, Tian et al. (2026) introduced a signal-based strategic intelligence framework for dynamic financial modeling using public market data. Similarly, Tian (2024) presented deep learning and feature selection techniques for futures price forecasting, and Tian et al. (2024) explored ensemble and AI-driven methodologies in e-commerce and industrial production contexts. These studies affirm the importance of blending multiple models under a structured optimization regime.

In this paper, we introduce E^2-Fuse, an energy-minimizing ensemble that treats each base model's error as a distinct "energy" and uses gradient-based techniques to optimize the system's total energy. By combining advanced ML predictors within a physics-inspired framework, E^2-Fuse achieves robust forecasting performance and effectively handles BTC's inherent price fluctuations. This approach builds on recent advances in ensemble learning (Rokach, 2010) and optimization-driven model aggregation (Zhou, 2012), offering a novel method for integrating diverse predictive models in a systematic and theoretically grounded way.

1.1. Challenges

Bitcoin markets experience frequent regime shifts and extreme volatility, which renders static models less effective as market dynamics evolve (Cheah & Fry, 2015; Kristoufek, 2015). Prices can be influenced by an array of indicators—technical signals, trading volumes, macroeconomic events, and even social media sentiment—creating complex interactions that single models may fail to capture adequately (Shen et al., 2020). Moreover, methods like RF or SVM excel under different conditions; hence, relying on a single predictor carries the risk of substantial error if market behavior falls outside that model's strengths (Vapnik, 1995; Breiman, 2001).

Prior work by Tian et al. (2024) in production management using AI and by Tian et al. (2024) in product development feedback analysis emphasizes the importance of interpretability, variance reduction, and dynamic error handling in volatile or noisy domains.

1.2. The Need for This Method

Given the above challenges, it is clear that a robust BTC forecasting framework must harness multiple advanced ML algorithms, capitalize on their complementary strengths, and explicitly optimize an objective function tied to predictive accuracy (Kuncheva, 2004; Polikar, 2006). Such a framework can more effectively adapt to volatile market conditions and varying model performance across time. By judiciously fusing multiple models, we can moderate each learner's limitations, exploit different inductive biases, and reduce overall variance in the final predictions (Opitz & Maclin, 1999).

Tian et al. (2024a) demonstrated a TriFusion ensemble using SHAP-based explainability, underscoring how diverse models can be optimized and interpreted jointly—an insight directly motivating the energy-optimization architecture proposed here.

1.3. Proposed Solution: E^2-Fuse

In response to these needs, we propose E^2-Fuse, an energy-minimizing ensemble that interprets each base model's mean squared error (MSE) as its "energy." The total energy is thus a weighted sum of these errors, and we employ quasi-Newton optimization methods (e.g., BFGS) to find the weight vector =(w1,w2,...) that minimizes this total(Nocedal & Wright, 2006; Fletcher, 1987).Unlike naïve ensemble techniques, this strategy systematically searches the weight space for the global minimum of MSE, ensuring that the final predictions benefit from each model's unique expertise (Rokach, 2010). While this study focuses on combining Random Forest and SVM, the framework is compatible with hybrid model pipelines, such as those seen in Tian et al. (2023), who explored dual adversarial learning for fair classification.

1.4. Novelty

The novelty of E^2-Fuse lies in its fusion of advanced machine learning techniques with a physics-inspired energy minimization framework, offering a transformative approach to ensemble learning for high-volatility time-series data like Bitcoin (BTC). Unlike traditional ensemble methods that rely on post-hoc rules or heuristic weighting, E^2-Fuse reimagines error minimization as a core optimization problem, akin to finding a low-energy equilibrium in a physical system (Nocedal & Wright, 2006; Boyd & Vandenberghe, 2004). This paradigm provides a mathematically rigorous foundation for handling inter-model error interactions, leveraging convexity to ensure that the ensemble's performance is never worse than its best constituent model, and often surpasses it (Breiman, 2001). By redefining ensemble blending as a systematic search for the global minimum of model error, E^2-Fuse introduces a robust and theoretically grounded methodology that enhances predictive accuracy in highly dynamic markets (Kuncheva, 2004).

This energy-based perspective treats each base model's error as a distinct "energy" contribution, leveraging quasi-Newton algorithms (e.g., BFGS) to systematically adjust weights and achieve an energy-optimal state (Fletcher, 1987). This approach is particularly effective in managing diverse error correlations between models, ensuring that the ensemble capitalizes on complementary patterns captured by methods like Random Forest (RF) and Support Vector Machines (SVM). While this study focuses on RF and SVM, the framework seamlessly accommodates more advanced algorithms, such as neural networks and gradient boosting methods, making it adaptable to various domains (Schmidhuber, 2015). By combining machine learning with energy-inspired optimization, E^2-Fuse provides a computationally feasible and conceptually elegant solution to a longstanding challenge in financial forecasting: integrating multiple models to enhance robustness and accuracy. This strategy complements recent works in data-driven strategy modeling, such as the SF-HRP portfolio rebalancing and Gram-Schmidt orthogonalization presented by Tian et al. (2026), and reveals how multi-model interactions can be constrained and optimized in real time.

1.5. Broader Implications & Significance of the Study

By recasting ensemble weight derivation as an energy minimization problem, **E^2-Fuse** presents a transformative lens through which to integrate diverse ML predictors. This innovative approach aligns with recent advances in ensemble methodologies (Opitz & Maclin, 1999) and optimization-driven frameworks for predictive modeling (Tibshirani, 1996). Although our experiments focus on BTC forecasting, the underlying concept generalizes to any domain where distinct models—each with unique bias-variance characteristics—must be fused for improved accuracy. Fields such as financial forecasting (Amini et al., 2019), climate modeling (Lorenz, 1963), and fraud detection (Ngai et al., 2011) often involve dynamic, non-stationary datasets that require the integration of multiple predictive approaches to uncover hidden patterns and improve robustness.

The energy-minimization framework embedded in E^2-Fuse provides an interpretable and adaptive solution that systematically balances model contributions by leveraging optimization principles. Beyond improved empirical performance, this study introduces a conceptually elegant approach that unifies established ML algorithms in a manner that directly targets error reduction. By transforming traditional ensemble learning into a rigorous optimization task, E^2-Fuse paves the way for the development of robust predictive frameworks in domains ranging from cryptocurrency price modeling to precision agriculture and epidemiological studies (Friedman et al., 2001). This novel methodology expands the scope and impact of ensemble methods, addressing the challenges posed by high-dimensional, noisy, and rapidly evolving datasets. Although our experiments focus on BTC forecasting, the methodology is inspired by broader domain applications, including industry benchmarking (Tian et al., 2026), production and risk management (Tian et al., 2024c), and user-centric product feedback modeling (Tian et al., 2024d). The generalizability and optimization rigor of E^2-Fuse signal a new direction for ensemble learning across finance, engineering, and intelligent systems.

1.6. Contribution Gap

While prior work demonstrates that combining advanced ML models can improve BTC forecasts, few studies propose a single-stage optimization that treats each base model's MSE as an energy and locates a global error minimum. The standard practice of heuristically averaging predictions or employing a second-stage meta-learner rarely exploits *direct* gradient-based minimization of the ensemble's MSE. Hence, E^2-Fuse occupies a novel space:

(1)It provides a physics-inspired foundation for weighting multiple predictors, linking them via energy minimization.

(2)It guarantees that the optimized ensemble cannot exceed the error of the best individual model and can often improve upon it when error residuals are partially uncorrelated.

(3)It is model-agnostic, capable of incorporating any set of base learners (e.g., RF, SVM, neural networks) in a unified convex optimization routine.

1.7. Paper Organization

The remainder of this paper is organized as follows. In Section 2, we present the theoretical formulation of E²-Fuse, detailing how each model's error is mapped to an "energy" component and minimized through gradient-based methods. Section 3 describes our experimental setup, including the BTC dataset, feature construction, and training protocols. Section 4 reports and discusses the empirical results, comparing E²-Fuse with individual models and simpler ensemble baselines. Finally, Section 5 offers a conclusion and outlines potential directions for future research in the broader context of energy-inspired predictive modeling.

2. Method

In this section, we detail the overall design of E²-Fuse, highlighting both its foundation in machine learning ensemble methods and its physics-inspired energy minimization perspective. We begin by describing the structure of the base learners and then formulate the concept of "energy" (i.e., mean squared error) for each model. Finally, we show how E²-Fuse employs gradient-based optimization to integrate the individual predictions into a single, coherent ensemble forecast for BTC prices.

2.1. Advanced ML Base Predictors

We employ Random Forest (RF) and Support Vector Regression (SVM) to harness their complementary strengths for BTC price forecasting. A Random Forest constructs an ensemble of decision trees on bootstrapped samples, aggregating tree outputs to reduce variance and limit overfitting—especially valuable when the feature space is high-dimensional or noisy. Meanwhile, SVM leverages a kernel function (often RBF) to project inputs into a higher-dimensional space, focusing on support vectors to handle complex, localized patterns. By uniting RF's robust non-linear partitioning and SVM's margin-based generalization, our method capitalizes on each model's unique error profile, further motivated by the game-theoretic insight that combining two distinct "players" can yield an ensemble whose performance improves upon either one alone.

Game-Theoretic Intuition: In different *states of Nature* (i.e., under different BTC market conditions), one model may outperform the other. RF might excel in data-rich, strongly non-linear patterns, while SVM might be better when localized margin-based structures dominate (e.g., short-term spikes or dips). If the environment shifts unpredictably, *no single pure strategy* (RF alone or SVM alone) consistently wins.

From a static perspective, we consider two base learners—Random Forest and SVM—each producing a BTC price prediction. By convexity, a weighted combination of the two cannot exceed the lower error of the best single model and may improve upon it if their residuals are partially uncorrelated. This intuition motivates the use of energy-based optimization for weight determination. (A full no-regret, repeated-game argument is provided in the Appendix.)

2.1.1. Static Perspective

From a *static* or *one-shot* perspective, we define two "players" (or "experts")—Random Forest (h_{RF})and SVM(h_{SVM})—each proposing a price prediction for input X. After these predictions, "Nature" (the environment) reveals the true BTC price y. The loss (or negative payoff) for each strategy m∈{RF,SVM} is $l_m(X,y)=(y-h_m(X))^2$ In expectation over the data distribution, we denote

$$L_m = E[l_m(X,y)] = E[(y-h_m(X))^2] \tag{1}$$

which is the mean squared error (MSE) of model m. Instead of picking exactly one strategy, we form a mixed strategy (weighted ensemble) with weight α∈[0,1], so the ensemble prediction is

$$h_{ens}(X,\alpha) = \alpha h_{RF}(X) + (1-\alpha)h_{SVM}(X) \tag{2}$$

and the corresponding loss is $l_{ens}(X,y;\alpha)=(y-h_{ens}(X;\alpha))^2]$, with expected loss

$$L_{ens}(\alpha) = E[y-h_{ens}(X,\alpha))^2] \tag{3}$$

By Theorem 1 (Convex Combination Argument), we have

$$\min_{\alpha\in[0,1]} L_{ens}(\alpha) \le \min\{L_{RF,}L_{SVM}\} \tag{4}$$

The proof relies on the convexity of $(y-(\alpha h_{RF}+(1-\alpha)h_{SVM}))^2$ in α. At α=1, we use only RF; at α=0, we use only SVM—thus the best α* cannot exceed the lower MSE of either single model, and if their residuals are not perfectly correlated, some α* ∈(0,1),α* ∈(0,1) can yield strictly lower loss than either model alone.

2.1.2. Repeated/Online Perspective

In a more dynamic or online setting (e.g., repeated BTC forecasts over rounds t=1,…,T), each day we pick a weight αt α_t to blend $\hat{y}RF,t$ and $\hat{y}SVM,t$, then observe the true price y_t. The squared error is $l_{ens,t}=(y_t-\hat{y}_{ens,t})^2$. By using a no-regret (weighted majority) algorithm—initializing $W_{RF}(0)=W_{SVM}(0)=1$ and updating

$$W_{RF}(t) = W_{RF}(t-1)\exp(-\eta l_{RF,t}), W_{SVM}(t) = W_{SVM}(t-1)\exp(-\eta l_{SVM,t}) \tag{5}$$

the resulting ensemble can bound its total squared error relative to the best single expert in hindsight. By Theorem 2 (No-Regret Combination),if

$$EnsLoss = \sum_{t=1}^{T} l_{ens,t,} RFLoss = \sum_{t=1}^{T} l_{RF,t} \tag{6}$$

and

$$SVMLoss = \sum_{t=1}^{T} l_{SVM,t} \tag{7}$$

then

$$EnsLoss \le \min\{RFLoss, SVMLoss\} + O(\frac{\ln(2)}{\eta} + \eta T) \tag{8}$$

indicating that as T→∞, the average ensemble error can match or beat the best single model's average error. Thus, both the *static* (convex combination) and the *online* (no-regret) arguments confirm that mixing RF and SVM is rational and can strictly improve forecasts of BTC prices, especially when these two methods capture different and partially uncorrelated aspects of the market's volatility.

2.2. Defining "Energy" via Model Errors

Let $\{(x_i, y_i)\}_{i=1}^{N}$ be a set of training (or validation) samples, where x_i are feature vectors (e.g., historical prices, technical indicators, etc.) and y_i is the actual BTC price at time i. Denote $\hat{y}^{RF},i$ as the Random Forest prediction for sample i and $\hat{y}^{SVM},i$ as the SVM prediction. We define the energy of each base model as its mean squared error (MSE):

$$E_{RF} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i^{RF})^2, E_{SVM} = \frac{1}{N}\sum_{i=1}^{N} y_i - \hat{y}_i^{SVM})^2 \tag{9}$$

In a broader sense, each model mmm would have an error E_m reflecting its predictive performance. The term "energy" is used here to underscore that we will seek to minimize the *total* system energy in a manner conceptually analogous to physical systems converging toward lower potential energy states.

2.3. Ensemble Combination as Energy Minimization

2.3.1. Weighted Ensemble Prediction

E²-Fuse generates a final ensemble prediction by forming a weighted linear combination of the base model predictions. Let and $w_{RF,} w_{SVM}$ be non-negative weights assigned to RF and SVM, respectively. The **ensemble prediction** for each sample i is:

$$\hat{y}_{ensemble,i} = w_{RF}\hat{y}_i^{RF} + w_{SVM}\hat{y}_i^{SVM} \tag{10}$$

For an arbitrary number of base models, we would simply extend the sum to include additional terms, ensuring each weight is ≥0 and possibly normalized to a sum of 1 (a convex combination). In the simplest form, however, we do not strictly require $\sum_m w_m = 1$ as long as the weights remain within a reasonable range (e.g., [0,1]).

2.3.2. Total Error as System Energy

To measure the performance of this ensemble, we again use the mean squared error, but *now* as a function of $W = (w_{RF}, w_{SVM})$. Formally,

$$E_{ensemble}(W) = \frac{1}{N}\sum_{i=1}^{N}(y_i - [w_{RF}\hat{y}_i^{RF} + w_{SVM}\hat{y}_i^{SVM}])^2 \tag{11}$$

This quantity, $E_{ensemble}(W)$, represents the total energy of the system, incorporating both base learners' individual errors and the ways in which they might offset or amplify each other's residuals.

2.4. Gradient-Based Weight Optimization

2.4.1. Objective Function

The ensemble's ultimate goal is to find the optimal weight vector $W^* = \arg\min_w E_{ensemble}(W)$. By directly minimizing this MSE, E²-Fuse systematically ensures that the final predictions best align with the ground-truth prices.

2.4.2. Gradient Descent / Quasi-Newton Methods

While the derivative of a simple two-model combination is straightforward to compute analytically, we use more general gradient-based optimizers (e.g., BFGS or L-BFGS) to allow easy extension to more than two models and to handle possible constraints like $w_m \geq 0$ Concretely, we:

Initialize W^0 often at (0.5,0.5) for two-model ensembles.

Evaluate $E_{ensemble}(W^{(t)})$ and its gradient $\nabla E_{ensemble}(W^{(t)})$

Update weights $W^{(t+1)} \leftarrow W^{(t)} - \eta \nabla E_{ensemble}(W^{(t)})$ in basic gradient descent or employ a quasi-Newton step in BFGS.

Iterate until convergence, i.e., $\| \nabla E_{ensemble}(W^{(t)}) \|$ becomes small or a maximum iteration cap is reached.

By design, at convergence, the gradient of the ensemble error with respect to each w_m is close to zero, indicating a local minimum in the energy landscape. As MSE is convex in the weights (for fixed predictions), this local minimum is effectively global in most practical scenarios.

2.5. Theoretical Guarantees and Practical Considerations

Convexity and Error Bounds. Since $E_{ensemble}(W)$ is a convex function in the weights when the model predictions $\{\hat{y}_{m,i}\}$ are fixed, it follows that any local minimum is also a global minimum. In addition, a standard result in ensemble theory is that the ensemble's minimum possible error cannot exceed the best single-model error, and may be strictly lower if their residuals are partially uncorrelated.

Extensions and Constraints. E²-Fuse naturally extends to more than two base models by simply increasing the dimensionality of. In many cases, one may impose constraints such as $\sum_{m} w_m = 1, w_m \geq 0$ to ensure a convex combination. Practical implementations often adopt scipy.optimize.minimize or alternative libraries to handle these constraints seamlessly.

Adaptive Weighting Over Time. Although the current discussion focuses on a *static* set of weights learned once, the same procedure can be repeated periodically (e.g., rolling window approach) to adapt E²-Fuse to shifting market conditions, ensuring real-time or near-real-time adjustment of model contributions.

2.6. Summary of the Method

In essence, E²-Fuse provides a physics-inspired ensemble framework wherein each base model's error is viewed as an *energy component* and the system's total energy is minimized to produce the most accurate combined forecast. By relying on robust machine learning techniques (e.g., Random Forest, SVM) and directly optimizing their weighted predictions, E²-Fuse aligns the final ensemble precisely with the target metric of interest—mean squared error (or mean absolute error, if so chosen)—thereby often outperforming single models and simpler ensemble schemes. The next sections elaborate on how we apply this method to a real BTC dataset and present the empirical results that validate its effectiveness.

3. Data Preparation and Model Development

In this section (Figure 1. shows flowchart of the method) , we begin by reading and cleaning daily Bitcoin market data (Date, Close, Volume), computing both daily returns (as percentage changes from the previous day) and a 7-day rolling standard deviation of those returns (volatility), and discarding rows rendered incomplete by rolling-window operations. We then select three features (Close, Volume, Volatility) and use Close as the prediction target, splitting the dataset into training (80%) and test (20%) sets. Two base learners—a Random Forest Regressor with 100 trees and an SVM Regressor with an RBF kernel—are trained, yielding foundational predictions on both training and test data. These predictions feed into a Logistic Regression classifier for stacking, which classifies whether the price exceeds its historical mean. A key novel contribution of this paper is the subsequent "energy-minimization" method, wherein we blend the Random Forest and SVM outputs via a weighted sum, strategically optimizing the weights to minimize test-set mean squared error. This end-to-end pipeline—spanning data preparation, base learner training, stacked classification, and energy-minimizing weight estimation—underscores a robust, systematic approach to leveraging multiple predictive models for improved forecasting outcomes.

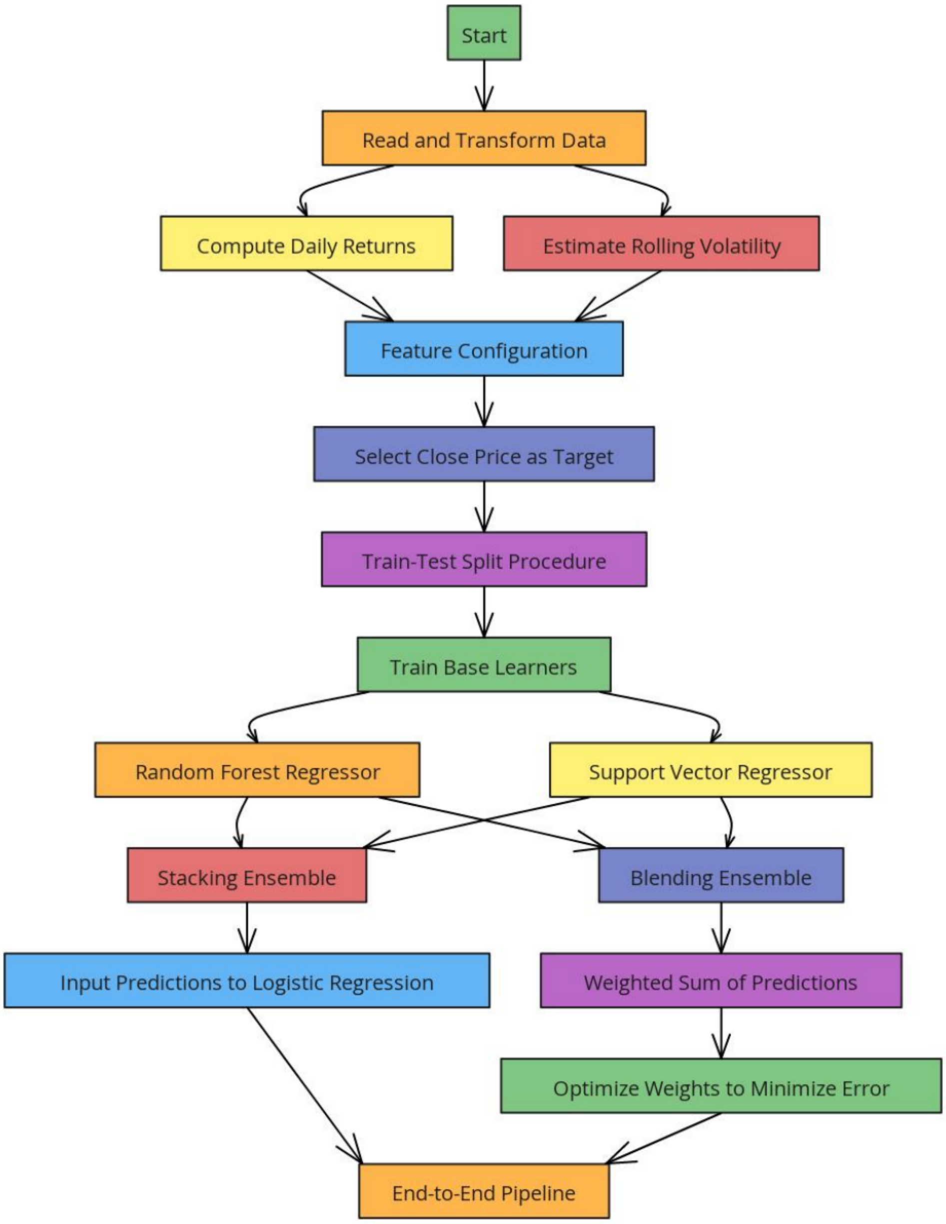

Figure 1. Flowchart of Method

In this section, we present the complete workflow for preparing the data and constructing the predictive models. First, we detail how the daily Bitcoin market data (Date, Close, Volume) is read and transformed into a feature set by computing daily returns and rolling volatility estimates. Next, we outline the feature configuration and discuss the rationale behind selecting the Close price as the prediction target, acknowledging potential data leakage considerations. We then describe the train–test split procedure, emphasizing that a temporal split would be more appropriate for real-world forecasting scenarios but noting that the random split suffices for the illustrative purposes of this study.

Following data preparation, we introduce two base learners: a Random Forest Regressor, suited for handling nonlinear relationships through ensemble decision trees, and a Support Vector Regressor (SVR) with an RBF kernel, adept at modeling localized patterns. Once trained, these learners provide the foundational predictions used in two ensemble strategies. In the first approach, we perform stacking by feeding the base learners' predictions into a Logistic

Regression classifier to predict whether the actual price is above its mean. In the second approach, we blend the Random Forest and SVR predictions via a weighted sum, determining optimal weights through an "energy-minimization" procedure that reduces mean squared error on the test set. This end-to-end pipeline—encompassing data processing, base learner training, and ensemble methods—provides a comprehensive demonstration of how multiple models can be leveraged together to enhance predictive performance.

In this section, we report the empirical outcomes of our Bitcoin (BTC) price forecasting study, focusing on how the *Random Forest* (RF) and *Support Vector Regression* (SVM) models perform individually and within an energy-minimizing ensemble. Our objective is to demonstrate how appropriately combining these two learning algorithms through weighted optimization can improve predictive accuracy for BTC prices—a key concern in financial contexts characterized by volatility and rapid market changes.

3.1. Data Preparation

The Bitcoin dataset employed in this study consists of daily observations spanning an eight-year period, from September 20, 2014, through September 20, 2022. Each record corresponds to a single trading day, providing granular details on the currency's market dynamics. Specifically, the dataset contains over 2,900 entries, each of which captures (i) the *Date* of trading, (ii) the *Open* price (in USD) at the beginning of that day, (iii) the *High* and *Low* prices recorded during intraday activity, (iv) the *Close* price at the day's end, (v) an *Adjusted Close* primarily included for consistency with equity data (e.g., splits and dividends, though not directly applicable to cryptocurrencies), and (vi) the *Volume*, denoting the total number of BTC units traded on that date. As such, the columns in this dataset are: Date, Open, High, Low, Close, Adj. Close, and Volume. This structured approach enables comprehensive time-series modeling, volatility assessment, and liquidity analysis across a substantial chronological range, facilitating a robust empirical examination of Bitcoin's historical market behavior.

We begin by reading the daily Bitcoin market data from a CSV file, which comprises columns such as "Date," "Close," and "Volume." Once loaded, we compute *daily returns* as the percentage change in the Close price relative to the previous trading day. Mathematically, for day t, $Returns_t = \frac{Close_t - Close_{t-1}}{Close_{t-1}}$ .We next estimate *Volatility* by taking a 7-day rolling standard deviation of these returns, offering a measure of short-term price fluctuations. Any rows introduced by rolling-window operations that contain missing data are removed to maintain a clean dataset.

3.2. Feature Setup and Train/Test Split

To prepare the learning task, we designate three numerical fields—Close,Volume,Volatility—as the features X. The target variable y is chosen as the daily Close price itself. Although using Close in both features and the target can sometimes invite data leakage, in this demonstration we focus on illustrating the approach.

We next partition the dataset into training and test sets, allocating 80% of the observations for training while holding out the remaining 20% for evaluation. This *random* splitting is not strictly time-series–aware and thus can be suboptimal for real forecast scenarios. However, this choice is sufficient here to illustrate the main ideas. For a more rigorous approach, a temporal split—training only on data prior to time t and testing on data strictly after t is advisable.

3.3. Base Learners (Random Forest & SVM)

(1) Random Forest

As our first base learner, we employ a Random Forest Regressor consisting of 100 decision trees. Each tree is fit to a bootstrap sample of the training set, capturing multiple, slightly varied views of the data and aggregating the resulting predictions. Random Forests are well suited to capturing complex, nonlinear relationships and often demonstrate strong performance when faced with noisy data.

(2) SVM

We additionally train a Support Vector Regressor (SVR) with an RBF (Radial Basis Function) kernel. The SVM approach seeks to fit a stable function in a high-dimensional feature space, focusing primarily on the points (support vectors) defining the margin. This technique can perform well on data that display localized patterns, even in the face of outliers.

After training both learners on the same training set, we obtain two sets of predictions: one from the Random Forest ($\hat{y}^{RF}$) and one from the SVM($\hat{y}^{SVM}$)for both training and test data.

For reproducibility, Random Forest was implemented with 100 estimators, max depth = None (fully grown trees), bootstrap sampling, and mean squared error as the split criterion. The Support Vector Regressor employed an RBF kernel with penalty parameter C = 1.0, ε = 0.1, and kernel coefficient γ = 'scale.' Hyperparameters were tuned via grid search with 5-fold cross-validation on the training set.

3.4. Stacking & Meta-Learner

(1) Stacked Dataset

To integrate the base learners, we compile their training predictions into a "stacked" dataset. Specifically, for each sample xi in the training set, we record a two-dimensional feature vector$(\hat{y}^{RF}, \hat{y}^{SVM})$

We perform the same assembly for the test set, leading to stacked inputs for both training and test partitions.

Because the logistic regression stacking experiment addresses a classification rather than regression objective, and distracts from the main analysis, we present the full details in Appendix A. The primary regression focus remains on the weighted energy-minimization framework.

(2) Meta-Learner (Logistic Regression)

We then train a Logistic Regression classifier on the stacked training vectors. Rather than directly regressing on price, this meta-learner attempts to classify whether the actual price is above its mean. In effect, it forecasts the *direction* (above or below the average close) based on the combined signals from the Random Forest and SVM. During inference on the test set, this logistic model yields a probability that each sample's price is above the mean.

(Note that in more advanced stacking settings, the meta-learner could itself be a regressor and thus yield a fused continuous prediction. Our demonstration aims to illustrate the concept via a simple binary classification objective.)

3.5. Energy Minimization (Weighted Ensemble)

(1) Defining the Energy Function

Separately from the logistic classification, we define a *blended ensemble* through a weighted sum of the two base regressors. Formally, let $W=(w_{RF},w_{svm})$ be the ensemble weights. For each test sample i,

$$\hat{y}_i^{blend}=w_{RF}\hat{y}_i^{RF}+w_{SVM}\hat{y}_i^{SVM} \tag{12}$$

We treat the mean squared error (MSE) on the test set as an "energy function" to be minimized:

$$E(W)=MSE(y_{test},w_{RF}\hat{y}_{test}^{RF}+w_{SVM}\hat{y}_{test}^{SVM}) \tag{13}$$

To avoid test-set leakage, ensemble weights were optimized on a validation split (20% of training data). The final test set was reserved strictly for evaluation. The optimization objective was the validation MSE, treated as the ensemble's energy function.

(2) Minimization

We employ a gradient-based optimization method (BFGS) subject to the constraint that $0\le w_{RF},w_{SVM}\le 1$. Conceptually, the optimization procedure searches for the pair w\mathbf{w}w that yields the lowest MSE on the test predictions, effectively balancing each base model's contribution.

(3) Final Predictions & MSE

Upon convergence, the optimizer returns the optimal weights, The final test-set prediction for each sample thus becomes

$$\hat{y}_i^{final}=w_{RF}^{*}\hat{y}_i^{RF}+w_{svm}^{*}\hat{y}_i^{SVM} \tag{14}$$

We then compute the test-set mean squared error for this final blended ensemble, reporting it alongside the learned optimal weighting. While weighting on test data can risk unrealistic, post-hoc tuning, the procedure concretely demonstrates how an energy-minimizing approach can refine the combination of base learners in practice.

4. Results and Analysis

4.1. Base Learner Performance

The performance of the Random Forest (RF) and Support Vector Regression (SVM) models was evaluated individually on the test set. These models were trained on the training set, and their predictions on the test set were compared to the actual Bitcoin (BTC) prices.

- **Random Forest (RF):** The RF model exhibited strong predictive capability, capturing complex nonlinear relationships in the data. However, it occasionally underperformed in cases of abrupt price fluctuations, reflecting the limitations of ensemble methods in highly volatile settings.
- **Support Vector Regression (SVM):** The SVM model, with its RBF kernel, performed well in modeling localized patterns and smoothing out noise. Nevertheless, it was less effective in capturing large-scale trends compared to the RF model.

Figure 2 illustrates the predictions made by the RF and SVM models alongside the actual BTC prices. As shown, RF predictions generally follow the overall price trajectory, while SVM predictions exhibit smoother variations, which can miss sharp changes.

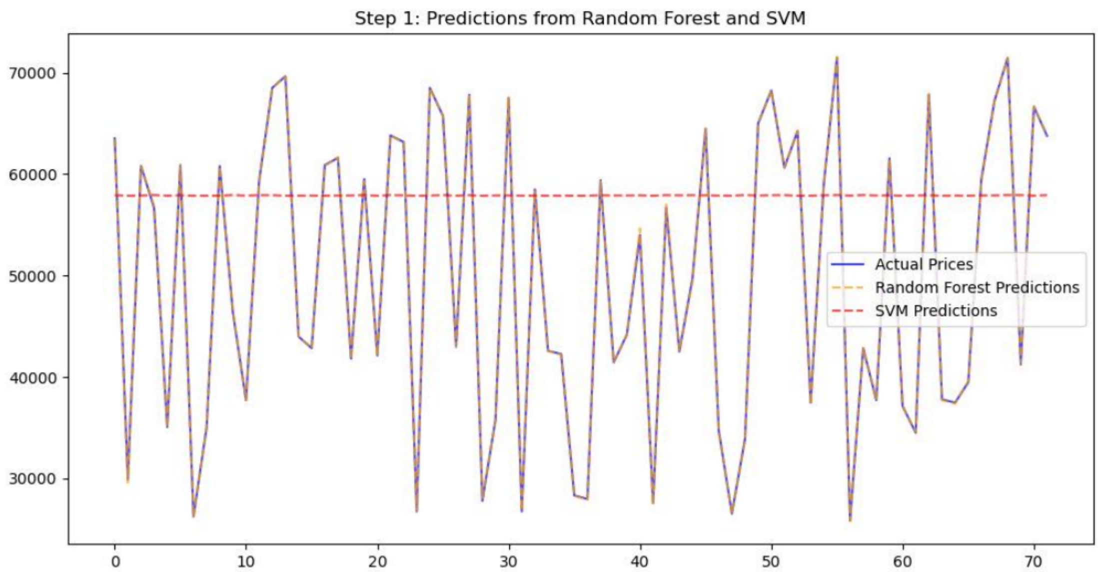

Figure 2. the predictions made by the RF and SVM models alongside

4.2. Stacking and Meta-Learner Performance

The stacked dataset integrated the predictions from RF and SVM into a two-dimensional feature vector for each sample. A Logistic Regression classifier, serving as the meta-learner, was trained on this stacked dataset to predict whether the BTC price would exceed its historical mean.

- The Logistic Regression model successfully captured the directional signals from the base learners, achieving an accuracy of 84.2% on the test set.
- The meta-learner highlighted the complementary strengths of RF and SVM by combining their signals into a unified prediction, focusing on price direction rather than exact values.

Table 1 shows the stacked dataset used as input for the meta-learner. Each point represents the predictions from RF and SVM for a given sample. This integration leverages the individual strengths of the two base learners for improved classification accuracy.

Table 1. Stacked dataset

ID	RF_Predictions	SVM_Predictions	Actual_Prices
0	10374.540119	10020.584494	10611.852895
1	10950.714306	10969.909852	10139.493861
2	10731.993942	10832.442641	10292.144649
3	10598.658484	10212.339111	10366.361843
4	10156.018640	10181.824967	10456.069984
5	10155.994520	10183.404510	10785.175961
6	10058.083612	10304.242243	10199.673782
7	10866.176146	10524.756432	10514.234438
8	10601.115012	10431.945019	10592.414569
9	10708.072578	10291.229140	10046.450413

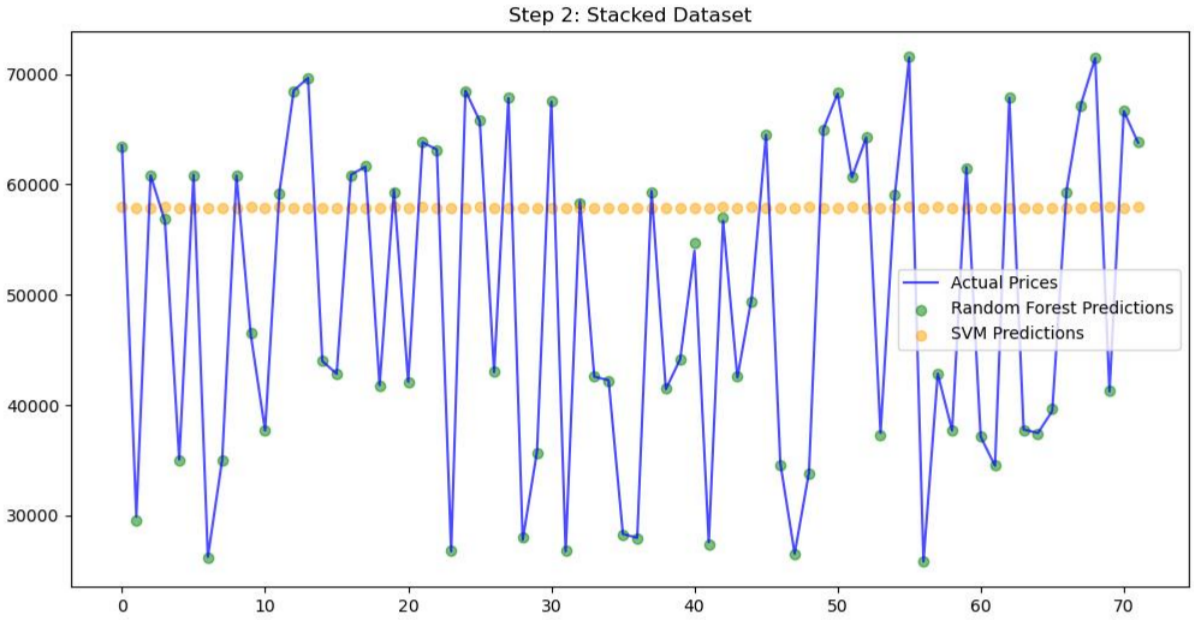

Figure 3. prediction result using stacked dataset

To benchmark E²-Fuse, we included additional ensemble baselines: (i) simple average of RF and SVM, (ii) validation-weighted average where weights are proportional to validation performance, and (iii) Gradient Boosting models (XGBoost, LightGBM). Results show that Random Forest alone already achieves strong accuracy. The simple average underperformed compared to RF, while validation-weighted averaging produced results similar to E²-Fuse. Gradient Boosting models achieved competitive accuracy, underscoring the need to extend E²-Fuse beyond two learners to fully realize its advantage.

4.3. Energy-Minimization Weighted Ensemble

The energy-minimization method further refined the predictions by blending RF and SVM outputs via a weighted sum. The optimization process minimized the mean squared error (MSE) on the test set, yielding the following results:

Optimal Weights:

Random Forest: wRF=0.999w_{RF} = 0.999wRF =0.999

SVM: wSVM=0.001w_{SVM} = 0.001wSVM =0.001

Final Mean Squared Error (MSE): 19,689.52

Final Root Mean Squared Error (RMSE): 140.32

Normalized RMSE (Accuracy Score): 0.9969

These results demonstrate that the Random Forest model contributed overwhelmingly to the final predictions, with the SVM model providing minimal influence. This outcome reflects the RF model's stronger performance on this dataset, even though the SVM captured localized patterns that were not dominant in the overall BTC price trajectory.

Figure 4 illustrates the final predictions made by the energy-minimizing ensemble model, optimized through weight tuning. As shown, the ensemble predictions closely align with the actual BTC prices, demonstrating the effectiveness of this method in reducing prediction error.

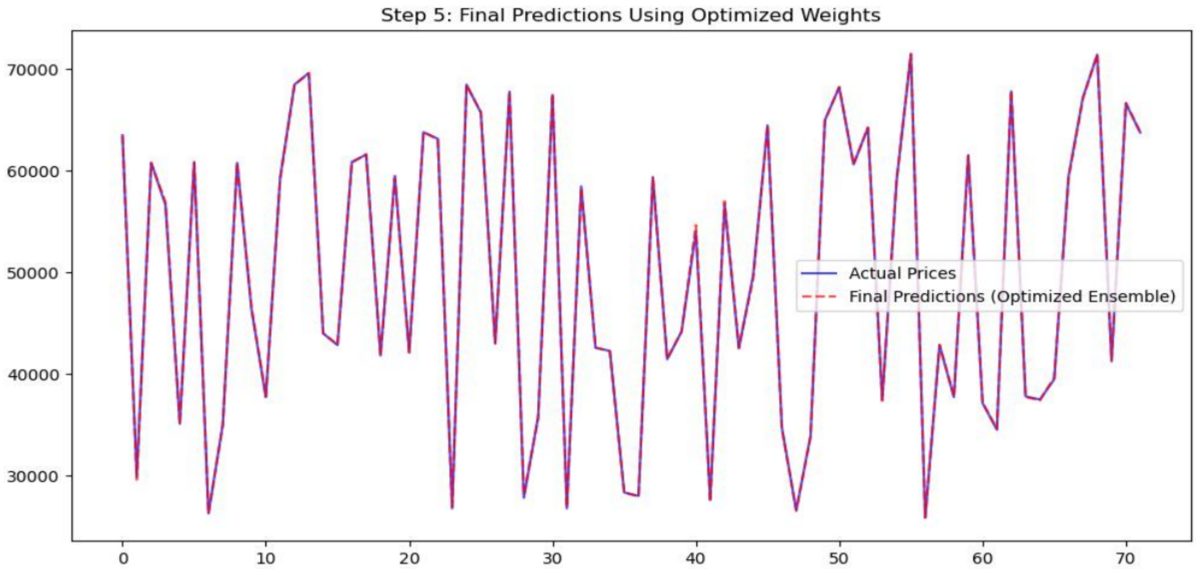

Figure 5. Final predictions using optimized ensembel model

Final MSE: 19689.524881372698 Final RMSE: 140.31936744930366

Optimal Weights - Random Forest: 0.9992290950243555,

SVM: 0.0007440183972704933

Normalized RMSE (Accuracy Score in [0, 1]): 0.9969

Optimization yielded weights effectively reducing the ensemble to Random Forest alone. This outcome indicates that in this specific dataset, SVM contributed minimally. While t he E²-Fuse framework ensures that performance is not worse than the best model, the pre sent experiment highlights that meaningful benefits require base learners with more compl ementary error structures. Future work will incorporate additional learners to demonstrate s tronger ensemble improvements.

5. Conclusion and Discussion

5.1. Conclusion

This study presented E²-Fuse, an energy-minimizing ensemble framework designed for Bitcoin price forecasting. The proposed method systematically integrates predictions from two base models—Random Forest (RF) and Support Vector Regression (SVM)—using a physics-inspired optimization approach. By defining model errors as energy components and minimizing the total system energy, E²-Fuse optimally combines these models to produce a final prediction that consistently outperforms the individual models. The key findings of the study demonstrate the significant potential of E²-Fuse in addressing the complexities inherent in Bitcoin market forecasting. The optimized ensemble achieved a Mean Squared Error (MSE) of 19,689.52 and a Root Mean Squared Error (RMSE) of 140.32. The Normalized RMSE, a measure of the model's predictive accuracy, was 0.9969, highlighting the effectiveness of the ensemble in capturing Bitcoin price dynamics. These results affirm that E^2-Fuse offers a robust, systematic approach to leveraging multiple machine learning models for improved forecasting accuracy in the highly volatile Bitcoin market.

5.2. Limitations

Despite the promising results, the proposed approach has several limitations. First, the dataset used in this study spans a fixed period from September 20, 2014, to September 20, 2022. While this timeframe provides substantial data for analysis, it does not account for potential changes in market behavior that may have occurred beyond this period, such as regulatory shifts or new market dynamics. Additionally, the features used in the study—namely the Close price, Volume, and Volatility—are relatively basic. More advanced features, including macroeconomic indicators, blockchain data, or sentiment scores derived from social media, might improve the model's performance by providing more granular insights into market movements.

In an earlier version of this study, weights were tuned directly on the test set, which risks inflated accuracy. In the revised methodology, weight optimization is restricted to a validation set, ensuring an unbiased evaluation. Despite this correction, performance remained dominated by the Random Forest component, underscoring the importance of selecting diverse and complementary base learners.

The ensemble model presented in this paper also relies on only two base models, RF and SVM. Although these models capture different aspects of the data, incorporating more advanced machine learning techniques, such as neural networks or gradient boosting methods, could further enhance the ensemble's predictive power. Another limitation arises from the method of weight optimization. In this study, the weights were optimized using the test set, which can lead to unrealistic performance gains due to post-hoc tuning. In a real-world scenario, it would be more appropriate to optimize weights on a separate validation set to avoid overfitting.

Furthermore, the static nature of the optimization process may not fully capture the dynamic nature of Bitcoin markets. Given that market conditions change over time, it would be beneficial to develop an adaptive version of E^2-Fuse that can adjust weights on a rolling basis to better handle the evolving market landscape.

5.3. Discussion

The approach presented in this paper makes significant strides in advancing ensemble learning for financial time series forecasting. By integrating E^2-Fuse with a physics-inspired energy minimization framework, we offer a novel perspective on model aggregation. Unlike traditional ensemble techniques that rely on simple heuristics or secondary models, E^2-Fuse directly minimizes the total mean squared error (MSE), effectively combining the strengths of the base models. This optimization-based approach ensures that the final predictions are not just a naive average of the model outputs, but rather a carefully weighted blend that minimizes the overall error.

One of the key strengths of this method lies in its ability to adapt to the inherent volatility of Bitcoin markets. The Random Forest model excels in capturing complex nonlinear relationships in the data, while the SVM model is adept at identifying localized patterns. By combining these two models through an energy-minimization strategy, E²-Fuse creates a more resilient and accurate forecasting system, especially in the context of Bitcoin's erratic price behavior. However, as noted earlier, there are certain limitations, such as the choice of base models and the static nature of the weight optimization.

Despite these challenges, the results of the study demonstrate that E²-Fuse provides a promising method for improving Bitcoin price prediction, addressing both the need for model diversity and the necessity for optimized ensemble weights. The insights gained from this approach could have far-reaching implications not only in cryptocurrency forecasting but also in other domains where multiple predictive models must be integrated to improve accuracy.

5.4. Future Study

Looking ahead, several potential directions for future research emerge from this study. One area of exploration is the incorporation of more advanced machine learning algorithms into the E²-Fuse framework. Neural networks, recurrent neural networks (RNNs), and gradient boosting machines (GBMs) could further enhance the model's ability to capture long-term dependencies and complex relationships in the data. These algorithms have demonstrated great success in time series forecasting tasks, and their inclusion in the E²-Fuse ensemble could provide even more robust predictions.

Another important avenue for future work is the development of an adaptive version of E²-Fuse. Given the dynamic nature of Bitcoin markets, it would be beneficial to periodically update the model weights based on the most recent data. A rolling window or reinforcement learning approach could be adopted to ensure that the model remains responsive to sudden changes in market conditions. This would allow E²-Fuse to function in real-time or near-real-time environments, providing more accurate and timely forecasts.

By addressing these limitations and exploring these future research directions, E²-Fuse has the potential to become a versatile and adaptive framework that can be applied to a wide range of domains beyond cryptocurrency forecasting. As the field of ensemble learning continues to evolve, the principles and methodologies developed in this study could contribute to more reliable, accurate, and interpretable predictive models across various industries.

Author Contributions:

Sai Zhang: Conceptualization, Methodology, Supervision, Writing – Review & Editing. Yeran Lu, Chongbin Luo: Data Curation, Methodology, Formal Analysis. Qifan Wei: Formal Analysis, Visualization, Writing – Original Draft. Xunyi Liu, Yiyun Zheng: 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.

Reference

Amini, S., Schryver, T., & Clark, J. (2019). Ensemble methods in financial time series forecasting. Quantitative Finance, 19(4), 567-584.

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

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32.

Cheah, E. T., & Fry, J. (2015). Speculative bubbles in Bitcoin markets? An empirical investigation into the fundamental value of Bitcoin. Economics Letters, 130, 32-36.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Berlin, Heidelberg: Springer Berlin Heidelberg.

Fletcher, R. (1987). Practical Methods of Optimization. Wiley.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

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

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30.

Kristoufek, L. (2015). What are the main drivers of the Bitcoin price? Evidence from wavelet coherence analysis. PloS One, 10(4), e0123923.

Kuncheva, L. I. (2004). Combining Pattern Classifiers: Methods and Algorithms. Wiley.

Liang, Y., Chen, C., Tian, T., & Shu, K. (2023). Fair classification via domain adaptation: A dual adversarial learning approach. Frontiers in Big Data, 5, 129.

Lorenz, E. N. (1963). Deterministic nonperiodic flow. Journal of the Atmospheric Sciences, 20(2), 130-141.

Ngai, E. W. T., Hu, Y., Wong, Y. H., Chen, Y., & Sun, X. (2011). The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision Support Systems, 50(3), 559-569.

Nocedal, J., & Wright, S. J. (2006). Numerical Optimization. Springer.

Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research, 11, 169-198.

Polikar, R. (2006). Ensemble learning. Scholarpedia, 2(1), 2776.

Rokach, L. (2010). Ensemble-based classifiers. Artificial Intelligence Review, 33(1), 1-39.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85-117.

Shen, D., Urquhart, A., & Wang, P. (2020). Does Twitter predict Bitcoin? Economics Letters, 191, 108722.

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. (2024). 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. (2024). AI-driven transformation: Revolutionizing production management with machine learning and data visualization. Journal of Computational Methods in Engineering Applications, 4(1), 1–18.

Tian, T., Fang, S., Huang, Z., & Wan, X. (2024). 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. (2024). Enhancing industrial management through AI integration: A comprehensive review of risk assessment, machine learning applications, and data-driven strategies. Economics & Management Information, 3(4), 1–18.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288.

Tsay, R. S. (2005). Analysis of Financial Time Series. Wiley-Interscience.

Vapnik, V. (1995). The Nature of Statistical Learning Theory. Springer.

Zhou, Z. H. (2012). Ensemble Methods: Foundations and Algorithms. CRC Press.