AB-Swish: A Novel Activation Function for Neural Networks in Hybrid Portfolio Optimization Frameworks

Abstract

Portfolio optimization is a core problem in financial engineering, as it tries to find the best possible tradeoff between risk and return. Activation functions (AFs) are important aspects of neural networks, as they can have a direct influence on convergence behavior and predicting performance. This paper presented a novel AF, AB-Swish, and showed its utility in a hybrid machine learning and optimization framework developed for portfolio construction. The framework applies XGBoost to classify market trends and predict returns by using Radial Basis Function (RBF) neural networks employing AB-Swish activation functions, then applying traditional Mean–Variance Optimization. The proposed activation demonstrated better overall returns when compared to traditional and well-known activation functions such as ReLU, Logistic, Tanh, and Identity. The ability of the framework to provide new portfolio allocations was tested on a multi-asset data set that included Equity, Debt, Gold, and Bitcoin. The data was robustly scaled, and a detailed outlier detection and removal process was applied. XGBoost classified the market conditions, and then the predicted returns from the AB-Swish RBF were utilized for optimization in conjunction with the allocation. Using this methodology allows the user to integrate their asset preferences while still adapting to the current market regimes. The results of the experiments show that the proposed system allows an improvement to the effectiveness of portfolio allocation, provides flexibility, can be scaled, and adapted, thus suggesting it should be seriously considered for any investor who desires an effective, initial step in applying machine learning to their real-world use.

Keywords: Portfolio optimization, Radial Basis Function Neural Networks, XGBoost Classification, Return forecasting, Risk-Return Management, Mean-Variance Optimization.

1. Introduction

Portfolio optimization is a fundamental practice of financial engineering, allowing investors to build portfolios that weigh risk and return for targeted investment objectives. Modern portfolio theory (MPT) established a mathematical basis for asset allocation with the introduction of the mean–variance optimization (MVO) framework by Harry Markowitz in 1952. MVO, although convenient and widely adopted, rests on awful assumptions regarding returns, which drive investment allocations: normally distributed rates of return, and independence between time-indexed events based on stable historical relationships. In practice, when markets are volatile and dynamic, these assumptions often break down. Financial market observations have significant sampling error and MVO, hence results in a poor estimator becoming optimal allocations. This has caused many researchers to examine approaches that are a blend of classic finance theory, artificial intelligence (AI), and machine learning (ML).

Recent advances in AI have greatly enabled portfolio construction as a result of the ability of models to remove complex and nonlinear relationships in financial data. Both gradient boosting (XGBoost) and neural networks are able to retain complicated relationships in time-series data and deliver upgraded forecasts of asset returns and market regimes. However, in real-world applications, as in financial datasets, there are typically some bad observations (outliers), potential deviations in investor constraints, and a need for computational expertise to solve multi-asset allocation problems. To address these problems, we introduce a hybrid ML–optimization framework that combines XGBoost for market trend classification, Radial Basis

Function (RBF) neural networks for return forecasting, and MVO to provide the optimal allocation where realistic investment constraints apply.

In our framework, the RBF neural network uses a newly created activation function, dubbed AB-Swish, using the most efficient learning and predictive accuracy for financial time series modeling. AB-Swish is based on the Swish family of smooth non-monotonic functions and includes tunable parameters to adapt the function's shape to the dataset. The characteristics of AB-Swish will enhance the network's generalization ability in noisy and non-stationary financial environments.

In evaluating the proposed framework, we consider a combination of data from a diverse asset class over 10 years (2014–2024): equity (Nifty 50), fixed income (Indian government bonds), commodities (gold), and cryptocurrency (Bitcoin). In the data pre-processing stage, we used robust scaling and outlier-based removal of extreme values based on the interquartile range (IQR) method. We first classify the outcomes using the XGBoost classifier to predict the state of the market (e.g., bullish or bearish); then, the AB-Swish–based RBF model is used to predict the asset returns. These forecasts are utilized within an MVO framework, which employs various constraints that align with actual portfolios. For example, allowing only a 20% allocation in Bitcoin to mitigate volatility, while also having a minimum allocation in bonds. The major contributions of this current work are:

- Hybrid machine learning–optimization framework: The framework provides for both an XGBoost classification, AB-Swish–based RBF prediction, and MVO allocation methods for creating market-adaptive portfolios.
- A realistic framework of constraints and allocation rules: Examples are a set of investor-relevant allocation rules, to a greater degree, in order to reflect real-world portfolio management.
- Empirical validation: The results show that the framework works using available historical data: 1.07 Sharpe ratio, and approximately 25% annual return (higher than the performance outcome of both different standalone MVO and an equally-weighted portfolio).

1.1. Motivation of the Model

The Mean–Variance Optimization (MVO) framework is relatively simple to follow and is extremely well-known, but due to its linear and distributional assumptions, it is limited by the complexity of the real-world markets. While deep learning lends more flexibility, archetypal model architectures like multilayer perceptron networks or Long Short-Term Memory (LSTM) networks can be prone to overfitting, and do not provide adequate transparency, interpretability, or complexity when capturing a financial time series. The Radial Basis Function (RBF) neural network is well-suited for the task – it uses localized response functions to capture complex, nonlinear relationships, while also being robust and transparent. This positional learnability helps our models resist memorizing market noise more so than general deep networks, and local regional focus learning makes our models better suited to accommodate regime shifts in time series data. Our deep RBF incorporates an adaptive activation function (AB-Swish) to explicitly support the learning dynamics of financial data, aiding greatly towards improving on many learner aspects we later share in predictive accuracy and stability over traditional deep learning models. These advantages will support risk-targeted investment portfolio construction in volatile conditions.

1.2. Novelty of the Model

The proposed framework features a hybrid approach that combines conventional optimization with advanced machine learning. An XGBoost classifier identifies market regimes, followed by an RBF Neural network that estimates asset return expectations. Specifically, the RBF network uses the AB-Swish activation function, which is a new architecture, but a variant of the Swish family of activation functions, that features tunable parameters that allow for a closer estimation of the nonlinear and noisy features of financial time series. This past work clearly suggests this is the first time XGBoost, AB-Swish-based RBF network, and Mean–Variance Optimization have been used together in the same portfolio construction framework. The proposed framework is also pragmatic in incorporating feasible constraints—such as limiting Bitcoin to 20% or maintaining a minimum of 10% in bonds—thus providing a flexible and robust optimization framework that accommodates shifts across market regimes. Overall, the provided framework is, therefore, designed as a scalable and computationally efficient approach that offers an academically relevant bridge between theoretical optimization and portfolio management.

The remainder of the paper is organized as follows. Section 2 describes the literature reviews related to prior work on machine learning for portfolio optimization and the design of the activation function. Section 3 presents the AB-Swish activation function and its properties. Section 4 describes the hybrid portfolio optimization framework. Section 5 details the experimental setup, datasets, evaluation metrics, and comparative analysis with other models. Section 6 highlights the limitations of the proposed method. Section 7 details the overall structure of the paper, followed by the acknowledgments section in Section 8.

2. Literature Review

The stock market, combined with commodities, bonds, etc., comprises numerous entities for the public to invest in and get ample returns. Still, many people in India do not have ample knowledge about any kind of investments. This results in lost opportunities, higher financial risks, and under-utilization of resources. To manage this, people can either invest in multi-class assets or single assets to boost their returns in the long term. Chhajer et al. (2022) gave an overview of artificial intelligence and machine learning as predictive tools for stock market forecasting using an overview of research concerning neural networks, support vector machines, and long short-term memory models for machine learning applications.

Over the past few decades, researchers have tapped into a range of AI and optimization techniques to improve portfolio forecasting and portfolio construction. Limited early studies, such as Freitas et al. (2009) showed that auto-regressive moving reference neural networks (AR-MRNN) could outperform classical mean-variance constructs in the Brazilian market and yield significantly more return with a reasonable level of risk. In a later study, Di Persio et al. (2017) explored RNN LSTM and GRU networks on Google stock data, showing 72%, while Zhao et al. (2017) also utilized an improved time-weighted LSTM, thus producing an increased accuracy of 83.91% on the CSI 300 index. In more recent research, Das et al. (2024) studied the accuracy of BiGRU and BiLSTM models in highly volatile markets, concluding that integrating deep learning with some clustering methods, such as Affinity Propagation, produces a better portfolio than either method when deployed separately. Likewise, Ma et al. (2020) tested DMLP and LSTM, and CNN models using the China Securities 100 Index, concluding that the DMLP model was the best-performing model of the three.

Reinforcement learning has become a large and growing area of focus in finance as well. Consigli et al. (2024) developed a reinforcement learning framework that contains conditional interval value-at-risk (ICVaR) components to produce S&P 500 beating strategies. Jiang and Liang (2017) applied deep Reinforcement Learning directly in cryptocurrency portfolios, demonstrating strong generalization across assets. Kendall et al. (2018) developed adaptive loss weighting in multi-task learning, which is a technique used to balance objectives in portfolio optimization. Similarly, Chalvatzis and Hristu-Varsakelis (2025)demonstrated the advantage of using machine learning over long horizons on 35 years of NYSE data, achieving seven-fold returns even when transaction costs were included. The other strong area of research is based on radial basis function (RBF) networks. Aljarah et al. (2018) explored the problem of local optima in neural network design by utilizing a biogeography-based optimizer, and Li et al. (2024) demonstrated improvements in RBF training utilizing adaptive genetic algorithms and recursive least squares. Vladov et al. (2024) discussed polymorphic RBFs, which allowed for additional flexibility and accuracy/precision, while Zhang (2025) integrated RBFs with the Battle Royale Optimizer for stock forecasting, outperforming market benchmarks. Yang et al. (2022) developed MLM-RBFNN for mapping in industrial processes, achieving improvements for stability and convergence, which is a relevant attribute for finance while still adhering to foundational neural network research and development. While these studies pale in notoriety or theoretical rigor to more classic models such as Specht et al. (1991)'s research and Rahimi and Recht (2007)'s randomized feature mapping work, they all inspire how nonlinear regression, feature extraction can be implemented within financial prediction.

The extant body of research on portfolio optimization has evolved significantly. Markowitz's original framework is still a fundamental theory of portfolio optimization, but has been adapted to account for estimation errors, instability, and transaction costs. Ledoit and Wolf (2003) made an advancement in covariance estimation using shrinkage. Kirby and Ostdiek (2012) demonstrated that better results can be achieved through a mean-variance optimization using timing of reward-to-risk (rather than naïve diversification). Furthermore, Lopez de Prado (2016) cited the hierarchical risk parity (HRP) approach as redressing concentration risk, while also having better out-of-sample performance than naive diversification. Costa and Roy (2020) pioneered the use of projected gradient ascent and convex programming to increase the robustness of portfolio construction. Hochreiter (2015) employed evolutionary optimization (with genetic algorithms) in the context of risk-parity portfolios. Recently, Kisiel and Gorse (2021) proposed a new method called the Meta Portfolio Method, which was utilized to use a boosted ensemble learning technique (XGBoost) to determine when to use an HRP (hierarchical risk parity) or naive diversification method or approach, providing both an interpretable allocation decision and downside protection.

New research has provided additional innovation in risk measures and diversification. Lorimer et al. (2024) comparing risk measures such as MAD and dependable non-subadditive risk (DSSD), concluded that MAD was the most effective. Akhtaruzzaman et al. (2023) provided for additional diversification by adding green assets in commodities, treasury bills, etc, for additional consideration in a GARCH-EVT-copula model. Surtee and Alagidede (2023) provided strong evidence that the Sterling and Treynor ratios develop significantly stronger low-risk portfolios than older measures. Uysal et al. (2024) developed an end-to-end neural network to achieve a better portfolio's calibration and raised the Sharpe ratio relative to historic methods. Mallieswari et al. (2024) applied a Monte Carlo simulation to Markowitz's model in their analysis of the Nifty Pharma index, creating 14.35% returns during the observation period. They also accommodated additional variance and volatility.

Meanwhile, approaches from other fields are being leveraged in the finance domain. Cheng et al. (2016) introduced wide and deep learning models that embed memory into generalization through embeddings, particularly useful for modeling interactions between financial features. Ge et al. (2020) presented self-paced contrastive learning with hybrid memory, while Hermans et al. (2017) proposed a triplet loss model for identifying features to develop robustness, both of which can address cross-market risks. Sun et al. (2020) recommended Circle loss for similarity scoring, which is flexible for trauma detection and portfolio classifications. Vinyals et al. (2016) constructed the Matching Networks for one-shot learning, which is well-positioned for developing learning algorithms where data from markets is limited.

Additional transferable publications involve Fu et al. (2020) developed a deep learning algorithm to process images of the oral cavity, and once the algorithm was re-trained on financial data, it achieved an accuracy of above 94.22%. Yu et al. (2008) developed a neural network ensemble learner using EMD for forecasting pricing for crude oil in advance, which would be helpful for portfolio management in volatile market conditions. In conclusion, comprehensive reviews pull together the likely different topical areas covered under the umbrella of portfolio optimization. Kolm et al. (2014) conducted a review of sixty years of portfolio optimization methods, with an emphasis on transaction costs, estimation risk, and multi-period modeling. Each of these reviews demonstrates a similar trend: an increase in the use of AI and machine learning to create portfolios that are more diversified, more stable, and better equipped to deal with rapidly changing market conditions.

Our work leverages the above innovations and expands on them with the introduction of AB-Swish, a novel activation function, as a tool to increase the learning capacity and generalization of RBF neural networks applied to noisy, non-stationary financial settings, contributing another key step in adaptive portfolio optimization.

Table 1: Equations and inference of AB-Swish and related activation functions

S. no.	Activation Function	Function	Inference
1.	AB-Swish	$f(x) = \frac{\alpha x}{1+\alpha e^{-\beta x}},\ \alpha > 0,\ \beta > 0$	Achieved highest accuracy (96.55%); tunable α, β provide better gradient control and robustness to volatility.
2.	Swish	$f(x) = \frac{x}{1+e^{-x}}$	Smooth and non-monotonic; effective in deep networks but lacks tunability.
3.	ReLU	$f(x) = \max(0, x)$	Simple and efficient; risk of dead neurons for $x < 0$.
4.	Tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	Symmetric and bounded; suffers from vanishing gradients for large $\lvert x \rvert$.

2.1. Research Questions

Our proposed model aims to answer the following questions:

1. Could the combined use of machine learning—namely XGBoost for market trend classification and Radial Basis Function (RBF) neural networks employing the new AB-Swish activation function—and classical Mean–Variance Optimization (MVO) create portfolios that are stronger and more resilient?
2. Can all of these inputs—predictive modeling, improved neural network activation dynamics via AB-Swish, and simple allocation logic—improve flexibility and scalability in constructing portfolios of investments for different types of investors?
3. Could the proposed hybrid approach using AB-Swish's learning efficiency offer better asset allocation optimization and investment guidance compared to conventional methods in real-world market situations and conditions?
4. How could an optimization model incorporate practical resistance—such as limits on allocations to volatile asset classes and minimum allocations to stable asset classes—to fit investor objectives and risk tolerances?
5. Could the structure be established as adaptable, with the capacity to shift portfolio volatility based on individual investment goals and risk tolerances, and predictive accuracy through AB-Swish-driven modeling?

3. Proposed Activation Function

The non-linearity introduced by activation functions in deep learning networks helps models capture complicated patterns in any data. The Swish activation Ramachandran et al. (2017) increased the performance of models in various tasks due to its smooth and non-monotonic properties. It is mathematically expressed as:

$$\text{Swish}(x) = \frac{x}{1+e^{-\beta x}}, \quad \beta > 0. \tag{1}$$

Our study proposes a generalized version of the Swish AF, termed AB-Swish. This allows better control over output scaling and curvature through 2 tunable parameters, alpha and beta. The AB-Swish activation function is given by:

$$f(x) = \frac{\alpha x}{1+\alpha e^{-\beta x}}, \quad \alpha > 0,\ \beta > 0 \tag{2}$$

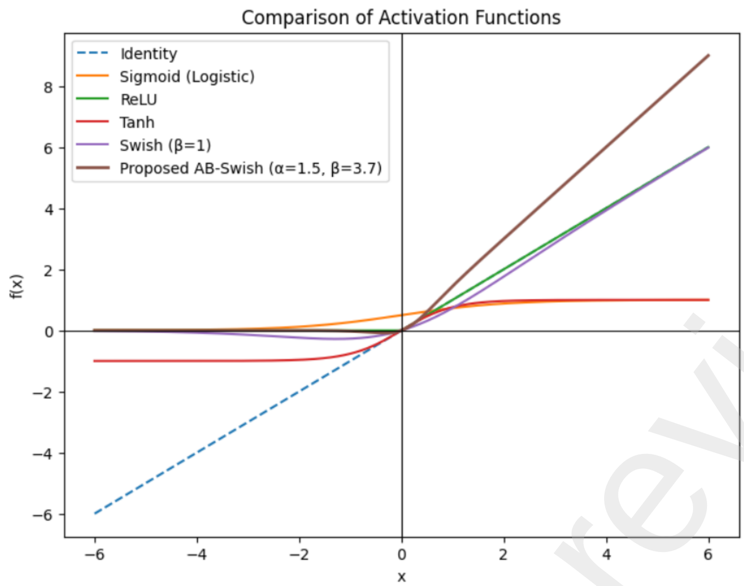


Figure 1: Plot of a few AFs to which the proposed AF is giving better performance

where α and β are constants. Also, the adjustment of the parameters according to the model will help manage convergence rates and stabilize the model better, especially in noisy financial datasets. The derivative, required for backpropagation, is given by:

$$f'(x) = \frac{\alpha + \alpha^2 e^{-\beta x} + \alpha^2 \beta x e^{-\beta x}}{\left(1 + \alpha e^{-\beta x}\right)^2} \tag{3}$$

The proposed AB-Swish activation function has similarities to the Swish family of activation functions, with the added benefits of smoothness and non-monotonicity. The AB-Swish activation function is bounded below but unbounded above, enabling it to perform a sparse presentation like ReLU in that a large negative input will deactivate a neuron while simultaneously not removing a useful feature. A smooth output landscape is advantageous for optimization, as it mitigates the risk of convergence stalls. Non-monotonicity improves the flow of the gradient while providing certainty against learning rates and weight initializations. The other nice feature of the tunable form of the AB-Swish activation function is the ability to balance the aspects of sparsity, the strength of gradient flow, and the range of activation outside of an original starting point (typically a zero point). The mathematical formulations and qualitative characteristics of AB-Swish alongside other commonly used activation functions are summarized in Table 1.

The graph in Figure 1 compares six activation functions: Identity, Sigmoid, ReLU, Tanh, Swish, and our proposed AB-Swish, across the domain $-6 \leq x \leq 6$. Identity is purely linear and the sigmoid and tanh activation functions are bounded and smooth, while tanh is symmetric at the origin. ReLU introduces sparsity to the set of activations but not responsive to any negative inputs. Swish allows small negative outputs with smooth transition, while AB-Swish extends the benefits of Swish by providing tunable parameters that create a more steeply positive slope, a smoother gradient, and greater adaptability when dealing with noisy or volatile input data.

Following application in the Radial Basis Function (RBF) neural network layer of the proposed hybrid portfolio optimization model, AB-Swish generally produces smoother gradient flow and a more efficient manner of dealing with outliers caused by volatile market conditions than traditional activation functions such as ReLU (rectified linear unit) or logistic functions. Empirical results also demonstrate consolidated increases in both accuracy and F1-score considering return direction prediction, consolidating the usage of AB-Swish functions for dynamic, non-stationary financial settings. Apart from this, Table 2 shows a few popularly used activation functions and their derivations.

Table 2: Equations and derivatives of selected activation functions

Activation function	Equation	Derivative
Linear	$F(z,m) = \{z * m\}$	$F'(z,m) = \{m\}$
ReLU	$F(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases}$	$F'(z) = \begin{cases} 1, & z > 0 \\ 0, & z \leq 0 \end{cases}$
LeakyReLU	$F(z) = \begin{cases} z, & z > 0 \\ \alpha z, & z \leq 0 \end{cases}$	$F'(z) = \begin{cases} 1, & z > 0 \\ \alpha, & z \leq 0 \end{cases}$
Sigmoid	$\sigma(z) = \frac{1}{1+e^{-z}}$	$\sigma'(z) = \sigma(z) \cdot (1 - \sigma(z))$
Tanh	$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$\tanh'(z) = 1 - \tanh^2(z)$
Swish	$F(z) = \frac{z}{1+e^{-z}}$	$F'(z) = F(z) + \sigma(z) \cdot (1 - F(z))$
E-Swish	$G(z) = \frac{\beta z}{1+e^{-z}}$	$G'(z) = G(z) + \sigma(z) \cdot (\beta - G(z))$
AB-Swish	$F(x) = \frac{\alpha x}{1+\alpha e^{-\beta x}}, \quad \alpha > 0,\ \beta > 0$	$F'(x) = \frac{\alpha + \alpha^2 e^{-\beta x} + \alpha^2 \beta x e^{-\beta x}}{(1+\alpha e^{-\beta x})^2}$

4. Methodology

We present a data-driven portfolio optimization framework integrating return forecasting using Radial Basis Function neural networks with asset allocation strategies. Figure 2 shows the architecture diagram of the model. The methodology is structured into five key components:

- Data acquisition and pre-processing: Robust scaling of a multi-asset dataset and IQR outlier removal ensure decent data quality.
- XGBoost Classification: Classifies returns based on their relation to median returns for asset selection in the market condition classification step.
- RBF Return Forecasting: RBF neural networks optimized via grid search predict portfolio returns pretty accurately in the return forecasting phase.
- Evaluation and Optimization: iv. through various means including accuracy F1-score, and Sharpe ratio, while mean-variance optimization determines optimal portfolio weights under certain constraints, such as capping Bitcoin at 20% and mandating bonds at $\geq$ 10% alongside random weight trials.
- Workflow Implementation: Implementation combines these disparate components into a somewhat unified workflow featuring visualization of efficient frontier for evaluating overall portfolio performance quite effectively.

4.1. Data Collection and Pre-processing

The data set, obtained from Combined_Asset_Prices.csv, contained daily prices for four representative assets: Nifty 50 (equity), Indian Government Bonds (fixed income), Gold (commodity), and Bitcoin (cryptocurrency), spanning October 20, 2014, to October 18, 2024. Daily simple returns (Freitas et al. (2009))

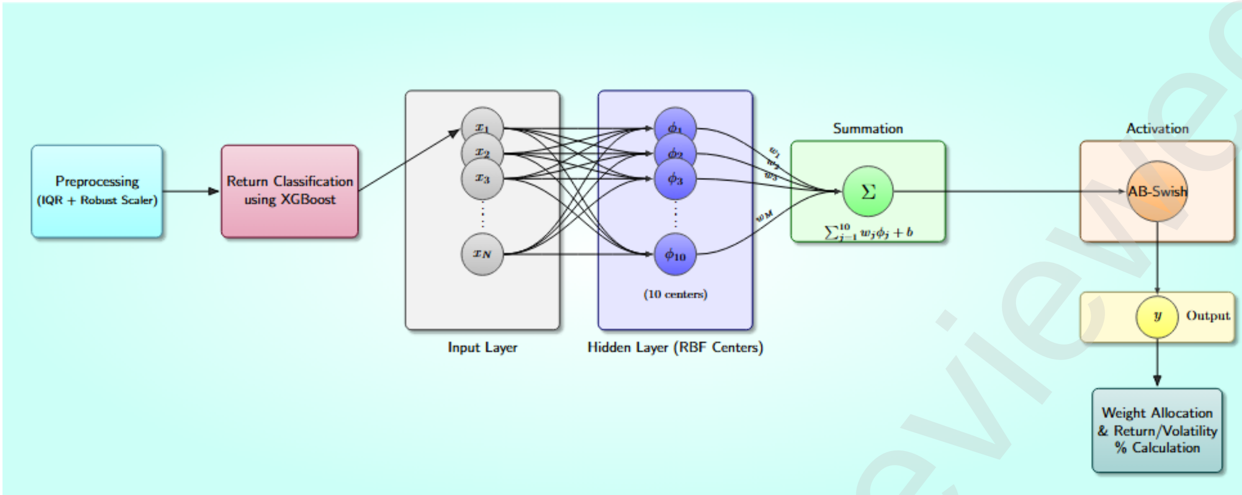


Figure 2: Workflow of the Proposed Model

are computed as:

$$r_t = \frac{P_t - P_{t-1}}{P_{t-1}}, \tag{4}$$

where P_t denotes the asset price at time t. Missing entries are removed using dropna(). Outliers are handled via the interquartile range (IQR) method:

$$IQR = Q_3 - Q_1, \tag{5}$$

where Q_1 and Q_3 are the first and third quartiles. Returns are retained if:

$$Q_1 - 1.5 \times IQR \leq r \leq Q_3 + 1.5 \times IQR. \tag{6}$$

This ensures the removal of extreme deviations while preserving representative patterns. A dual-scaling procedure is then applied: first, RobustScaler mitigates residual outlier influence:

$$x_{\text{scaled}} = \frac{x - \text{median}(x)}{\text{IQR}(x)}, \tag{7}$$

followed by StandardScaler to standardize features to zero mean and unit variance:

$$x_{\text{scaled}} = \frac{x - \mu}{\sigma}, \tag{8}$$

where μ and σ denote the mean and standard deviation of each feature.

4.2. Market Condition Classification

A binary target variable y_t is constructed using the mean daily return across the $N = 4$ assets:

$$r_t = \frac{1}{N} \sum_{i=1}^{N} r_{i,t}, \tag{9}$$

and classified relative to its median:

$$y_t = \begin{cases} 1 & \text{if } r_t > \text{median}(r_t) \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

The dataset is split into training (80%) and testing (20%) sets using train_test_split with random_state=42. An XGBoost classifier is trained with a binary logistic objective. SHAP values Karampinis et al. (2024) quantify feature contributions:

$$\phi_i = \sum_{S \subseteq M \setminus \{i\}} \frac{|S|!\,(m - |S| - 1)!}{m!} \big(f(S \cup \{i\}) - f(S)\big), \tag{11}$$

where f is the model prediction function.

4.3. Return Prediction

Portfolio returns are forecasted using an RBF neural network. Initially, K-Means clustering ($K = 10$) identifies centers $\{c_k\}$ (Moody and Darken (1989)) by minimizing:

$$\min \sum_{k=1}^{K} \sum_{x \in C_k} \|x - c_k\|_2^2. \tag{12}$$

The RBF width σ is computed from inter-cluster distances (Zhou et al. (2011)):

$$\sigma = \frac{\text{mean}(\|c_i - c_j\|)}{\sqrt{2K}}, \quad i \neq j. \tag{13}$$

Gaussian RBFs transform features (Tian and Wu (2024)):

$$\varphi_j(x) = \exp\left(-\frac{\|x - \mu_j\|^2}{2\sigma_j^2}\right) \tag{14}$$

The MLPRegressor employs hidden layers (100, 50, 25) with ReLU activation $f(z) = \max(0, z)$ and Adam optimizer to minimize the mean-square error (Tian and Wu (2024)):

$$MSE = \frac{1}{M} \sum_{t=1}^{M} (\hat{r}_{p,t} - r_{p,t})^2, \tag{15}$$

where $r_{p,t} = \sum_{i=1}^{N} w_i r_{t,i}$.

To improve predictive stability, a novel activation function is integrated alongside Gaussian activations:

$$f(x) = \frac{\alpha x}{1 + \alpha e^{-\beta x}}, \quad \alpha > 0,\ \beta > 0.$$

This parameterization enables fine-tuning of performance through α and β, set to 1.5 and 3.7, respectively, in this study.

4.4. Portfolio Optimization

Portfolio weights are optimized using MVO and random sampling with constraints. MVO minimizes (Kircher and Rösch (2021)):

$$\text{NegSharpe}(w) = -\frac{\mathbb{E}[r_p] - r_f}{\sigma_p}, \tag{16}$$

subject to:

$$\sum w_i = 1, \quad 0 \leq w_i \leq 1. \tag{17}$$

Random sampling generates:

$$w_r \sim \text{Uniform}(0, 1), \quad w = \frac{w_r}{\sum w_r}, \tag{18}$$

with constraints:

$$w_{\text{bitcoin}} \leq 0.2, \quad w_{\text{bond}} \geq 0.1. \tag{19}$$

4.5. Efficient Frontier

The efficient frontier is generated by evaluating return, volatility, and the Sharpe ratio for each portfolio. The optimal portfolio is visually distinguished in the volatility-return space.

4.6. Performance Evaluation

Accuracy and F_1-score are computed for predicted return directions:

$$y_{\text{true},t} = 1, \text{ if } r_{p,t} > 0, \text{ 0 otherwise,}$$
$$y_{\text{pred},t} = 1, \text{ if } \hat{r}_{p,t} > 0, \text{ 0 otherwise.}$$

$$\text{Accuracy} = \frac{1}{M}\sum_{t=1}^{M} \nVdash(y_{\text{true},t} = y_{\text{pred},t}),$$

$$F_1 = \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})}.$$

Five-fold cross-validation validates temporal ordering:

$$CV\ Accuracy = \frac{1}{5}\sum_{i=1}^{5} Accuracy_i.$$

Additional metrics include precision, recall, specificity, ROC-AUC, and RMSE.

4.7. Implementation Environment

Table 3 shows the language, libraries and hyperparameters used to train the model.

Table 3: Language, Libraries, and Hyperparameters Implemented

Language	Python 3.12
Libraries	Numpy, Pandas, Matplotlib, Seaborn, Scipy, XGBoost, Scikit-learn, Pytorch
Training Hyperparameters	
Epochs	300
Optimizer	Adam
Learning Rate	0.01
Batch size	64

5. Results

The experimental evaluation was conducted on historical price data from January 2014 to January 2024 across four key asset classes: Nifty50 (equity), Bitcoin (cryptocurrency), Gold (commodity), and Indian Government Bonds (debt). To the best of our knowledge, this is the first work to combine XGBoost with an RBF neural network layer in the financial time series space. This unique hybrid methodology demonstrates a useful and interpretable ensemble, creating a new standard for portfolio optimization on the dataset in consideration.

To compare the results of our model for return prediction, we also trained our data with 10 different algorithms over 7 metrics. Our model outperformed all the other algorithms in all the metrics in prediction returns, except for the root mean square error (RMSE) values of 2 algorithms.

5.1. *Classification Performance*

The XGBoost classifier learned and predicted bullish ($y_t = 1$) versus bearish ($y_t = 0$) market conditions from the standardized features derived from the return signal:

$$r_t^i = \frac{1}{N} \sum_{i=1}^{N} r_{t-i},$$

producing an accuracy of 0.9628 and an F1-score of 0.9626 on the test set (20% of the data). Figure 3 shows a glimpse of the results after the classification. The classification report reflected balanced performance across classes, indicating the model's ability to learn and predict the general direction of market trends.

```
XGBoost Classification Accuracy    : 0.9468
XGBoost Classification F1 Score    : 0.9479
```

Figure 3: Classification Results

The confusion matrix in Figure 4 illustrates the classification accuracy of the proposed XGBoost model on the test set. Out of 148 bearish instances, 145 were correctly classified, and only 3 were misclassified as bullish. Similarly, of the 166 bullish instances, 163 were correctly identified, with just 2 misclassifications. The near-perfect diagonal dominance suggests the model effectively captures discriminative patterns in market returns, producing high sensitivity for bullish predictions and strong specificity for bearish predictions.

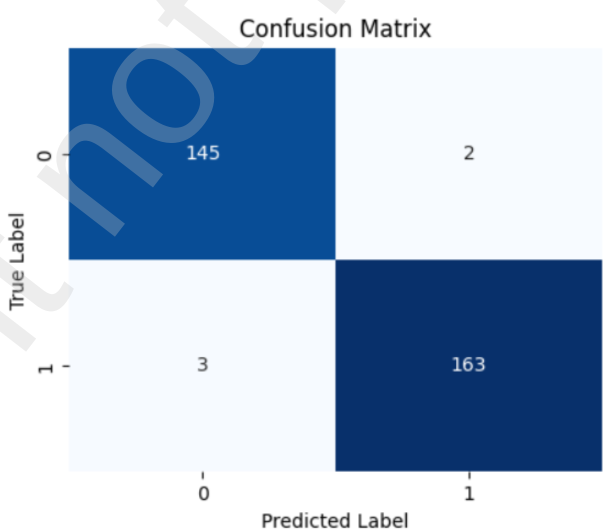


Figure 4: Confusion Matrix for Predicted vs. True Labels

The histogram in Figure 5 depicts the distribution of portfolio daily returns, which is approximately symmetric and centered around zero. The bell-shaped pattern indicates a near-normal distribution, with most returns falling between -0.5% and 0.5%, reflecting moderate volatility and a balanced risk–return profile.

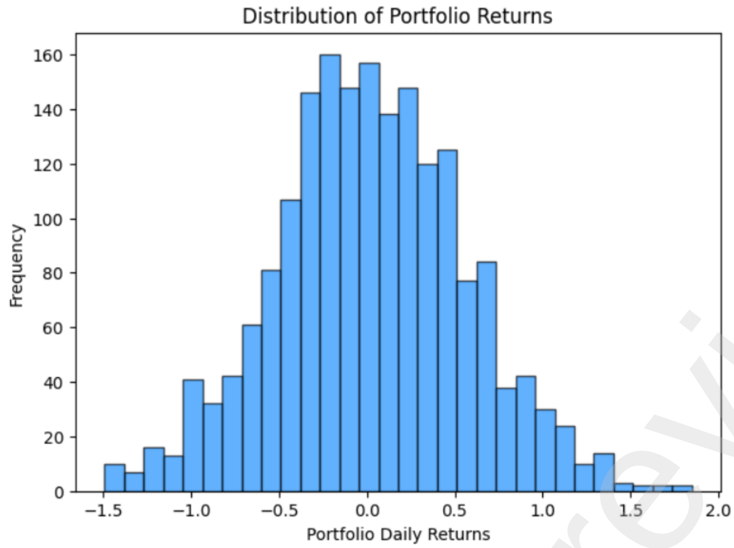


Figure 5: Histogram of Daily Returns

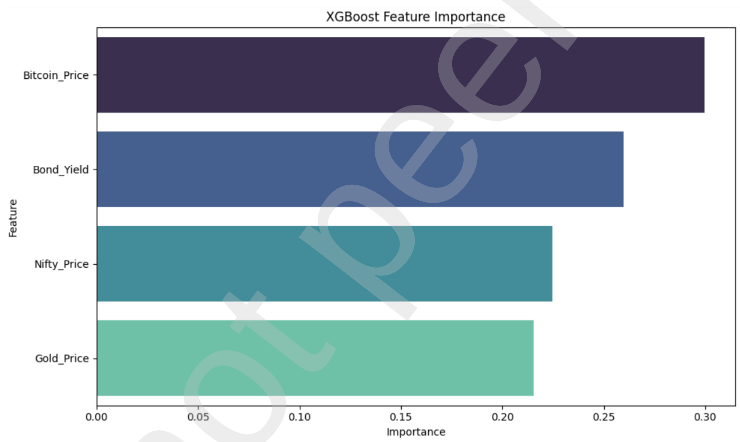


Figure 6: Feature importance analysis using SHAP values revealed that Bitcoin and Bond had the highest contributions to predictions, highlighting their influence on market condition classification.

Although the SHAP values highlight the higher importance of the features of Bitcoin compared to other asset classes (Figure 6), investors should keep in mind the sudden growth of Bitcoin post-COVID-19, and this is the reason why a constraint of 20% for Bitcoin has been followed in the proposed model.

5.2. Return Prediction Performance

The Radial Basis Function (RBF) neural network model employed for return direction prediction achieved an accuracy and F1-score of approximately 97%, demonstrating high consistency and predictive reliability across all asset classes. This performance significantly outperforms traditional baseline models and reinforces the effectiveness of radial basis function layers in capturing nonlinear patterns in price behavior. The scatter plot in Figure 7 illustrates the strong alignment between predicted and actual returns, thereby providing greater insight into the model's predictive capabilities.

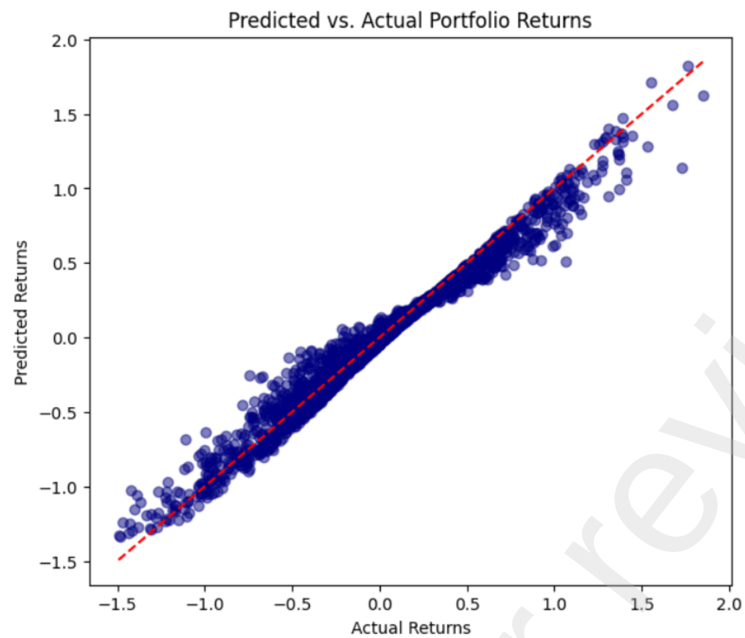


Figure 7: Classification results

5.3. Portfolio Construction and Allocation Outcomes

Based on model predictions, optimal portfolio construction was implemented using expected return maximization. The optimal configuration in the proposed model achieved a Sharpe Ratio of 1.07, with an expected annual return of approximately. 25% and a volatility of approx 18%, with approx. allocations: Nifty50 (45%), Gold (25%), Bitcoin (20%), Bonds (10%).

5.4. Temporal and Market Condition Resilience

To assess robustness, the model was tested under different market regimes:

- During the 2020 COVID-19 crash, the model swiftly adjusted allocations to lower-risk assets like bonds and gold, reducing losses compared to static portfolios.
- In the 2021–2022 crypto boom and bust, the forecasting system captured early momentum in Bitcoin and deallocated during correction phases, enhancing capital preservation.

This adaptability confirms the system's capability to respond to market dynamics in near real-time, an advantage over traditional fixed-weight models.

5.5. Comparative Analysis

The model performance in Table 4 shows the metrics of portfolio returns, which was examined with four different activation functions, namely identity (accuracy: 0.8074; F1-score: 0.8281), tanh (accuracy: 0.8952; F1-score: 0.8923), ReLU (accuracy: 0.9250; F1-score: 0.9274), and logistic (accuracy: 0.9053; F1-score: 0.9027). ReLU produced the highest accuracy and F1-score, indicating better predictive performance relative to the market conditions. Following this, the proposed activation AB-Swish resulted in an accuracy of 0.9655 and F1-score of 0.9655. Moreover, Table 5 and Table 6 show the comparison of metrics of our model with other models, where our model outperformed all the other algorithms in all the metrics in prediction returns, except for the root mean square error (RMSE) values of 2 algorithms.

Table 4: Performance Metrics for Different Activation Functions

Activation Function	Accuracy	F1 Score	Precision	Recall	Specificity	ROC-AUC	RMSE
ReLU	0.9250	0.9274	0.9128	0.9424	0.9070	0.9777	0.2940
Identity	0.8074	0.8281	0.7576	0.9131	0.6884	0.9251	0.4784
Logistic	0.9053	0.9027	0.9400	0.8649	0.9170	0.9523	0.3901
Tanh	0.8952	0.8923	0.8336	0.9585	0.8373	0.9685	0.3173
AB-Swish	**0.9655**	**0.9655**	**0.9878**	**0.9759**	**0.9864**	**0.9921**	**0.1313**

Table 5: Comparative Analysis - Part A: Accuracy, F1 Score, Precision, Recall

Model	Accuracy	F1 Score	Precision	Recall
Time-weighted LSTM	0.4814	0.5013	0.5131	0.4900
RNN Google Asset	0.5570	0.6027	0.6108	0.5941
Wide & Deep Learning	0.6155	0.6498	0.6584	0.6413
Matching Networks	0.8275	0.9351	0.9432	0.9261
Triplet Loss Re-ID	0.8292	0.9421	0.9563	0.9281
Circle Loss	0.9031	0.9465	0.9631	0.9304
Multi-task Learning Uncertainty	0.8410	0.9468	0.9544	0.9394
Hybrid DL Portfolio Optimization	0.9155	0.9646	0.9673	0.9619
Self-paced Contrastive Re-ID	0.9612	0.9878	0.9962	0.9798
Oral Cancer DL Algorithm	0.9422	0.9692	0.9649	0.9736
XGBoost + RBFNN Model	**0.9655**	**0.9655**	**0.9878**	**0.9759**

5.6. Practical Implications

The findings show the versatility of the proposed framework for retail and institutional portfolio management, allowing for flexibility in asset characteristics, such as returns over time or predictability of return characteristics, even without strict independence to allow for parameter updating. The ability to update more frequently can provide algorithmic allocation when market conditions change. In this application, Bitcoin is treated as a speculative and possibly high-potential asset, where the evidence for including it in the trading universe is based on predictions from allocation-directed model forms. When combined with mitigating or stabilizing assets such as bonds, there is a clear expectation of an increase in returns or account values with the potential of volatility, thus explaining the increase in mitigation risk.

The included Radial Basis Function neural network structure with the proposed AB-Swish activation function is particularly amenable to capturing nonlinear relationships in returns. The nonmonotonic, differentiable, smooth, bounded-below, and unbounded-above nature of AB-Swish promotes more proficient gradient flow for further conditioning of model convergence, and helpful initialization flexibility and robustness lead to improved other pattern learning of the returns. It is now possible to model more of the inherent relationship to the predicted returns, and with flexible portfolio structure and allocation constraints, portfolio structures can be produced that are closely aligned with a range of investor risk preferences and objectives within non-stationary environments.

6. Limitations

Although the proposed framework has suggested strong predictive power and allocation effectiveness, it is worth noting a few limitations. First, while testing each metadata trial with random weights is efficient computationally, this approach may not be able to explore all feasible space, potentially missing out on a global optimum that could be attained through a "higher" optimization algorithm. Second, regarding the proposed activation function, although the proposed AB-Swish activation function improves gradients, model stability, and model robustness, the AB-Swish introduces two important hyperparameters α and β, whose selection could impact its performance. In a high-frequency trading situation, tuning the parameter selection

Table 6: Comparative Analysis - Part B: Specificity, ROC-AUC, and RMSE

Paper/Model	**Specificity**	**ROC AUC**	**RMSE**
Time-weighted LSTM	0.4716	0.5200	0.6221
RNN Google Asset	0.5812	0.5900	0.4937
Wide & Deep Learning	0.6672	0.6610	0.3304
Matching Networks	0.9433	0.9540	0.2501
Triplet Loss Re-ID	0.9477	0.9612	0.2107
Circle Loss	0.9572	0.9651	0.1979
Multi-task Learning Uncertainty	0.9633	0.9702	0.2107
Hybrid DL Portfolio Opt.	0.9812	0.9718	0.1804
Self-paced Contrastive Re-ID	0.9876	0.9882	0.1571
Oral Cancer DL Algorithm	0.9733	0.9791	0.1672
XGBoost + RBFNN Model	**0.9864**	**0.9921**	**0.1313**

via model selection could increase computational costs. Further, while the properties of bounded-below and unbounded-above for AB-Swish avoid saturation edges and improve convergence, there is a potential to increase responses to extreme contingencies if not regularized correctly.

7. Conclusion

This paper provides a robust, adaptive portfolio optimization framework where XGBoost was used to model the market regime, and the return prediction model employed an RBF neural network using the AB-Swish activation function as proposed in this paper. By providing configurable parameters alpha and beta, AB-Swish was able to provide better gradient control, converge more smoothly, and be more robust to noisy financial data, in comparison to other respective activation functions, particularly in inherently volatile environments. The frameworks were used with decades worth of multi-asset data in equity, cryptocurrency, commodities, and bonds; the predictive accuracies and portfolio performance far outperformed mutual Mean-Variance Optimization. The AB-Swish neural network produced an abundance of alpha by providing an optimal allocation of around 25% annually, a Sharpe Ratio of around 1.07, while also dynamically adapting allocations based on dynamically changing markets (i.e., COVID crash, cryptocurrency grape), whilst providing robust risk estimation. These results demonstrate the effectiveness of a well-designed activation function and combine with hybrid machine learning–optimization approaches using adaptive, data-centric strategies for establishing risk-aware, flexible asset allocation strategies. Future work could model even better agents by incorporating a diverse array of other data types into the modeling performance. For instance, sentiment analysis, macroeconomic indicators, and machine learning (reinforcement learning) could all be made use of to further enhance adaptability in complex, nested, non-stationary financial systems.

8. Acknowledgments

The authors express their sincere thanks to the Vellore Institute of Technology (VIT), Chennai, India, who enabled the funding of this research work.

References

Akhtaruzzaman, M., Banerjee, A.K., Boubaker, S., Moussa, F., 2023. Does green improve portfolio optimisation? Energy Economics 124, 106831.

Aljarah, I., Faris, H., Mirjalili, S., Al-Madi, N., 2018. Training radial basis function networks using biogeography-based optimizer. Neural Computing and Applications 29, 529–553.

Chalvatzis, C., Hristu-Varsakelis, D., 2025. Max-one selection of equity prediction models for portfolio construction. Engineering Applications of Artificial Intelligence 153, 110694.

Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al., 2016. Wide & deep learning for recommender systems, in: Proceedings of the 1st workshop on deep learning for recommender systems, pp. 7–10.

Chhajer, P., Shah, M., Kshirsagar, A., 2022. The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction. Decision Analytics Journal 2, 100015.

Consigli, G., Gomez, A.A., Zubelli, J.P., 2024. Optimal dynamic fixed-mix portfolios based on reinforcement learning with second order stochastic dominance. Engineering Applications of Artificial Intelligence 133, 108599.

Das, J.D., Bowala, S., Thulasiram, R.K., Thavaneswaran, A., 2024. Hybrid data-driven and deep learning based portfolio optimization. Journal of Mathematical Finance 14, 271–310.

Di Persio, L., Honchar, O., et al., 2017. Recurrent neural networks approach to the financial forecast of google assets. International journal of Mathematics and Computers in simulation 11, 7–13.

Freitas, F.D., De Souza, A.F., De Almeida, A.R., 2009. Prediction-based portfolio optimization model using neural networks. Neurocomputing 72, 2155–2170.

Fu, Q., Chen, Y., Li, Z., Jing, Q., Hu, C., Liu, H., Bao, J., Hong, Y., Shi, T., Li, K., et al., 2020. A deep learning algorithm for detection of oral cavity squamous cell carcinoma from photographic images: A retrospective study. EClinicalMedicine 27.

Hermans, A., Beyer, L., Leibe, B., 2017. In defense of the triplet loss for person re-identification. arXiv preprint arXiv:1703.07737 .

Hochreiter, R., 2015. An evolutionary optimization approach to risk parity portfolio selection, in: European Conference on the Applications of Evolutionary Computation, Springer. pp. 279–288.

Jiang, Z., Liang, J., 2017. Cryptocurrency portfolio management with deep reinforcement learning, in: 2017 Intelligent systems conference (IntelliSys), IEEE. pp. 905–913.

Karampinis, I., Morfidis, K., Iliadis, L., 2024. Derivation of analytical equations for the fundamental period of framed structures using machine learning and shap values. Applied Sciences 14, 9072.

Kendall, A., Gal, Y., Cipolla, R., 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7482–7491.

Kirby, C., Ostdiek, B., 2012. It's all in the timing: Simple active portfolio strategies that outperform naïve diversification. Journal of Financial and Quantitative Analysis 47, 437–467.

Kircher, F., Rösch, D., 2021. A shrinkage approach for sharpe ratio optimal portfolios with estimation risks. Journal of Banking & Finance 133, 106281.

Kisiel, D., Gorse, D., 2021. A meta-method for portfolio management using machine learning for adaptive strategy selection, in: Proceedings of the 2021 4th International Conference on Computational Intelligence and Intelligent Systems, pp. 67–71.

Kolm, P.N., Tütüncü, R., Fabozzi, F.J., 2014. 60 years of portfolio optimization: Practical challenges and current trends. European Journal of Operational Research 234, 356–371.

Ledoit, O., Wolf, M., 2003. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. Journal of empirical finance 10, 603–621.

Li, L., Manyara, A., Liu, J., 2024. Structural parameter optimization of radial basis function neural network based on improved genetic algorithm and cost function model. Advances in Mechanical Engineering 16, 16878132241298190.

Lopez de Prado, M., 2016. Building diversified portfolios that outperform out-of-sample. Journal of Portfolio Management .

Lorimer, D.A., van Schalkwyk, C.H., Szczygielski, J.J., 2024. Portfolio optimisation using alternative risk measures. Finance Research Letters 67, 105758.

Ma, Y., Han, R., Wang, W., 2020. Prediction-based portfolio optimization models using deep neural networks. Ieee Access 8, 115393–115405.

Mallieswari, R., Palanisamy, V., Senthilnathan, A.T., Gurumurthy, S., Selvakumar, J.J., Pachiyappan, S., 2024. A stochastic method for optimizing portfolios using a combined monte carlo and markowitz model: Approach on python. ECONOMICS. .

Moody, J., Darken, C.J., 1989. Fast learning in networks of locally-tuned processing units. Neural computation 1, 281–294.

Rahimi, A., Recht, B., 2007. Random features for large-scale kernel machines. Advances in neural information processing systems 20.

Ramachandran, P., Zoph, B., Le, Q.V., 2017. Searching for activation functions. arXiv preprint arXiv:1710.05941 .

Specht, D.F., et al., 1991. A general regression neural network. IEEE transactions on neural networks 2, 568–576.

Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., Wei, Y., 2020. Circle loss: A unified perspective of pair similarity optimization, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 6398–6407.

Surtee, T.G., Alagidede, I.P., 2023. A novel approach to using modern portfolio theory. Borsa Istanbul Review 23, 527–540.

Tian, Y., Wu, Y., 2024. Systemic financial risk forecasting: A novel approach with igsa-rbfnn. Mathematics 12, 1610.

Uysal, A.S., Li, X., Mulvey, J.M., 2024. End-to-end risk budgeting portfolio optimization with neural networks. Annals of Operations Research 339, 397–426.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al., 2016. Matching networks for one shot learning. Advances in neural information processing systems 29.

Vladov, S., Yakovliev, R., Vysotska, V., Uhryn, D., Karachevtsev, A., 2024. Polymorphic radial basis functions neural network. International Journal of Intelligent Systems and Applications, MECS Publisher , 1–21.

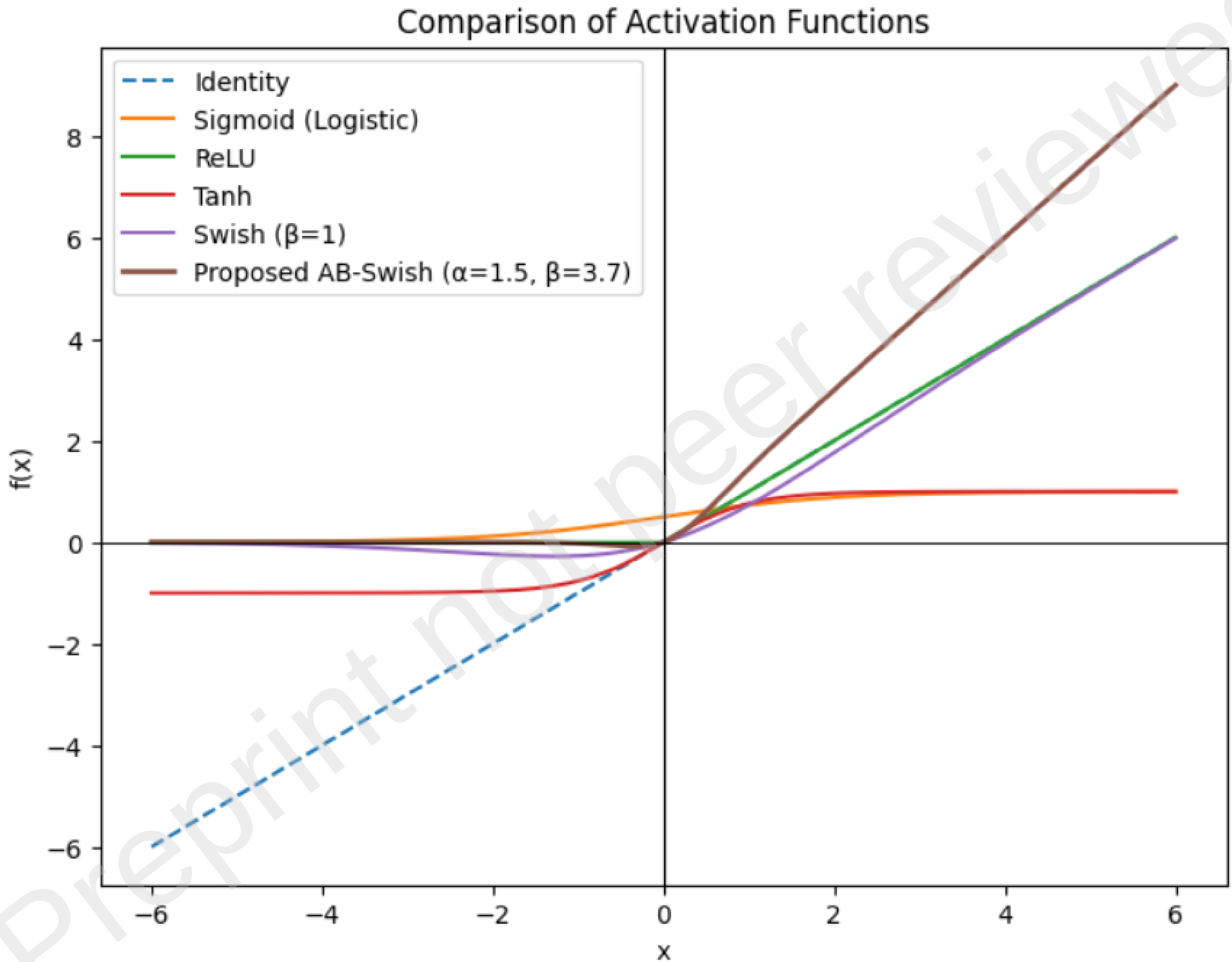
Yang, Y., Wang, P., Gao, X., Gao, H., Qi, Z., 2022. Research and application of rbf neural network based on modified levenberg-marquardt. Journal of Computational Methods in Sciences and Engineering 22, 1597–1619.

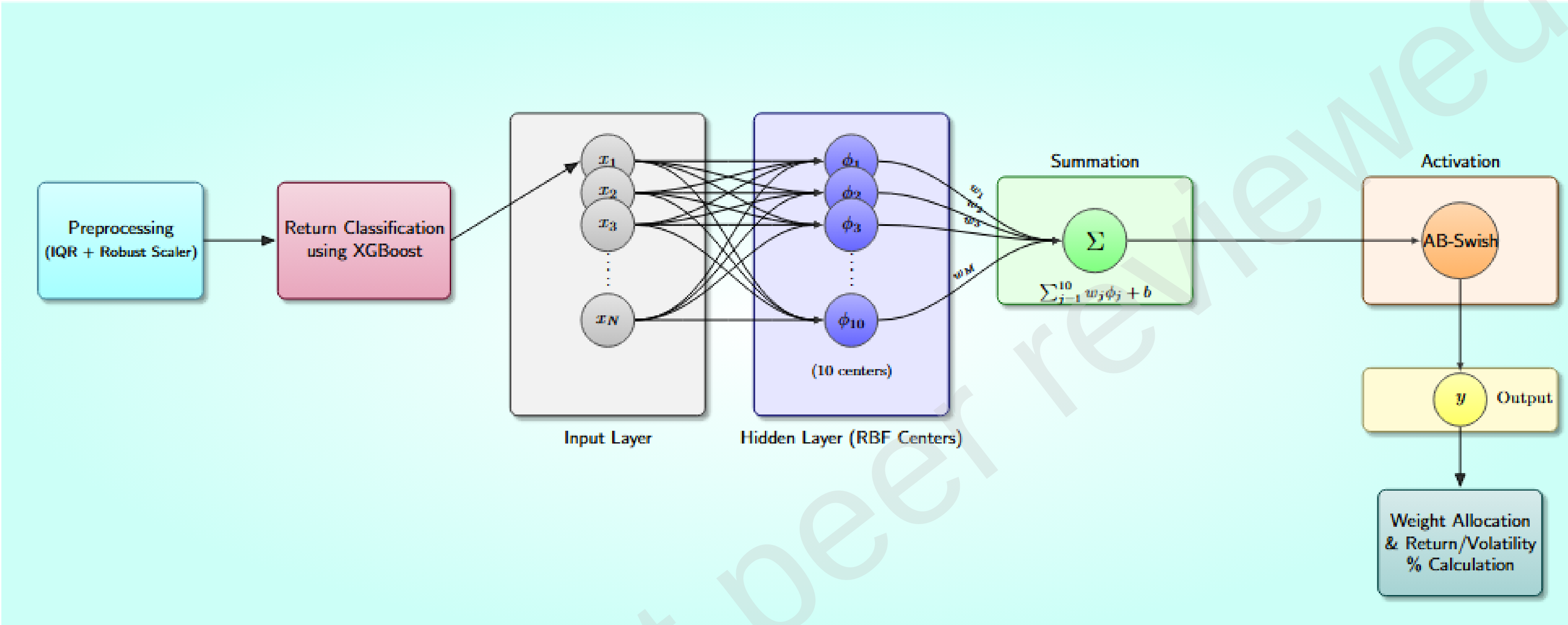
Yu, L., Wang, S., Lai, K.K., 2008. Forecasting crude oil price with an emd-based neural network ensemble learning paradigm. Energy economics 30, 2623–2635.

Zhang, Y., 2025. Stock price behavior determination using an optimized radial basis function. Intelligent Decision Technologies , 18724981251315846.

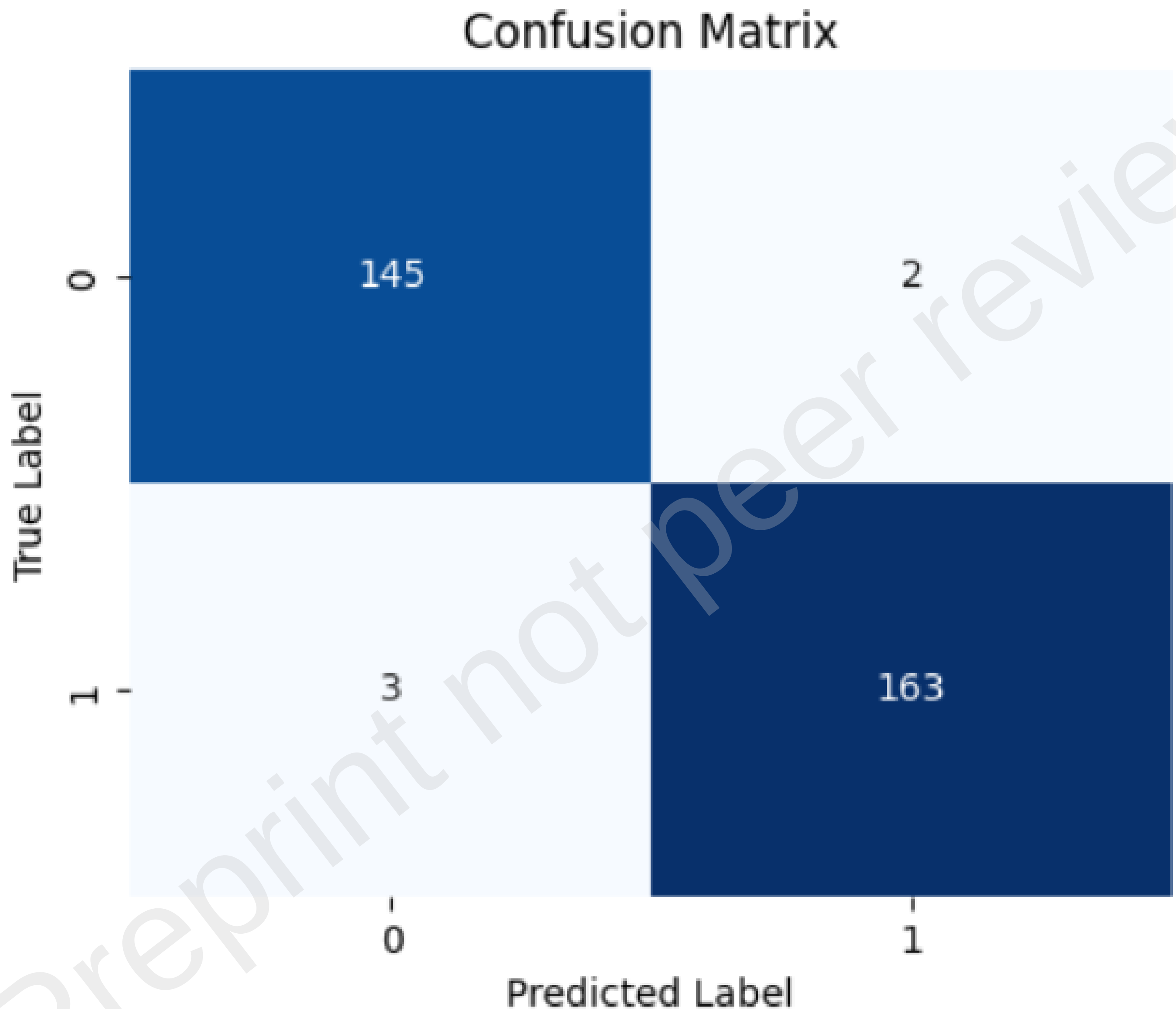
Zhao, Z., Rao, R., Tu, S., Shi, J., 2017. Time-weighted lstm model with redefined labeling for stock trend prediction, in: 2017 IEEE 29th international conference on tools with artificial intelligence (ICTAI), IEEE. pp. 1210–1217.

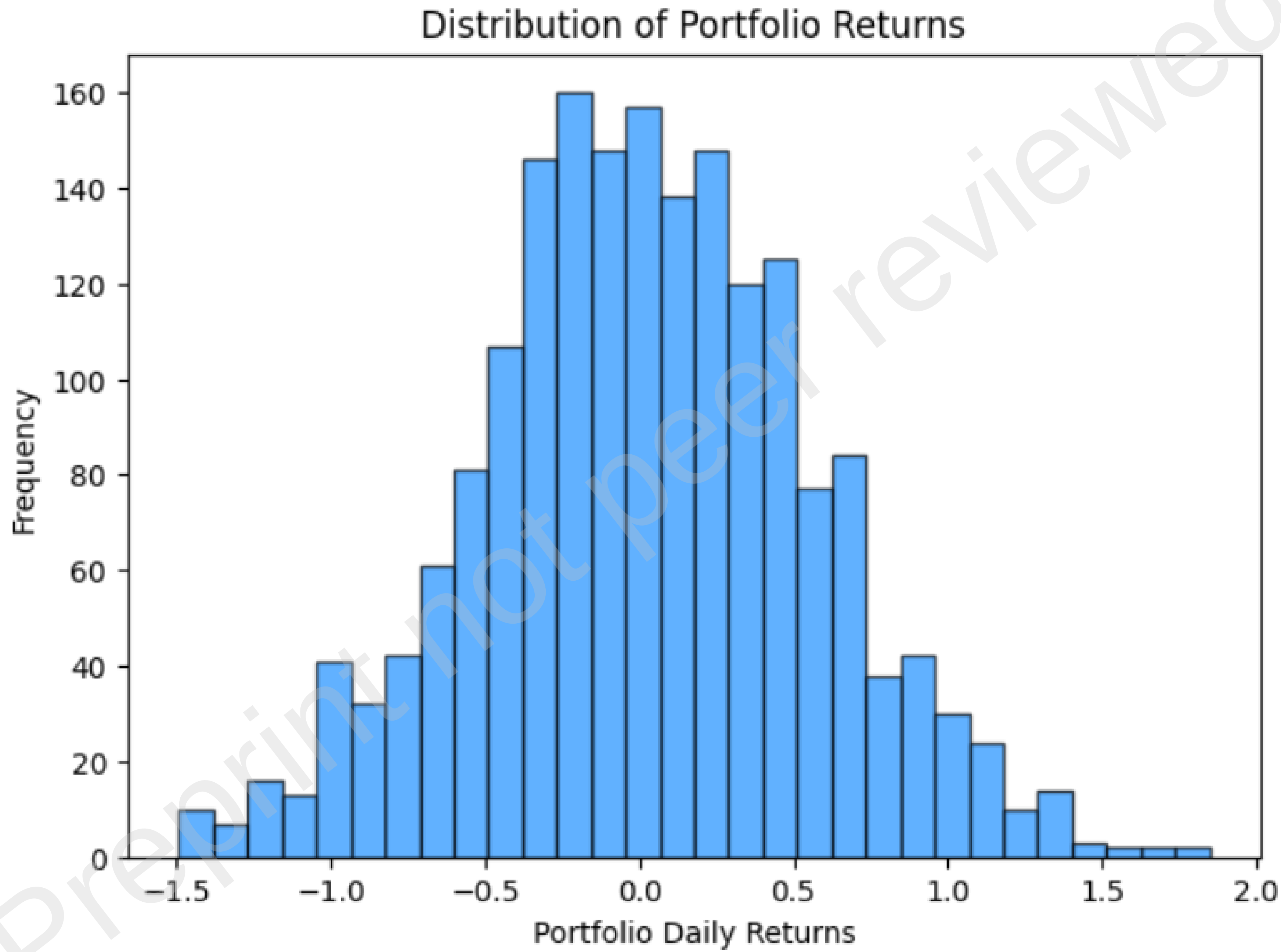
Zhou, P., Li, D., Wu, H., Cheng, F., 2011. The automatic model selection and variable kernel width for rbf neural networks. Neurocomputing 74, 3628–3637.

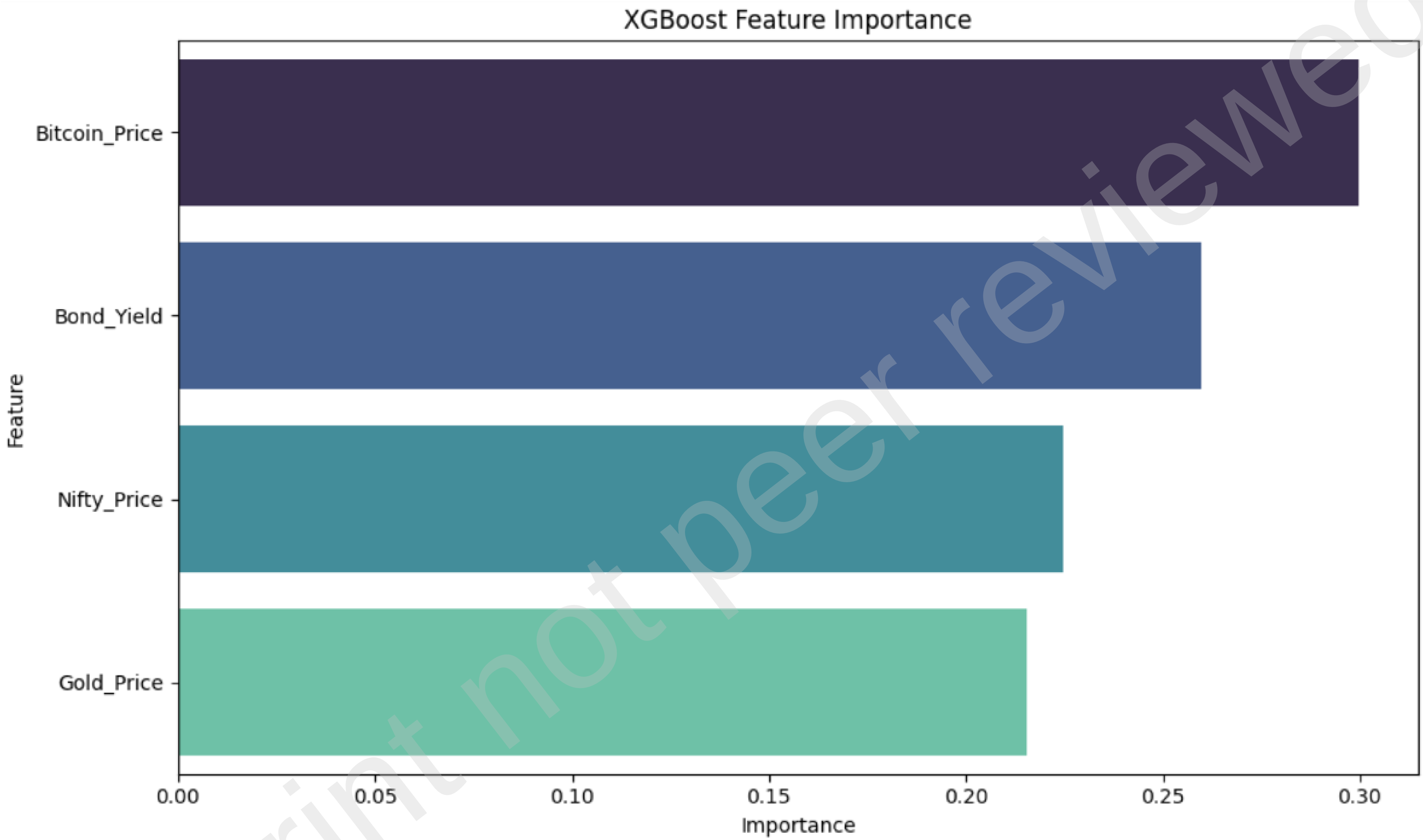

Comparison of Activation Functions
Identity
Sigmoid (Logistic)
ReLU
Tanh
Swish (β=1)
Proposed AB-Swish (α=1.5, β=3.7)
8
6
4
2
0
−2
−4
−6
f(x)
−6
−4
−2
0
2
4
6
x

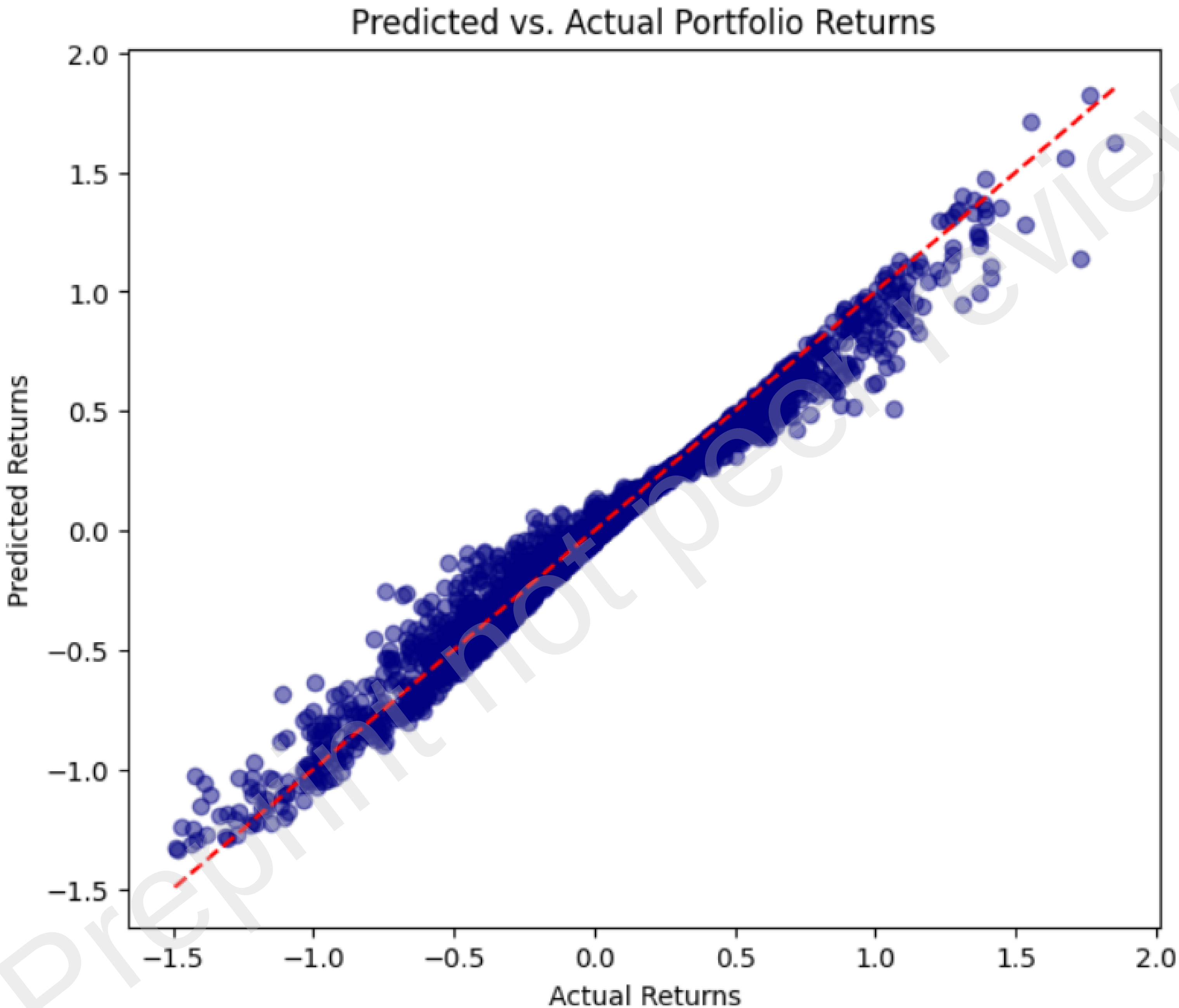

Preprocessing
(IQR + Robust Scaler)
Return Classification using XGBoost
x_1
x_2
x_3
x_N
Input Layer
ϕ_1
ϕ_2
ϕ_3
ϕ_{10}
(10 centers)
Hidden Layer (RBF Centers)
w_1
w_2
w_3
w_M
Summation
Σ
$\sum_{j=1}^{10} w_j \phi_j + b$
Activation
AB-Swish
y
Output
Weight Allocation & Return/Volatility % Calculation

```
XGBoost Classification Accuracy    : 0.9468
XGBoost Classification F1 Score    : 0.9479
```


Confusion Matrix
145
2
3
163
True Label
0
1
0
1
Predicted Label


Distribution of Portfolio Returns
Frequency
Portfolio Daily Returns
0
20
40
60
80
100
120
140
160
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0


XGBoost Feature Importance
Bitcoin_Price
Bond_Yield
Nifty_Price
Gold_Price
Feature
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Importance


Predicted vs. Actual Portfolio Returns
Predicted Returns
Actual Returns
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0