




Article

# Improving Forecasting Accuracy of Stock Market Indices Utilizing Attention-Based LSTM Networks with a Novel Asymmetric Loss Function

Shlok Sagar Rajpal <sup>1</sup>, Rajesh Mahadeva <sup>1</sup>, Amit Kumar Goyal <sup>2,\*</sup> and Varun Sarma <sup>3</sup>

<sup>1</sup> Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India; shlok.sagar.rajpall@gmail.com (S.S.R.); rajesh.mahadeva@manipal.edu (R.M.)

<sup>2</sup> Department of Electronics and Communication Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

<sup>3</sup> Department of Humanities and Management, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India; varun.sarda@manipal.edu

\* Correspondence: amit.goyal@manipal.edu

## Abstract

This study presents a novel approach to financial time series forecasting by introducing asymmetric loss functions. This is specifically designed to enhance directional accuracy across major stock indices (S&P 500, DJI, and NASDAQ Composite) over a 33-year time period. We integrate these loss functions into an attention-based Long Short-Term Memory (LSTM) framework. The proposed loss functions are evaluated against traditional metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and other recent research-based losses. Our approach consistently achieves superior test-time directional accuracy, with gains of 3.4–6.1 percentage points over MSE/MAE and 2.0–4.5 percentage points over prior asymmetric losses, which are either non-differentiable or require extensive hyperparameter tuning. Furthermore, proposed models also achieve an F1 score of up to 0.74, compared to 0.63–0.68 for existing methods, and maintain competitive MAE values within 0.01–0.03 of the baseline. The optimized asymmetric loss functions improve specificity to above 0.62 and ensure a better balance between precision and recall. These results underscore the potential of directionally aware loss design to enhance AI-driven financial forecasting systems.

**Keywords:** stock market forecasting; financial time series; asymmetric loss function; long-short term memory



Academic Editor: Xianrong (Shawn) Zheng

Received: 7 August 2025

Revised: 30 September 2025

Accepted: 8 October 2025

Published: 17 October 2025

**Citation:** Rajpal, S.S.; Mahadeva, R.; Goyal, A.K.; Sarma, V. Improving Forecasting Accuracy of Stock Market Indices Utilizing Attention-Based LSTM Networks with a Novel Asymmetric Loss Function. *AI* 2025, 6, 268. <https://doi.org/10.3390/ai6100268>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Financial time series are inherently complex and unique because they are volatile, nonlinear, and non-stationary; they possess distinctive features like fat-tailed distributions, volatility clusters, sudden regime changes, and irregular trends. Moreover, financial markets are driven by various factors like company-specific performance measures, investor sentiment, geopolitical events, and macroeconomic indicators. Early models such as ARIMA [1] and GARCH [2] were designed for time series forecasting, but their linear and stationary assumptions limited their ability to capture the dynamics of stock prices. Machine learning methods such as Random Forests and Support Vector Machines (SVMs) [3–5] offered better representation of nonlinearities but struggled with sequential dependencies

in time series. As a result, neither statistical methods nor machine learning models consistently outperformed simple moving averages. The deep learning techniques can extract complex features from raw sequential data, making them highly effective for financial forecasting. Among various DL architectures, Recurrent Neural Networks (RNNs) and particularly Long Short-Term Memory (LSTM) networks have gained widespread adoption in financial modeling.

A significant challenge in stock price prediction arises when models are trained with traditional symmetric loss functions such as MSE or MAE. These functions bias predictions toward the conditional mean of returns, leading to forecasts that under-represent sharp price movements and directional shifts [6]. As a result, the practical utility of such models for trading strategies, where accurate directional prediction is often more important than minimizing average error, is limited. The study of asymmetric loss functions in forecasting can be traced back to early contributions by Granger et al. [7] and Christoffersen et al. [8], who introduced piecewise loss functions to account for situations where over- and under-predictions have unequal costs; while these approaches highlighted the importance of asymmetry, subsequent research has struggled to design loss functions that are both continuous and differentiable, which are desirable properties for stable optimization. Michańków et al. [9] proposed the MADL loss function, which was asymmetric but not differentiable, and tested it on crypto and commodity forecasting. They further proposed the concept of a new differentiable variant, GMADL, but did not validate it [10]. Dessain [11] proposed three non-differentiable and one differentiable asymmetric loss functions, but validated only the first three.

A common limitation in the literature is that researchers often evaluate their loss functions on different datasets, horizons, and asset classes, which makes results difficult to compare directly. In contrast, we benchmark our proposed loss against AdjMSE, MADL and GMADL under the exact same data and model setup. We propose a loss function that is asymmetric and differentiable, along with three different scaling techniques that adjust the penalty in proportion to returns. This design reduces the influence of small, noisy daily fluctuations while emphasizing significant market movements. We hope this provides a clear and consistent reference point for future research in asymmetric loss functions for financial forecasting.

The motivation behind this work is to develop a loss function that is not restricted to a single task but can generalize across different financial forecasting applications. Financial time series models are used in diverse contexts such as directional prediction for trading, volatility estimation for derivatives pricing, and downside risk measurement in portfolio management. Each of these use cases demands sensitivity to sharp market movements while remaining robust to small, noisy fluctuations. Existing loss functions often excel in narrow setups but fail to transfer across tasks or datasets. By proposing a differentiable asymmetric loss with scalable penalties, we aim to provide a unified framework that maintains stability during optimization, captures directional accuracy, and adapts to varying financial objectives. This work takes a step toward building a generalizable loss function tailored for the complexities of financial time series.

The remainder of this paper is organized as follows. The related works section reviews prior literature, covering classical statistical methods, machine learning, and deep learning approaches for financial time series prediction, and various loss functions, including asymmetric, shape-aware (e.g., DILATE, TILDE-Q), and those based on trading metrics. The methodology section outlines the dataset, model architecture, and proposed loss functions. The results section presents empirical findings and analyzes performance across different metrics. Finally, the conclusion summarizes key insights and suggests directions for future work.

## 2. Literature Review

Financial forecasting has steadily shifted from traditional statistical models to machine learning and deep learning methods due to the statistical properties of stock price returns, which exhibit fat tails and heteroskedasticity. As forecasting models improve, recent research papers have focused on data processing and loss function design, in addition to model architecture.

Statistical and Traditional Machine Learning Approaches: Box and Jenkins [1] introduced the ARIMA model, which assumes stationarity and linear relationships among variables; while effective in short-term forecasting, ARIMA struggled to capture the non-linear and highly volatile nature of stock price movements [12]. Bollerslev's [2] GARCH model improved over ARIMA by modeling time-varying volatility, defining the variance of current error terms as a function of past values. This development made GARCH popular among models used to forecast market volatility, but it was not able to model stock prices or returns accurately. Due to the limitations of traditional statistical models, research work shifted towards machine learning methods. Kyoung-jae [5] applied Support Vector Machines (SVMs) to financial time series and found them more effective than conventional models, though they required heavy manual feature engineering. Huang et al. [4] extended this approach by combining SVMs with technical indicators; while this improved predictive accuracy, it also increased computational complexity, limiting scalability. Kumar et al. [13] tested hybrid models that integrated ARIMA with SVM, Artificial Neural Networks (ANNs), and Random Forests (RFs). Their results showed that ARIMA-SVM combinations produced stronger forecasts and better trading performance than other pairings. Patel et al. [3] compared several machine learning models using a sliding window framework to evaluate predictive accuracy. Random Forests emerged as the most reliable, though their lack of temporal sensitivity meant forecasts often lagged behind actual price movements.

ARIMA and GARCH were designed for time series forecasting, but were not able to model stock price forecasts. In contrast, machine learning methods were able to model nonlinear patterns, but were not able to handle sequential time series data. As a result, both statistical methods and machine learning models failed to outperform simple moving averages. Therefore, researchers began experimenting with neural networks, which handled nonlinear data better and could handle sequential data input.

**Deep Learning Methods:** Nelson et al. [14] applied Long Short-Term Memory (LSTM) neural networks to predict Brazilian stocks listed on the Bovespa using raw price data and a set of technical indicators. Their LSTM model outperformed traditional machine learning models, including Multi-Layer Perceptrons (MLPs), Random Forests, and SVMs, demonstrating LSTM's superior ability to capture temporal dependencies in financial time series. They achieved directional accuracy of 59%. A landmark study by Fischer and Krauss [15] used LSTM networks for predicting directional movements for the stocks constituting the S&P 500 from 1992 until 2015, achieving an accuracy of 51.4% in the US market. Based on this work, Ghosh et al. [16] generalized the work to intraday trading and proved that LSTM networks could produce daily returns of 0.64% as opposed to 0.54% with random forests, with directional accuracy of 60.1% on out-of-sample data. Chandola et al. [17] proposed a hybrid predictive model that combined Word2Vec embeddings and Long Short-Term Memory (LSTM) networks to predict financial markets, but it was limited by the availability of date-wise arranged textual data.

### 2.1. Asymmetric Loss Functions

Christoffersen and Diebold [8] developed a theoretical framework for forecasting under asymmetric loss functions, showing that when forecast errors have unequal costs depending

on their sign, the optimal forecast is generally biased relative to the conditional mean. The LinEx (linear-exponential) loss function penalizes forecast errors asymmetrically. Overpredictions or underpredictions incur different costs depending on a parameter that controls the direction of asymmetry. The Lin-Lin (linear-linear) loss function applies linear penalties to errors but with different weights for overpredictions and underpredictions, allowing flexible asymmetric cost sensitivity. Derivations of both LinEx and Lin-Lin functions show that optimal forecasts systematically deviate from the conditional mean to account for the differing costs of over- and underprediction. This highlights the importance of incorporating conditional heteroskedasticity and higher-order moments in forecasting models with asymmetric costs.

Building on this, Elliott et al. [18] proposed a general family of loss functions that nests the Lin-Lin, quadratic, and other common forms as special cases. This framework allowed for flexible testing of forecast rationality under asymmetric preferences using a Generalized Method of Moments (GMMs) approach. Their work showed that allowing for asymmetry can reverse conclusions drawn under symmetric loss, validating forecasts previously considered biased. Markov and Tan [19] conducted extensive empirical studies on analyst earnings forecasts, using over 260,000 individual predictions. They estimated the asymmetry parameter  $\alpha$  for each analyst and found strong evidence that analysts are more averse to over-predicting than under-predicting earnings. They also documented systematic variation in  $\alpha$  across firms, linking it to analyst employment outcomes. Their results emphasize that forecast evaluation in financial settings depends critically on the assumed loss structure. Patton and Timmermann [20] demonstrated that many standard properties of optimal forecasts, which hold under Mean Squared Error (MSE) loss, can fail under more general and realistic conditions. The authors show that under asymmetric loss and nonlinear DGPs, optimal forecasts can be biased, forecast errors can exhibit serial correlation even at the one-step horizon, and forecast error variance can decrease with forecast horizon, all violating the standard MSE-based optimality properties. They provide analytical results for the optimal forecast under the Linex loss function and a Markov regime-switching DGP, and discuss implications for GARCH(1,1) processes.

Hsu and Tai [21] proposed a composite loss function integrating magnitude and directional errors to improve profitability in commodity price prediction. A hyperparameter  $\gamma$  controls the influence of the directional term relative to the magnitude error. The approach was applied to futures data for gold, soybeans, and crude oil, using LSTM and Temporal Convolutional Network (TCN) architectures with MinMax-normalized inputs. The directional penalty was tested for various  $\gamma$  values (0.001, 0.01, 0.1, 1), with  $\gamma = 0.01$  found optimal. Smaller values made the directional term negligible, while larger values caused training instability due to abrupt loss changes when predicted and actual directions mismatched. The loss function improved annualized returns by 12–15% for gold and soybean and increased directional accuracy to 81% (from 73% under MSE). For crude oil, higher volatility limited the loss function's stabilizing effect, producing  $\pm 8\%$  swings. The study also showed that standard error measures (e.g.,  $R^2$ , MASE) correlated poorly with trading profitability, underscoring the need for loss functions aligned with trading objectives. The optimal  $\gamma$  must be determined empirically for each asset, and further adaptation may be required for different market conditions or asset classes.

Yin [22] proposed the Direction-Integrated Mean Square Error (DI-MSE) to address a core limitation in stock prediction: conventional losses such as MSE minimize value error but ignore directional accuracy, which is crucial for trading.

DI-MSE loss is defined as:

$$\mathcal{L}_{\text{DI-MSE}} = \underbrace{\frac{1}{n'} \sum_{D_i=1} W_i (\hat{y}_i - y_i)^2}_{\text{DLC}} + \underbrace{\frac{1}{N - n'} \sum_{D_i=0} 1}_{\text{DLW}}, \quad (1)$$

$$D_i = \begin{cases} 1, & \text{sign}(\Delta y_i) = \text{sign}(\Delta \hat{y}_i) \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

and  $W_i$  is a weight that reflects the proportion of upward or downward moves in the dataset.

Applied to MinMax-scaled OHLC data from 28 assets (20 stocks, 8 indices, 2015–2023) with LSTM and BiLSTM models, DI-MSE improved directional accuracy from 53% (MSE baseline) to 60% on average, with only a minor decrease in mean price accuracy (0.2–0.35%). However, the weighting scheme  $W_i$  depends on historical move distributions and may not adapt optimally to shifting market regimes.

Dessain [11] introduced a family of adjoint loss functions for return prediction, addressing the asymmetry of financial forecasting errors. Four variants were tested on daily returns of 105 NYSE and Euronext stocks (1996–2020), using a multilayer perceptron trained on technical indicators and OHLCV data. A threshold-based long-only trading rule is used, with transaction costs taken into account. Let  $y_{\text{true}}$  denote the actual return and  $y_{\text{pred}}$  the predicted return; then the loss functions are formulated below.

1. **AdjLoss1:**

$$\mathcal{L}_{\text{AdjLoss1}} = \begin{cases} \alpha \cdot (y_{\text{pred}} - y_{\text{true}})^2 & \text{if } y_{\text{true}} \cdot y_{\text{pred}} < 0 \\ \frac{1}{\alpha} \cdot (y_{\text{pred}} - y_{\text{true}})^2 & \text{if } y_{\text{true}} \cdot y_{\text{pred}} \geq 0 \end{cases}, \quad (3)$$

where  $\alpha > 1$ ; Dessain used  $\alpha = 2$  &  $\alpha = 1.5$ .

2. **AdjLoss2:**

$$\mathcal{L}_{\text{AdjLoss2}} = \begin{cases} \beta \cdot (y_{\text{pred}} - y_{\text{true}})^2 & \text{if } y_{\text{true}} \cdot y_{\text{pred}} < 0 \\ (y_{\text{pred}} - y_{\text{true}})^2 & \text{otherwise} \end{cases}, \quad (4)$$

where  $\beta > 1$ ; Dessain used  $\beta = 2$  &  $\beta = 1.5$ .

3. **AdjLoss3:**

$$\mathcal{L}_{\text{AdjLoss3}} = \begin{cases} (1 + \gamma) \cdot (y_{\text{pred}} - y_{\text{true}})^2 & \text{if } y_{\text{true}} \cdot y_{\text{pred}} < 0 \\ \gamma \cdot (y_{\text{pred}} - y_{\text{true}})^2 & \text{if } y_{\text{true}} \cdot y_{\text{pred}} > 0 \end{cases}, \quad (5)$$

where  $0 < \gamma < 1$ ; Dessain used  $\gamma = 0.1$ .

4. **AdjLoss4:** (fully differentiable)

$$\mathcal{L}_{\text{AdjMSE4}} = \frac{\delta \cdot (y_{\text{pred}} - y_{\text{true}})^2}{1 + \delta - \left( \frac{\delta - 0.5}{1 + e^{100 \cdot y_{\text{true}} \cdot y_{\text{pred}}}} \right)}, \quad (6)$$

where  $\delta > 1$  controls the steepness; Dessain used  $\delta = 0.4$ .

After testing, all loss functions outperformed MSE in risk-adjusted return (D ratio), consistency, and directional accuracy. AdjLoss2 ( $\beta = 2.5$ ) achieves the highest and most stable D ratio (mean 1.585, compared to 0.217 for MSE), and also outperforms MSE and LinEx in terms of D-Return and D-VaR. However, the penalty parameters will require periodic recalibration for optimal performance, and the loss functions were non differentiable.

Michańków et al. [9] proposed the Mean Absolute Directional Loss (MADL) for LSTM-based algorithmic trading. MADL penalizes misaligned forecasts in proportion to realized return magnitude, aligning optimization with trading profitability. Tests on Bitcoin and crude oil data showed that LSTMs trained with MADL achieved superior returns and risk-adjusted performance: an annualized compounded return of 109.94% and an information ratio of 1.062 (vs. 44.26% and 0.030 under MAE). MADL-based strategies

also reduced drawdowns. However, its reliance on sign and absolute value introduces non-differentiability, slowing convergence, and it risks over-penalizing in low-volatility markets if not calibrated.

The MADL loss is defined as

$$\mathcal{L}_{\text{MADL}} = \frac{1}{N} \sum_{i=1}^N (-\text{sign}(R_i \cdot \hat{R}_i) \cdot |R_i|), \quad (7)$$

where  $R_i$  is the real return,  $\hat{R}_i$  the predicted return, and  $N$  the number of forecasts.

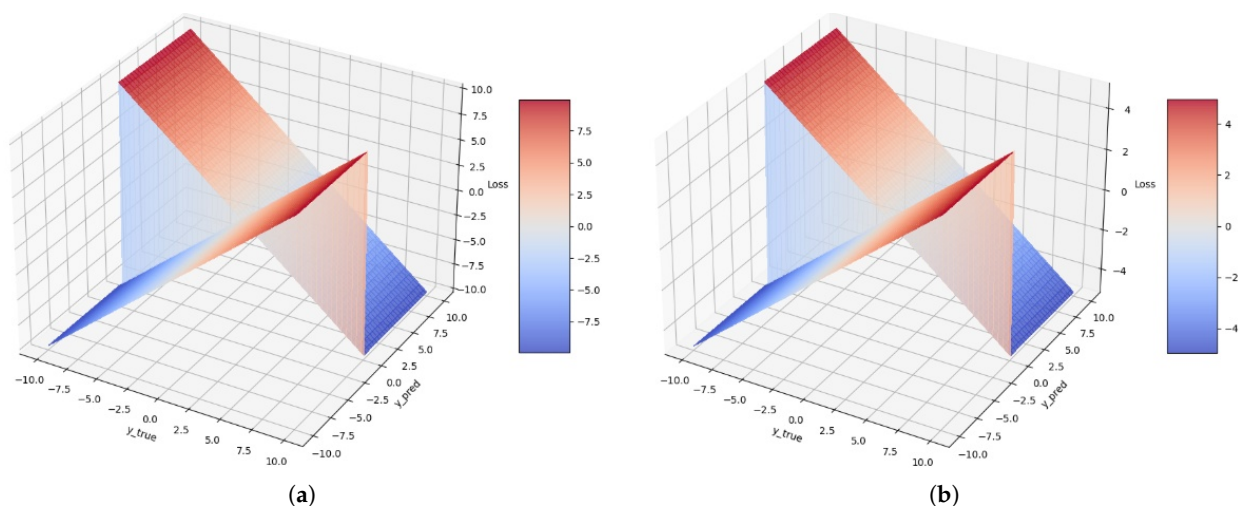
Building on their earlier work, Michańków et al. [10] proposed the Generalized Mean Absolute Directional Loss (GMADL), which is differentiable and has greater flexibility by using a smooth sigmoid-based directional penalty and tunable parameters, making it more suitable for gradient-based optimization.

The GMADL function is defined as

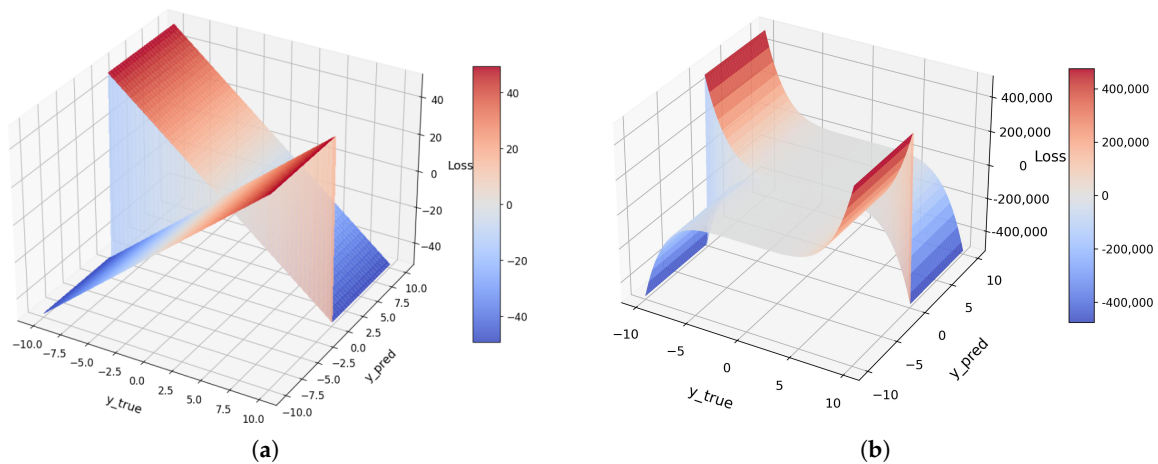
$$\text{GMADL} = \frac{1}{N} \sum_{i=1}^N \left( - \left( \frac{1}{1 + \exp(-aR_i\hat{R}_i)} - 0.5 \right) \cdot |R_i|^b \right), \quad (8)$$

where  $a$  controls the steepness of the sigmoid (directional penalty smoothness) and  $b$  adjusts the emphasis on large returns. This design ensures differentiability at all points and allows practitioners to fine-tune the loss function to specific trading objectives and market conditions. The GMADL loss function has two key hyperparameters,  $a$  and  $b$ , which control directional sensitivity and magnitude emphasis, respectively. The parameter  $a$  governs the steepness of the directional penalty through the sigmoid term  $\text{sigmoid}(a r \hat{r})$ . The parameter  $b$  modulates the magnitude emphasis by raising the absolute true value  $|r|$  to the power  $b$ .

Michańków et al. [9,10] only formulated the function and outlined its potential benefits, without conducting empirical tests. They suggest future research directions, including integrating transaction cost models and dynamic parameter adjustment based on market regime detection. Figures 1 and 2 represent how MADL and GMADL are different and how hyperparameters affect the loss magnitude and shape.



**Figure 1.** 3D visualizations of MADL and GMADL ( $a = 1000$ ,  $b = 1$ ) loss functions: (a) 3D surface plot of the MADL loss function; (b) 3D view of GMADL loss surface for  $a = 1000$ ,  $b = 1$ .



**Figure 2.** 3D visualizations and GMADL loss functions: (a) 3D view of GMADL loss surface for  $a = 100, b = 1$ ; (b) 3D view of GMADL loss surface for  $a = 1000, b = 5$ .

Liao et al. [23] proposed Tre-Loss, a loss framework for time-series forecasting that explicitly captures both global trend direction and local trend shape, addressing the shortcomings of standard losses such as MSE and MAE under distortions and non-stationarity. The global component uses Pearson correlation to align the overall trend direction, while the local component penalizes mismatches in pointwise trend using a tanh-weighted term. In experiments across multiple benchmark datasets, Tre-Loss outperformed MSE, DILATE [24], and TILDE-Q [25] on metrics including MAE, SMAPE, and shape-based metrics (DTW, TDI). The global trend term was found to be especially critical for accuracy, with optimal weighting determined via cross-validation. The main limitation of Tre-loss is increased computational cost relative to MSE, though it remains more efficient than DILATE.

## 2.2. Drawbacks Of Present Asymmetric Loss Functions

Asymmetric loss functions aim to align model training with economic objectives but face persistent challenges. Two common forms are directional asymmetry, which penalizes incorrect trend forecasts more than magnitude errors, and magnitude asymmetry, which assigns unequal penalties to over- versus under-predictions. Research indicates that the degree of asymmetry significantly influences model behavior; moderate asymmetry reduces extreme errors without introducing systematic bias. In contrast, strong asymmetry can skew predictions toward conservative estimates or induce mean reversion. These functions often introduce non-differentiable thresholds at decision boundaries, leading to noisy gradients and unstable training dynamics. Their effectiveness depends on careful calibration to avoid overfitting to specific market regimes. Studies emphasize that while these functions improve economic alignment, their success relies on balancing asymmetry intensity with robustness to distributional shifts and noisy data.

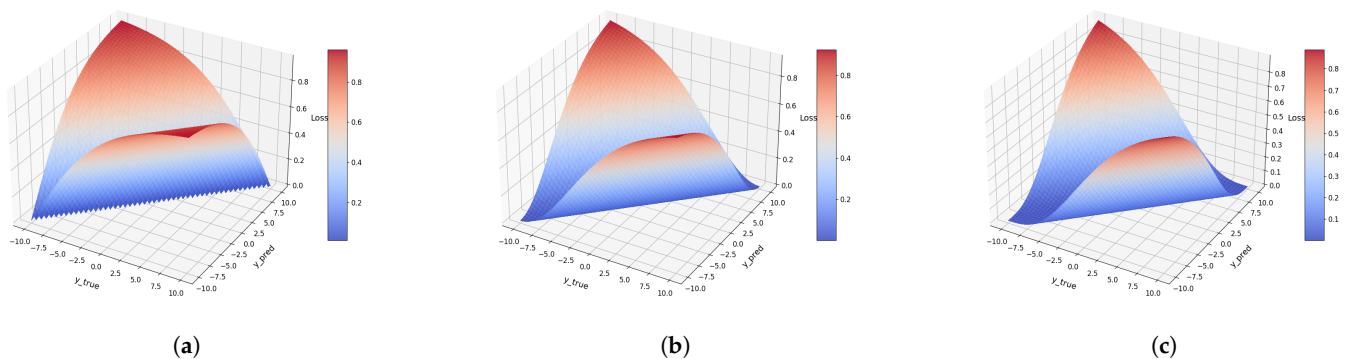
Our research builds upon these foundations by systematically comparing multiple loss functions for forecasting percentage changes in close prices using attention-augmented LSTM models. We introduce a novel set of configurable loss functions that are asymmetric and differentiable everywhere. These functions are designed to reflect directional priorities and error sensitivity in financial forecasting while maintaining smooth gradients to support stable optimization. This sets them apart from traditional approaches, often involving piecewise components or non-differentiable points, which can introduce noise during training. We conduct a structured experimental comparison across several loss functions

to assess how different formulations impact forecasting accuracy. To ensure fairness, we evaluate all loss functions using the same model architecture and the same datasets, enabling a controlled assessment of their effects.

### 3. Methodology

#### 3.1. Proposed Loss Function

We propose a loss function class that focuses on directional accuracy and magnitude-aware weighting. We scale the loss relative to the true observations to incentivize the model to focus on more significant movements instead of small noisy data. These loss functions are formulated by three main principles: input normalization, a bounded directional error component, and adaptive magnitude-based weighting. Input normalization is performed by dividing both the predicted value  $\hat{y}$  and the true value  $y$  by 10 to map data from the range of  $[-10, 10]$  to  $[-1, 1]$ . Unlike unbounded losses such as MSE, which are highly sensitive to outliers and can destabilize training, the bounded error formulation improves robustness while retaining sensitivity to directional signals. Figure 3 visualizes the base component of the loss function without the magnitude relative scaling. The minima for all the graphs lie along the  $y = \hat{y}$  line, where  $y$  and  $\hat{y}$  are the true and predicted return, respectively. The first graph (Figure 3a) is non-differentiable at the minima because of the modulus sign, but as  $\gamma$  increases (Figure 3b,c), the graphs become differentiable and sharper toward the minima.



**Figure 3.** 3D plots of the tanh loss function for different values of  $\gamma$ : (a)  $\gamma = 1$ ; (b)  $\gamma = 2$ ; (c)  $\gamma = 3$ .

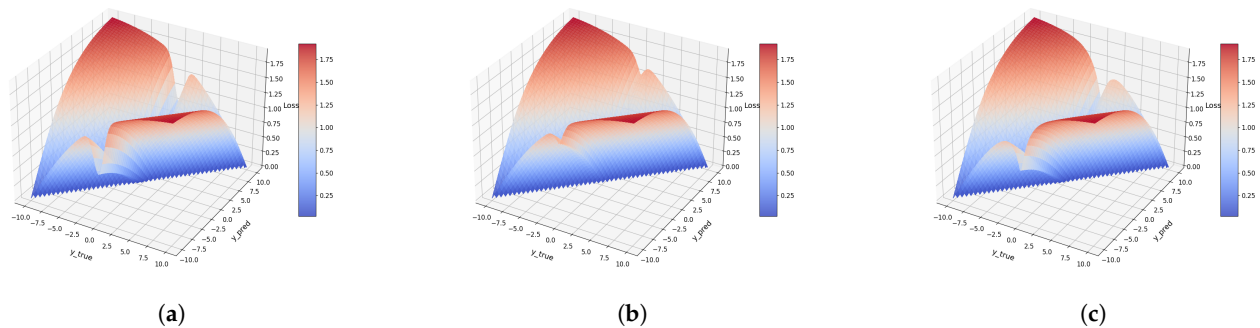
The proposed loss function is expressed as

$$\mathcal{L}(y, \hat{y}) = \left| \tanh\left(\frac{\hat{y} - y}{10}\right) \right|^\gamma \cdot (1 - e^{-\alpha W}). \quad (9)$$

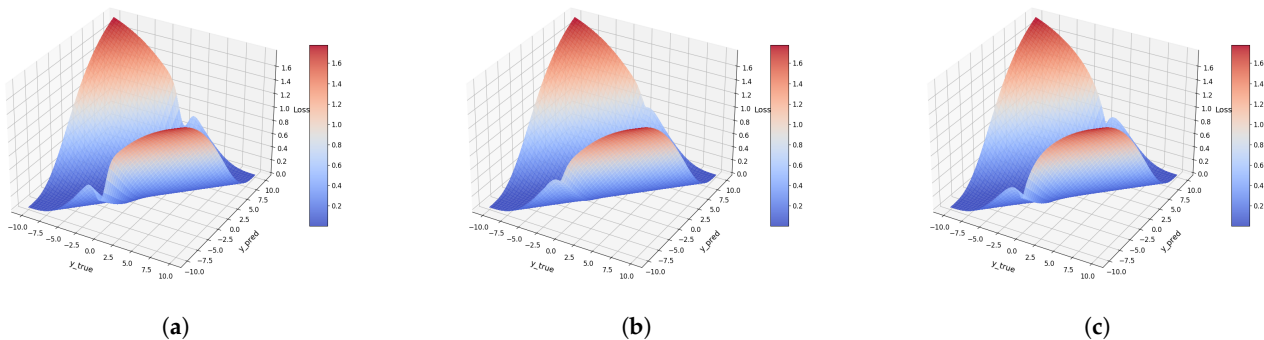
Here,  $\gamma$  controls the focus on directional accuracy,  $\alpha$  determines the strength of magnitude weighting, and  $W$  represents a scaling function. The base loss is tanh of prediction error, which prioritizes the direction of the error more than its absolute size; the exponent  $\gamma$  adjusts the sensitivity to errors. Magnitude-based weighting is applied with  $(1 - e^{-\alpha W})$ ; this component increases the loss for samples with larger true values, representing the idea that large price movements are more significant and should be prioritized during training.

Three variants of the proposed loss function are shown for different hyperparameter settings. Figure 4 presents the quadratic, exponential, and linear forms for  $(\alpha, \gamma) = (1, 1)$ , Figure 5 shows the same variants for  $(\alpha, \gamma) = (1, 3)$ , and Figure 6 illustrates them for  $(\alpha, \gamma) = (3, 1)$ . Increasing  $\alpha$  or  $\gamma$  sharpens the penalty contours, emphasizing larger deviations and magnitudes in the target variable. The visualizations of the proposed loss functions illustrate how varying the hyperparameters  $\alpha$  and  $\gamma$  shapes the loss landscape. Across all graphs, a valley appears along the line  $x = y$ , emphasizing correct directional predictions. A depression near  $x = 0$  reflects the magnitude-aware scaling, which reduces

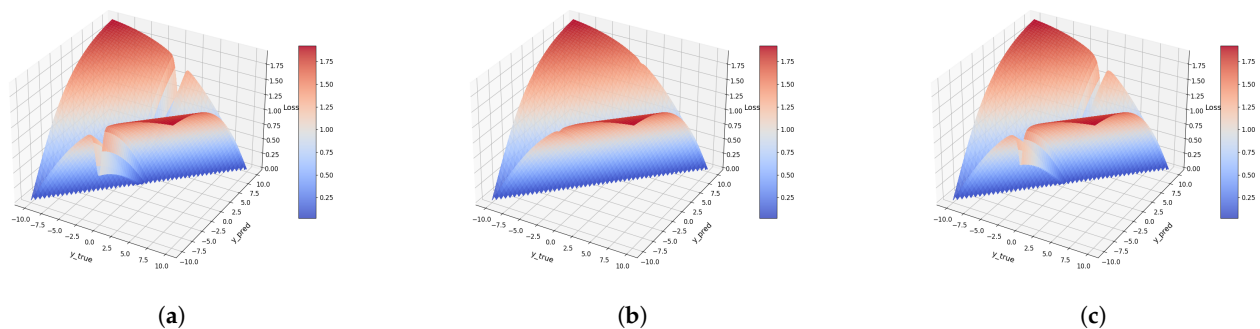
penalties for small true values. Increasing  $\gamma$  sharpens the graph towards the  $x = y$  line, as observed in Figure 5, and increasing  $\alpha$  decreases the depth of the valley at  $x = 0$ , as observed in Figure 6. After raising  $\gamma$  to a certain value the graph becomes too steep and makes the training process turbulent. After increasing  $\alpha$  to a certain value, the impact of the scaling function saturates as the magnitude of loss for smaller values begins to approximate the magnitude of loss from the base tanh function. The scaling function is built upon a fixed exponential form, with different variants (quadratic, exponential, and linear) applied on top. Directly applying these functions without the exponential base would overpower the tanh function, flattening the loss landscape and diminishing its sensitivity to directional accuracy. The quadratic variant increases the loss with the square of the true value, providing smooth emphasis on both moderate and large movements while remaining robust to small fluctuations. The exponential variant amplifies the loss sharply for larger true values, focusing learning on rare but extreme market events and creating a threshold-like effect. The linear variant scales the loss proportionally to the magnitude, ensuring a simple and interpretable contribution from all samples.



**Figure 4.** Comparison of quadratic, exponential, and linear variants at  $(\alpha, \gamma) = (1, 1)$ : (a) quadratic; (b) exponential; (c) linear.



**Figure 5.** Comparison of quadratic, exponential, and linear variants at  $(\alpha, \gamma) = (1, 3)$ : (a) quadratic; (b) exponential; (c) linear.



**Figure 6.** Comparison of quadratic, exponential, and linear variants at  $(\alpha, \gamma) = (3, 1)$ : (a) quadratic; (b) exponential; (c) linear.

**The quadratic weighting variant** is defined as

$$\mathcal{L}_{\text{sq}}(y, \hat{y}) = \left| \tanh\left(\frac{\hat{y} - y}{10}\right) \right|^\gamma \cdot \left(1 + \left(1 - e^{-\alpha y^2}\right)\right). \quad (10)$$

This version uses a quadratic term  $y^2$  in the weighting, which results in a smooth and gradual increase in emphasis as the magnitude of  $y$  grows. The quadratic form is particularly effective for situations where both moderate and large movements should be considered, providing a balanced approach that reduces sensitivity to noise in small fluctuations while still prioritizing larger moves.

**The exponential weighting variant** is given by

$$\mathcal{L}_{\text{exp}}(y, \hat{y}) = \left| \tanh\left(\frac{\hat{y} - y}{10}\right) \right|^\gamma \cdot \left(1 + \left(1 - e^{-\alpha e^{|y|}}\right)\right). \quad (11)$$

Here,  $e^{|y|}$  makes the weighting grow very rapidly for large  $|y|$ . This variant is highly sensitive to extreme price movements, making it suitable for applications where capturing rare but significant market events is critical.

**The linear weighting variant** is formulated as

$$\mathcal{L}_{\text{exp}}(y, \hat{y}) = \left| \tanh\left(\frac{\hat{y} - y}{10}\right) \right|^\gamma \cdot \left(1 + \left(1 - e^{-\alpha |y|}\right)\right). \quad (12)$$

This approach applies a direct, proportional increase in weighting as  $|y|$  increases. The linear form is simple and ensures that all samples contribute to the loss in proportion to their magnitude.

### 3.2. Data Collection and Preprocessing

Historical price data comprising Open, High, Low, Close, and Volume metrics were collected from Yahoo Finance for the S&P 500, Dow Jones Industrial Average (DJI), and NASDAQ Composite (IXIC) indices for 33 years, spanning 1 January 1990, to 30 December 2023. These three indices were selected due to their complementary characteristics: the S&P 500 provides a broad and diversified view of the U.S. equity market, the DJI represents blue-chip industrial stability, and the NASDAQ emphasizes technology-driven growth. Together, they represent the most liquid, widely tracked, and globally influential benchmarks for equity performance. To address scale disparities and non-stationarity in the data, we applied percentage change normalization [26]. For each time series  $X = x_1, x_2, \dots, x_T$ , daily returns  $P = p_2, p_3, \dots, p_T$  were computed as

$$p_t = \frac{x_t - x_{t-1}}{x_{t-1}} \times 100, \quad (13)$$

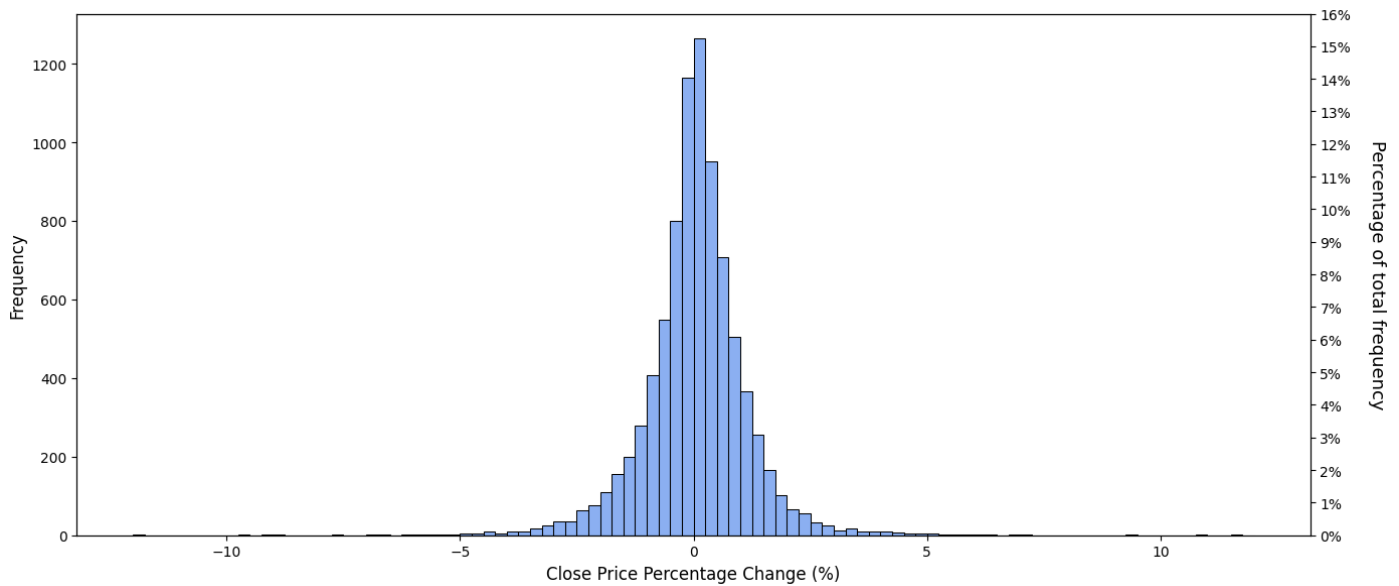
where  $x_t$  is the OHLCV value at time  $t$  and  $p_t$  is the normalized percentage change. This transformation ensures stationarity, reduces scale sensitivity across indices, and preserves directional information that is essential for forecasting tasks. Compared to normalization techniques such as Min–Max scaling and z-score standardization, percentage change normalization avoids look-ahead bias because each return is calculated using only current and past values. This makes it more suitable for time-series forecasting tasks where preserving causal structure is critical.

Table 1 displays a dataset sample after normalization. Figure 7 presents the histogram of daily close price percentage changes. Most daily returns are clustered tightly around zero, reflecting the relatively small day-to-day fluctuations typical in major stock indices. Outliers and extreme values suggest that large price movements occur more frequently

than expected under a proper normal distribution. This characteristic, commonly observed in financial time series, is referred to as “fat tails”.

**Table 1.** S&P500 Percentage return data.

Date (YYYY-MM-DD)	Close	High	Low	Open	Volume
1990-01-03	−0.2586	0.2502	1.6791	1.7799	18.6709
1990-01-04	−0.8613	−0.5075	−1.3971	−0.2586	−7.9707
1990-01-05	−0.9756	−0.8613	−0.4364	−0.8613	−10.435
1990-01-08	0.4514	0.4021	0.3505	0.9756	11.6193
⋮	⋮	⋮	⋮	⋮	⋮
2023-12-26	0.4232	0.2468	0.0957	0.1660	9.2803
2023-12-27	0.0143	0.0140	0.2196	0.3066	9.3297
2023-12-28	0.0307	0.1653	0.2533	0.2721	8.4003
2023-12-29	−0.2826	−0.1016	−0.6064	−0.0744	15.8289



**Figure 7.** Distribution of close price percentage changes for S&P500.

### 3.3. Model Architecture

The model used in this study is intended to work as a baseline for evaluating the performance of various loss functions; it uses LSTM layers for sequence modeling, enhanced with self-attention, residual connections, and PReLU activations.

Table 2 outlines the architecture of the custom LSTM model used in this study. The model processes input sequences of length 20, where each sequence consists of OHLCV (Open, High, Low, Close, Volume) features converted into percentage changes. Both LSTM layers use *tanh* for hidden states and *sigmoid* for gates, with kernels initialized by LeCun uniform, recurrent weights by orthogonal initialization, and biases set to zero. Regularization includes L2 penalties, dropout of 0.2, and recurrent dropout of 0.2 to mitigate overfitting. A self-attention mechanism emphasizes informative time steps, while residual connections preserve both local and global context. Layer normalization further stabilizes activations and accelerates convergence. Two dense layers with PReLU activations provide flexible non-linearity through learnable negative slopes, and the final linear output layer produces the predicted percentage change in closing price.

**Table 2.** Custom LSTM model architecture.

Layer/Shape	Activation/Regularization	Description
Input (20, 5)	—	Accepts preprocessed OHLCV data as percentage changes.
LSTM-1 (20, 5) → (20, 64)	Activation: Tanh; Recurrent: Sigmoid L2 regularization ( $1 \times 10^{-6}$ ) on kernel and recurrent weights Dropout: 0.2; Recurrent Dropout: 0.2 Initializers: LeCun uniform (kernel), orthogonal (recurrent), zeros (bias)	Extracts local temporal dependencies from the input sequence.
Self-Attention (20, 64) → (20, 64)	—	Computes self-attention over LSTM outputs to highlight relevant temporal features.
Residual Add (20, 64) → (20, 64)	—	Adds attention output back to LSTM activations to enhance feature representation.
LayerNormalization (20, 64)	—	Normalizes activations to stabilize and accelerate training.
LSTM-2 (20, 64) → (32)	Activation: Tanh; Recurrent: Sigmoid L2 regularization ( $1 \times 10^{-6}$ ) on kernel and recurrent weights Dropout: 0.2 Initializers: LeCun uniform (kernel), orthogonal (recurrent), zeros (bias)	Aggregates and summarizes sequential information into a condensed feature vector.
Dense-1 (32) → (16)	Linear; Activation: PReLU (learnable $\alpha$ )	Applies adaptive non-linearity to enhance feature representation.
Dense-2 (16) → (8)	Linear; Activation: PReLU (learnable $\alpha$ )	Further refines features and increases model expressiveness.
Output (8) → (1)	Linear	Outputs the predicted percentage change in closing price.

### 3.4. Software and Hardware Setup

Model development and experimentation were carried out using a personal laptop for code implementation and Kaggle’s GPU environment for training. The Kaggle setup provided an NVIDIA Tesla P100 GPU, sufficient for handling the sequence-based architectures explored in this study. All models were implemented in Python-3.11 using TensorFlow as the primary deep learning framework, with supporting libraries including NumPy and Pandas for data manipulation. No external experiment tracking or data management platforms were employed, ensuring a straightforward and reproducible workflow.

## 4. Results and Comparative Analysis

### 4.1. Evaluation Metrics

The model’s performance was evaluated using directional accuracy, which directly reflects the model’s effectiveness in capturing market sentiment. In addition to directional accuracy, we monitored the Mean Absolute Error (MAE) to evaluate the precision with which our model can estimate the magnitude, and the Precision, Recall, Specificity, and F1 score of the model to understand its bias and variance. Loss functions were tested with the S&P 500, DJI, and NASDAQ Composite indices. The model was trained for 100 epochs in each case, and the model state corresponding to the lowest validation loss was retained for final evaluation. This early stopping strategy helped mitigate overfitting and ensured that the best-performing model was preserved as measured on unseen data.

Tables 3–5 summarize the performance of traditional loss functions (MSE and MAE) on S&P500, DJI, and IXIC, respectively. Across all indices, both MSE and MAE show smooth and stable convergence during training.

**Table 3.** Performance metrics for traditional loss functions on S&P500.

Loss Function	Train Metrics						Test Metrics					
	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score
MSE	58.10	0.68	0.57	0.90	0.21	0.70	52.88	0.80	0.56	0.85	0.21	0.69
MAE	56.69	0.75	0.54	0.98	0.02	0.68	52.29	0.79	0.55	0.99	0.02	0.70

**Table 4.** Performance metrics for traditional loss functions on DJI.

Loss Function	Train Metrics						Test Metrics					
	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score
MSE	59.21	0.68	0.50	0.91	0.11	0.65	51.88	0.74	0.50	0.87	0.10	0.62
MAE	58.29	0.70	0.51	0.98	0.03	0.66	51.08	0.77	0.51	0.97	0.02	0.70

**Table 5.** Performance metrics for traditional loss functions on IXIC.

Loss Function	Train Metrics						Test Metrics					
	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score
MSE	57.54	0.66	0.53	0.94	0.18	0.68	52.24	0.79	0.50	0.92	0.08	0.64
MAE	55.06	0.72	0.50	0.97	0.04	0.66	52.17	0.78	0.51	0.95	0.07	0.69

For the S&P500 (Table 3), directional accuracy on the test set plateaus at around 52.88% for MSE and 52.29% for MAE, which is only slightly above random guessing and roughly aligned with the underlying class distribution, where 53.48% of returns are positive. Precision and recall are consistently high, particularly recall (0.85–0.99), while specificity is extremely low (0.21 for MSE and 0.02 for MAE), highlighting a systematic bias towards predicting positive movements. The resulting F1-scores (0.69–0.70) are largely driven by the high recall rather than balanced performance.

For DJI (Table 4), similar trends are observed. Test directional accuracy remains slightly lower than for the S&P500, with MSE achieving 51.88% and MAE 51.08%. Precision and recall patterns mirror those seen in S&P500, with recall near 0.87–0.97 and specificity very low (0.10 for MSE and 0.02 for MAE), indicating the model is again biased towards predicting positive returns. The MAE model shows a slightly higher F1-score on the test set (0.70) due to marginally improved recall, despite a lower directional accuracy.

For IXIC (Table 5), test directional accuracy is comparable, with MSE at 52.24% and MAE at 52.17%. Here, recall remains high (0.92–0.95) while specificity is particularly low (0.08–0.07), confirming the same bias towards positive movement predictions observed in S&P500 and DJI. Precision is moderate (0.50–0.51), and F1-scores (0.64–0.69) again reflect the dominance of high recall over balanced performance.

Across all indices, the consistent pattern is that traditional loss functions such as MSE and MAE favor positive predictions due to the class imbalance inherent in financial returns, while training metrics indicate smooth convergence and seemingly reasonable error reduction; test metrics reveal that these losses fail to directly incentivize directional correctness, leading to limited out-of-sample performance. Overall, directional accuracy barely exceeds the random baseline, specificity is low, and F1-scores are primarily driven by recall rather than balanced predictive performance.

MADL, GMADL and AdjMSE:

As illustrated in Tables 6–8, the MADL, GMADL, and AdjMSE loss functions exhibit distinct patterns in directional accuracy, magnitude prediction, and overall stability across the S&P500, DJI, and IXIC indices.

MADL demonstrates moderate directional accuracy on all three indices, with test set values of 52.47% (S&P500), 50.06% (DJI), and 51.18% (IXIC). Precision values remain consistent around 0.54–0.56, while recall is extremely high for S&P500 and IXIC, but lower for DJI, indicating a tendency to overpredict the dominant direction. Specificity is very low across most cases, highlighting poor prediction of the minority class, though DJI shows slightly better specificity. F1-scores remain relatively stable, driven largely by high recall. Minimal degradation from train to test metrics suggests that the flatness of the MADL loss surface near the axes produces negligible gradients for smaller or negative returns, causing the model to rely on prior directional bias rather than learning important patterns.

GMADL exhibits broadly similar behavior to MADL, but hyperparameter variations introduce greater variability. Directional accuracy generally remains near 52–53% on the test set, with precision, recall, and specificity patterns closely mirroring MADL. Extreme parameter configurations, such as GMADL on S&P500, cause recall to collapse while specificity rises sharply, reflecting overprediction of the minority class and a large drop in directional accuracy. MAE values are highly variable and occasionally large, especially on IXIC and DJI, indicating numerical instability and difficulties in optimization. Overall, GMADL's loss surface remains too flat for most normalized return inputs, limiting the learning signal despite the additional hyperparameters.

**Table 6.** Performance metrics for MADL, GMADL and AdjMSE on S&P500.

Loss Function	Train Metrics						Test Metrics					
	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score
MADL	53.35	0.78	0.53	0.99	0.0003	0.69	52.47	0.80	0.54	0.99	0.01	0.70
GMADL (a = 100, b = 1)	53.34	3.29	0.53	0.99	0.00	0.69	52.30	3.19	0.54	0.99	0.00	0.70
GMADL (a = 100, b = 2)	53.34	4.37	0.53	0.99	0.00	0.69	52.30	4.41	0.54	0.99	0.00	0.70
GMADL (a = 100, b = 5)	53.22	2.06	0.53	0.99	0.002	0.69	52.35	1.94	0.54	0.99	0.003	0.70
GMADL (a = 500, b = 1)	53.34	1.77	0.53	1.00	0.00	0.69	52.30	1.79	0.54	1.00	0.00	0.70
GMADL (a = 500, b = 2)	53.35	2.93	0.53	1.00	0.00	0.70	52.36	2.86	0.54	0.98	0.002	0.70
GMADL (a = 500, b = 5)	53.34	3.34	0.53	1.00	0.00	0.69	52.30	3.27	0.54	1.00	0.00	0.70
GMADL (a = 1000, b = 1)	51.82	1.51	0.54	0.02	0.97	0.05	50.93	1.51	0.52	0.05	0.95	0.09
GMADL (a = 1000, b = 2)	53.31	1.42	0.53	0.98	0.02	0.69	52.30	1.64	0.54	0.98	0.02	0.69
GMADL (a = 1000, b = 5)	53.36	1.46	0.53	0.99	0.01	0.69	52.36	1.46	0.54	0.99	0.001	0.70
AdjMSE1 ( $\alpha = 1.5$ )	53.27	0.76	0.54	0.89	0.12	0.67	52.06	0.79	0.55	0.91	0.12	0.68
AdjMSE1 ( $\alpha = 2.0$ )	53.55	0.75	0.54	0.98	0.02	0.69	52.94	0.79	0.55	0.98	0.04	0.70
AdjMSE2 ( $\beta = 2.25$ )	53.71	0.75	0.54	0.98	0.03	0.69	53.03	0.78	0.55	0.98	0.03	0.70
AdjMSE2 ( $\beta = 2.5$ )	54.35	0.74	0.54	0.97	0.06	0.69	53.12	0.79	0.56	0.98	0.05	0.69
AdjMSE3 ( $\gamma = 0.1$ )	53.72	0.76	0.54	0.98	0.03	0.69	52.94	0.79	0.55	0.97	0.04	0.70

**Table 7.** Performance metrics for MADL, GMADL and AdjMSE on DJI.

Loss Function	Train Metrics						Test Metrics					
	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score
MADL	50.38	0.73	0.53	0.21	0.79	0.31	50.06	0.78	0.55	0.26	0.74	0.36
GMADL (a = 100, b = 1)	52.91	2.16	0.53	1.00	0.00	0.69	51.10	2.15	0.55	1.00	0.00	0.71
GMADL (a = 100, b = 2)	52.91	1.81	0.53	1.00	0.00	0.69	51.04	1.84	0.55	1.00	0.00	0.71
GMADL (a = 100, b = 5)	52.91	1.99	0.53	1.00	0.00	0.69	51.04	2.01	0.55	1.00	0.00	0.71
GMADL (a = 500, b = 1)	52.91	1.84	0.53	1.00	0.00	0.69	51.04	1.86	0.55	1.00	0.00	0.71
GMADL (a = 500, b = 2)	52.91	2.04	0.53	1.00	0.00	0.69	51.04	2.06	0.55	1.00	0.00	0.71
GMADL (a = 500, b = 5)	52.91	5.35	0.53	1.00	0.00	0.69	51.04	5.58	0.55	1.00	0.00	0.71
GMADL (a = 1000, b = 1)	52.91	1.91	0.53	1.00	0.00	0.69	51.10	1.93	0.55	1.00	0.01	0.71
GMADL (a = 1000, b = 2)	53.95	2.56	0.00	0.00	1.00	0.00	51.90	2.65	0.00	0.00	1.00	0.00
GMADL (a = 1000, b = 5)	52.91	6.16	0.53	1.00	0.00	0.69	50.10	6.13	0.55	1.00	0.00	0.71
AdjMSE1 ( $\alpha = 1.5$ )	53.86	0.70	0.54	0.92	0.11	0.68	51.09	0.79	0.56	0.95	0.08	0.71
AdjMSE1 ( $\alpha = 2.0$ )	52.52	0.73	0.53	0.89	0.11	0.67	50.72	0.78	0.56	0.92	0.12	0.69
AdjMSE2 ( $\beta = 2.25$ )	53.14	0.73	0.53	1.00	0.01	0.69	51.53	0.78	0.55	1.00	0.02	0.71
AdjMSE2 ( $\beta = 2.5$ )	53.45	0.73	0.53	0.98	0.03	0.69	51.34	0.78	0.56	0.98	0.05	0.71
AdjMSE3 ( $\gamma = 0.1$ )	52.91	0.73	0.53	1.00	0.00	0.69	51.16	0.78	0.55	1.00	0.00	0.71

**Table 8.** Performance metrics for MADL, GMADL and AdjMSE on IXIC.

Loss Function	Train Metrics						Test Metrics					
	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score
MADL	52.38	1.00	0.54	0.82	0.16	0.65	51.18	1.01	0.56	0.79	0.24	0.66
GMADL (a = 100, b = 1)	54.82	2.19	0.55	1.00	0.00	0.71	53.65	2.09	0.56	1.00	0.01	0.71
GMADL (a = 100, b = 2)	54.85	1.71	0.55	1.00	0.00	0.71	52.59	1.62	0.56	1.00	0.00	0.71
GMADL (a = 100, b = 5)	54.84	1.31	0.55	1.00	0.00	0.71	53.53	1.30	0.56	1.00	0.00	0.71
GMADL (a = 500, b = 1)	54.84	4.33	0.55	1.00	0.00	0.71	53.53	4.35	0.56	1.00	0.00	0.71
GMADL (a = 500, b = 2)	54.94	1.70	0.55	0.99	0.02	0.71	53.76	1.61	0.56	0.99	0.01	0.71
GMADL (a = 500, b = 5)	51.09	6.30	0.00	0.00	1.00	0.00	50.41	6.32	0.00	0.00	1.00	0.00
GMADL (a = 1000, b = 1)	54.85	2.17	0.55	0.95	0.06	0.70	52.58	1.98	0.57	0.87	0.18	0.69
GMADL (a = 1000, b = 2)	54.84	5.64	0.55	1.00	0.00	0.71	52.53	5.59	0.56	1.00	0.00	0.71
GMADL (a = 1000, b = 5)	54.92	3.47	0.55	1.00	0.00	0.71	53.82	3.24	0.56	1.00	0.01	0.71
AdjMSE1 ( $\alpha = 1.5$ )	55.49	0.98	0.55	0.98	0.03	0.71	53.06	1.01	0.56	0.99	0.03	0.71
AdjMSE1 ( $\alpha = 2.0$ )	55.19	1.00	0.55	0.97	0.05	0.70	53.88	1.01	0.56	0.98	0.03	0.71
AdjMSE2 ( $\beta = 2.25$ )	55.32	1.00	0.55	0.96	0.05	0.70	54.11	1.01	0.57	0.95	0.10	0.71
AdjMSE2 ( $\beta = 2.5$ )	55.84	0.98	0.55	0.98	0.04	0.71	54.94	1.01	0.56	0.99	0.03	0.71
AdjMSE3 ( $\gamma = 0.1$ )	54.25	1.00	0.55	0.95	0.04	0.70	53.76	1.01	0.56	0.97	0.05	0.71

AdjMSE provides more robust and consistent performance across all indices. Directional accuracy improves relative to MADL and GMADL, particularly with the AdjMSE2 variant, achieving test set values of 53.12% (S&P500), 51.34% (DJI), and 54.94% (IXIC). Precision and recall remain high, often exceeding 0.55, while specificity improves substantially compared to MADL and GMADL, indicating better minority class prediction. MAE values are lower and more stable, suggesting smoother optimization and reliable convergence during training, while non-differentiable boundaries in AdjMSE can induce abrupt changes when predictions are near zero, leading to noisy gradients, AdjMSE2 consistently achieves the highest directional accuracy and balanced F1-scores across all indices, corroborating its effectiveness as reported in the original research.

Comparing across indices, S&P500 and IXIC show stronger bias toward positive returns under MADL and GMADL, while DJI appears more balanced, revealing that return distributions influence loss function behavior. GMADL shows extreme sensitivity to hyperparameter selection on DJI and IXIC, sometimes collapsing recall or precision entirely, whereas AdjMSE remains robust across varying return distributions.

#### 4.2. Proposed Functions

The comparative performance of proposed loss function variants across multiple settings of  $\alpha$  and  $\gamma$  demonstrates consistent improvements in both directional accuracy and error metrics. With  $\alpha = 1$ , the loss functions have limited sensitivity to the magnitude, leading to moderate directional accuracy but very low specificity, indicating a higher rate of false signals. Increasing  $\alpha$  to 2 provides a more balanced treatment of magnitude, improving test performance across all loss functions. At  $\alpha = 3$ , models tend to overfit the training data, offering the best training directional accuracy but slightly reduced test performance and higher MAE.  $\gamma$  adjusts the penalty for directional errors. When  $\gamma = 1$ , the loss is more tolerant of misdirection, resulting in high recall but low specificity. A moderate value of  $\gamma = 2$  offers a balanced trade-off, with strong improvements in both specificity and F1-score. However, with  $\gamma = 3$ , the tanh function starts to dominate the magnitude-based scaling, resulting in over-penalization of minor directional errors, which increases MAE without substantial gains in accuracy metrics.

For all three indices: S&P500, DJI, and IXIC, the performance of the proposed loss function variants demonstrates consistent trends across hyperparameters ( $\alpha$  and  $\gamma$ ), scaling variants, and train-test splits. Table 9 reports the detailed results for S&P500, Table 10 for DJI, and Table 11 for IXIC. In the tanh configuration, increasing  $\gamma$  generally improves test directional accuracy slightly while reducing training accuracy, suggesting that higher focusing mitigates overfitting. Across indices, the effect is most pronounced on S&P500, where test directional accuracy peaks at 55.04% for  $\gamma = 2$ , whereas training accuracy declines from 59.59% ( $\gamma = 1$ ) to 55.89% ( $\gamma = 3$ ). DJI and IXIC show narrower ranges, with test accuracy generally between 52.66 and 52.91% and training accuracy between 52.75–57.78%, indicating more conservative sensitivity to  $\gamma$ .

**Table 9.** Comparative performance of proposed loss function variants on S&P500.

$\alpha$ & $\gamma$	Train Metrics						Test Metrics					
	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score
<b>Tanh</b>												
$\gamma = 1$	59.59	0.73	0.58	0.90	0.25	0.70	54.89	0.79	0.56	0.79	0.26	0.66
$\gamma = 2$	57.22	0.70	0.56	0.90	0.19	0.69	55.04	0.81	0.56	0.84	0.22	0.67
$\gamma = 3$	55.89	0.72	0.56	0.82	0.26	0.66	54.01	0.80	0.55	0.77	0.26	0.65
<b>Tanh scaled with quadratic function</b>												
$\alpha = 1, \gamma = 1$	55.43	0.75	0.53	0.98	0.02	0.69	54.71	0.78	0.54	0.98	0.02	0.70
$\alpha = 1, \gamma = 2$	59.85	0.66	0.58	0.83	0.32	0.68	54.88	0.81	0.56	0.78	0.27	0.65
$\alpha = 1, \gamma = 3$	55.29	0.75	0.53	0.99	0.01	0.69	54.12	0.79	0.54	0.99	0.01	0.70
$\alpha = 2, \gamma = 1$	55.54	0.75	0.53	0.88	0.13	0.67	54.41	0.78	0.55	0.83	0.19	0.66
$\alpha = 2, \gamma = 2$	57.25	0.75	0.53	0.99	0.01	0.69	55.24	0.78	0.54	0.97	0.03	0.69
$\alpha = 2, \gamma = 3$	55.21	0.77	0.53	0.99	0.01	0.69	54.06	0.86	0.54	0.98	0.01	0.70
$\alpha = 3, \gamma = 1$	60.24	0.71	0.57	0.93	0.22	0.71	54.82	0.83	0.55	0.84	0.19	0.67
$\alpha = 3, \gamma = 2$	57.64	0.71	0.56	0.87	0.24	0.68	54.41	0.79	0.55	0.81	0.22	0.66
$\alpha = 3, \gamma = 3$	55.54	0.75	0.53	0.97	0.02	0.69	54.41	0.78	0.54	0.97	0.03	0.69
<b>Tanh scaled with exponential function</b>												
$\alpha = 1, \gamma = 1$	55.10	0.75	0.53	0.97	0.04	0.69	54.82	0.78	0.54	0.96	0.04	0.69
$\alpha = 1, \gamma = 2$	56.21	0.72	0.56	0.79	0.30	0.65	55.19	0.79	0.56	0.80	0.25	0.66
$\alpha = 1, \gamma = 3$	54.86	0.74	0.54	0.93	0.10	0.68	54.18	0.79	0.55	0.86	0.16	0.67
$\alpha = 2, \gamma = 1$	54.32	0.75	0.53	0.99	0.01	0.69	54.41	0.78	0.54	0.99	0.01	0.70
$\alpha = 2, \gamma = 2$	58.91	0.65	0.57	0.89	0.24	0.69	55.29	0.80	0.56	0.82	0.23	0.66
$\alpha = 2, \gamma = 3$	56.43	0.73	0.55	0.86	0.20	0.67	55.06	0.79	0.56	0.81	0.24	0.66
$\alpha = 3, \gamma = 1$	57.57	0.73	0.56	0.89	0.20	0.69	54.76	0.78	0.55	0.80	0.24	0.65
$\alpha = 3, \gamma = 2$	56.92	0.73	0.54	0.95	0.08	0.69	55.27	0.79	0.55	0.93	0.09	0.69
$\alpha = 3, \gamma = 3$	55.45	0.74	0.54	0.89	0.14	0.67	54.65	0.79	0.55	0.86	0.16	0.67
<b>Tanh scaled with linear function</b>												
$\alpha = 1, \gamma = 1$	57.56	0.75	0.54	0.99	0.01	0.69	54.65	0.79	0.55	0.99	0.02	0.70
$\alpha = 1, \gamma = 2$	55.98	0.76	0.55	0.84	0.21	0.66	54.36	0.79	0.54	0.99	0.01	0.70
$\alpha = 1, \gamma = 3$	54.39	0.76	0.54	0.88	0.15	0.67	53.30	0.79	0.54	0.99	0.02	0.70
$\alpha = 2, \gamma = 1$	56.58	0.74	0.56	0.91	0.18	0.69	54.71	0.79	0.55	0.85	0.19	0.67
$\alpha = 2, \gamma = 2$	55.44	0.72	0.55	0.95	0.10	0.70	55.06	0.79	0.55	0.91	0.12	0.69
$\alpha = 2, \gamma = 3$	55.37	0.79	0.53	0.99	0.01	0.69	54.13	0.79	0.55	0.91	0.11	0.68
$\alpha = 3, \gamma = 1$	56.45	0.76	0.55	0.92	0.13	0.69	54.83	0.79	0.55	0.99	0.02	0.70
$\alpha = 3, \gamma = 2$	62.85	0.60	0.60	0.88	0.34	0.72	54.89	0.81	0.56	0.76	0.30	0.65
$\alpha = 3, \gamma = 3$	56.74	0.69	0.56	0.87	0.22	0.68	54.36	0.79	0.55	0.81	0.23	0.66

**Table 10.** Comparative performance of proposed loss function variants on DJI.

$\alpha$ & $\gamma$	Train Metrics						Test Metrics					
	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score
<b>Tanh</b>												
$\gamma = 1$	53.56	0.72	0.53	0.97	0.04	0.68	52.66	0.78	0.55	0.94	0.08	0.70
$\gamma = 2$	53.72	0.72	0.53	0.97	0.04	0.69	52.72	0.78	0.55	0.96	0.05	0.70
$\gamma = 3$	52.75	0.72	0.52	0.96	0.04	0.68	52.72	0.78	0.55	0.97	0.04	0.70
<b>Tanh scaled with quadratic function</b>												
$\alpha = 1, \gamma = 1$	52.78	0.73	0.53	0.97	0.04	0.68	51.85	0.78	0.56	0.95	0.08	0.70
$\alpha = 1, \gamma = 2$	53.59	0.72	0.53	0.97	0.05	0.69	51.91	0.78	0.56	0.95	0.08	0.70
$\alpha = 1, \gamma = 3$	52.92	0.72	0.53	0.99	0.01	0.69	51.10	0.78	0.55	0.99	0.02	0.71
$\alpha = 2, \gamma = 1$	53.89	0.72	0.54	0.98	0.04	0.69	51.78	0.78	0.56	0.97	0.06	0.71
$\alpha = 2, \gamma = 2$	54.09	0.71	0.54	0.94	0.09	0.68	51.47	0.78	0.56	0.93	0.09	0.70
$\alpha = 2, \gamma = 3$	53.33	0.72	0.53	0.99	0.02	0.69	51.29	0.78	0.55	0.98	0.03	0.71
$\alpha = 3, \gamma = 1$	53.44	0.72	0.53	0.98	0.03	0.69	51.78	0.78	0.56	0.97	0.05	0.71
$\alpha = 3, \gamma = 2$	54.57	0.70	0.54	0.88	0.17	0.67	52.03	0.79	0.57	0.87	0.18	0.69
$\alpha = 3, \gamma = 3$	53.42	0.73	0.53	0.98	0.03	0.69	51.53	0.78	0.55	0.97	0.05	0.71
<b>Tanh scaled with exponential function</b>												
$\alpha = 1, \gamma = 1$	54.99	0.71	0.54	0.95	0.10	0.69	52.85	0.78	0.56	0.91	0.13	0.69
$\alpha = 1, \gamma = 2$	57.78	0.65	0.57	0.87	0.25	0.69	52.91	0.79	0.57	0.82	0.24	0.67
$\alpha = 1, \gamma = 3$	54.84	0.68	0.54	0.92	0.13	0.68	52.66	0.78	0.56	0.92	0.11	0.70
$\alpha = 2, \gamma = 1$	54.57	0.71	0.54	0.91	0.14	0.68	52.16	0.78	0.56	0.92	0.12	0.70
$\alpha = 2, \gamma = 2$	53.61	0.72	0.53	0.94	0.08	0.68	52.91	0.78	0.56	0.94	0.09	0.70
$\alpha = 2, \gamma = 3$	55.01	0.70	0.55	0.88	0.17	0.68	52.28	0.80	0.56	0.92	0.13	0.70
$\alpha = 3, \gamma = 1$	53.61	0.72	0.53	0.98	0.04	0.69	51.28	0.78	0.56	0.97	0.06	0.71
$\alpha = 3, \gamma = 2$	55.94	0.68	0.55	0.94	0.13	0.69	54.59	0.78	0.56	0.92	0.13	0.70
$\alpha = 3, \gamma = 3$	52.86	0.73	0.54	0.82	0.20	0.65	51.79	0.78	0.57	0.76	0.28	0.65
<b>Tanh scaled with linear function</b>												
$\alpha = 1, \gamma = 1$	55.60	0.69	0.55	0.93	0.13	0.69	53.34	0.78	0.56	0.94	0.11	0.70
$\alpha = 1, \gamma = 2$	54.96	0.69	0.54	0.92	0.13	0.68	53.91	0.80	0.56	0.92	0.12	0.69
$\alpha = 1, \gamma = 3$	52.80	0.75	0.53	1.00	0.00	0.69	52.10	0.80	0.55	0.99	0.01	0.71
$\alpha = 2, \gamma = 1$	53.03	0.73	0.53	0.97	0.04	0.69	52.91	0.78	0.56	0.97	0.05	0.71
$\alpha = 2, \gamma = 2$	54.34	0.73	0.54	0.88	0.17	0.67	53.03	0.82	0.56	0.88	0.17	0.69
$\alpha = 2, \gamma = 3$	53.44	0.74	0.54	0.83	0.21	0.65	52.16	0.81	0.56	0.84	0.20	0.67
$\alpha = 3, \gamma = 1$	57.60	0.67	0.57	0.85	0.26	0.68	52.22	0.81	0.57	0.81	0.26	0.67
$\alpha = 3, \gamma = 2$	54.85	0.69	0.54	0.90	0.15	0.68	52.66	0.81	0.56	0.89	0.15	0.69
$\alpha = 3, \gamma = 3$	53.33	0.73	0.53	0.95	0.07	0.68	51.47	0.79	0.55	0.96	0.05	0.70

**Table 11.** Comparative performance of proposed loss function variants on IXIC.

$\alpha$ & $\gamma$	Train Metrics						Test Metrics					
	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score	Dir Acc (%)	MAE	Precision	Recall	Specificity	F1-score
<b>Tanh</b>												
$\gamma = 1$	53.56	0.72	0.53	0.97	0.04	0.68	52.66	0.78	0.55	0.94	0.08	0.70
$\gamma = 2$	53.72	0.72	0.53	0.97	0.04	0.69	52.72	0.78	0.55	0.96	0.05	0.70
$\gamma = 3$	52.75	0.72	0.52	0.96	0.04	0.68	51.72	0.78	0.55	0.97	0.04	0.70
<b>Tanh scaled with quadratic function</b>												
$\alpha = 1, \gamma = 1$	55.14	0.99	0.55	0.98	0.03	0.71	54.23	1.00	0.56	0.97	0.05	0.71
$\alpha = 1, \gamma = 2$	54.95	1.01	0.55	0.99	0.01	0.71	53.71	1.03	0.56	1.00	0.00	0.71
$\alpha = 1, \gamma = 3$	55.25	0.99	0.56	0.92	0.11	0.69	53.48	1.00	0.56	0.89	0.11	0.68
$\alpha = 2, \gamma = 1$	55.10	0.99	0.55	0.99	0.01	0.71	53.59	1.01	0.56	1.00	0.00	0.71
$\alpha = 2, \gamma = 2$	55.11	0.99	0.55	1.00	0.01	0.71	53.88	1.01	0.56	0.99	0.02	0.71
$\alpha = 2, \gamma = 3$	55.51	0.99	0.55	0.98	0.04	0.71	53.30	1.01	0.56	0.97	0.04	0.71
$\alpha = 3, \gamma = 1$	55.49	0.98	0.55	0.97	0.05	0.70	53.65	1.01	0.56	0.98	0.03	0.71
$\alpha = 3, \gamma = 2$	54.92	1.00	0.55	0.99	0.02	0.71	52.00	1.01	0.56	0.99	0.03	0.71
$\alpha = 3, \gamma = 3$	55.20	0.99	0.55	0.99	0.02	0.71	53.53	1.00	0.56	0.99	0.02	0.71
<b>Tanh scaled with exponential function</b>												
$\alpha = 1, \gamma = 1$	54.89	0.99	0.55	1.00	0.01	0.71	52.76	1.00	0.56	1.00	0.01	0.71
$\alpha = 1, \gamma = 2$	57.53	0.95	0.57	0.91	0.16	0.70	53.17	1.02	0.57	0.89	0.16	0.69
$\alpha = 1, \gamma = 3$	54.78	1.00	0.55	0.99	0.01	0.71	52.94	1.01	0.56	0.98	0.03	0.71
$\alpha = 2, \gamma = 1$	55.03	0.99	0.55	0.99	0.01	0.71	53.76	1.01	0.56	0.99	0.02	0.71
$\alpha = 2, \gamma = 2$	55.14	0.99	0.55	0.99	0.02	0.71	53.71	1.00	0.56	0.99	0.01	0.71
$\alpha = 2, \gamma = 3$	60.22	0.86	0.59	0.93	0.21	0.72	55.88	1.02	0.57	0.89	0.14	0.69
$\alpha = 3, \gamma = 1$	54.89	0.99	0.55	1.00	0.00	0.71	52.53	1.01	0.56	1.00	0.00	0.71
$\alpha = 3, \gamma = 2$	55.26	0.99	0.55	0.98	0.03	0.71	53.82	1.00	0.56	0.97	0.05	0.71
$\alpha = 3, \gamma = 3$	54.97	0.99	0.55	1.00	0.01	0.71	52.59	1.00	0.56	1.00	0.01	0.71
<b>Tanh scaled with linear function</b>												
$\alpha = 1, \gamma = 1$	54.91	0.99	0.55	1.00	0.01	0.71	53.71	1.00	0.56	1.00	0.01	0.71
$\alpha = 1, \gamma = 2$	54.78	0.99	0.55	0.94	0.07	0.69	52.41	1.00	0.56	0.88	0.14	0.69
$\alpha = 1, \gamma = 3$	54.72	0.99	0.55	0.98	0.03	0.70	52.71	1.00	0.55	0.93	0.07	0.70
$\alpha = 2, \gamma = 1$	57.54	0.96	0.57	0.93	0.14	0.71	55.82	1.04	0.56	0.91	0.12	0.70
$\alpha = 2, \gamma = 2$	55.32	0.99	0.56	0.89	0.15	0.69	54.77	1.01	0.56	0.85	0.17	0.68
$\alpha = 2, \gamma = 3$	55.20	0.96	0.57	0.79	0.27	0.66	54.83	1.04	0.57	0.79	0.25	0.66
$\alpha = 3, \gamma = 1$	55.00	0.99	0.55	0.99	0.02	0.71	53.82	1.01	0.56	0.99	0.02	0.71
$\alpha = 3, \gamma = 2$	54.78	0.99	0.55	1.00	0.00	0.71	52.65	1.01	0.56	1.00	0.00	0.71
$\alpha = 3, \gamma = 3$	57.50	0.92	0.57	0.90	0.18	0.70	54.35	1.03	0.57	0.84	0.19	0.68

Quadratic scaling increases training directional accuracy across all indices, with peaks at 62.85% (S&P500,  $\alpha = 3$ ,  $\gamma = 2$ ), 54.57% (DJI,  $\alpha = 3$ ,  $\gamma = 2$ ), and 60.22% (IXIC,  $\alpha = 2$ ,  $\gamma = 3$ ). However, test directional accuracy shows more modest improvement (S&P500: 54.89%, DJI: 52.03%, IXIC: 55.88%), indicating overfitting when scaling is aggressive. Exponential scaling balances train-test metrics better: S&P500 reaches 55.29% test accuracy ( $\alpha = 2$ ,  $\gamma = 2$ ) with 58.91% training accuracy, DJI achieves 54.59% test accuracy ( $\alpha = 3$ ,  $\gamma = 2$ ) with 55.94% training accuracy, and IXIC achieves 55.88% test accuracy ( $\alpha = 2$ ,  $\gamma = 3$ ) with 60.22% training accuracy. Linear scaling consistently provides the most conservative learning behavior, with both train and test metrics remaining stable across hyperparameters, e.g., S&P500 test accuracy 54.89% ( $\alpha = 3$ ,  $\gamma = 2$ ), DJI 52.66% ( $\alpha = 3$ ,  $\gamma = 2$ ), IXIC 54.35% ( $\alpha = 3$ ,  $\gamma = 3$ ).

Moderate hyperparameter values ( $\alpha = 2$ ,  $\gamma = 2$ ) consistently produce the highest test directional accuracy across all scaled variants and indices, suggesting that neither too weak nor too strong scaling/focusing is optimal. Low  $\alpha$  values underutilize the scaling effect, while high  $\alpha$  often leads to overfitting in quadratic scaling, especially on S&P500 and IXIC. Increasing  $\gamma$  improves recall slightly across indices but sometimes reduces specificity, indicating a trade-off between detecting positive movements and minimizing false positives. Across all indices, MAE remains remarkably stable between train and test sets, generally ranging from 0.65 to 0.79 for S&P500, 0.65 to 0.82 for DJI, and 0.86 to 1.04 for IXIC, indicating that all variants reliably estimate magnitude regardless of directional performance. Precision is tightly bound (0.53–0.60), with minimal train-test discrepancy, confirming reliability in predicting positive movements. Scaled variants with moderate  $\alpha$  and  $\gamma$  improve specificity modestly, enhancing correct detection of negative movements.

S&P500 demonstrates the largest gains from quadratic scaling due to higher variability, whereas DJI shows more muted sensitivity to scaling and hyperparameters, reflecting a narrower range of returns. IXIC exhibits higher MAE variance due to its volatile return distribution, making exponential scaling more advantageous for balancing recall and specificity. Across all indices, exponential scaling generally provides the best trade-off between overfitting and underfitting, quadratic scaling maximizes training accuracy but risks overfitting, and linear scaling yields conservative yet stable improvements.

Taken together, these results show that our proposed loss functions provide both higher directional accuracy and improved specificity compared to existing approaches. In practical terms, this means models trained with these functions are better at capturing true market direction while reducing false signals, a combination that directly supports decision-making in stock market forecasting and trading. The ability to more reliably anticipate upside and downside movements while limiting bias towards one class makes these loss functions particularly valuable for portfolio allocation, risk management, and the design of systematic trading strategies.

## 5. Conclusions and Future Improvements

This study highlights the critical role of loss functions in financial forecasting. By focusing on directional accuracy and error magnitude, the proposed loss functions consistently outperformed standard functions such as MSE and MAE and addressed limitations found in loss functions proposed by prior research papers that were either non-differentiable or overly dependent on hyperparameter tuning. The results demonstrate that well-structured, differentiable, and asymmetric loss functions can provide more reliable forecasts, achieving higher directional accuracy, balanced error sensitivity, and improved robustness compared to traditional approaches.

Future research can be grouped into three themes. First, methodological improvements such as jointly optimizing model weights and loss hyperparameters, or enabling

adaptive loss surfaces based on market regimes. Second, architectural exploration, particularly transformer-based models that can better capture long-range dependencies in financial time series. Third, feature enrichment through the integration of implied volatility, macroeconomic indicators, sentiment measures, and technical indicators like RSI, MACD, and Bollinger Bands. Beyond these, practical applications include portfolio optimization, high-frequency trading, and reinforcement learning frameworks that incorporate tailored losses to adapt dynamically to transaction costs and evolving risk constraints.

Overall, the findings underline the practical impact of loss function design: by improving directional accuracy and error sensitivity, these functions can translate directly into more robust trading systems and risk-aware forecasting tools with immediate utility in financial decision-making.

**Author Contributions:** Conceptualization, S.S.R., A.K.G., R.M., and V.S.; methodology, S.S.R., R.M., and V.S.; software, S.S.R., R.M., and V.S.; validation, S.S.R., and V.S.; formal analysis, S.S.R., R.M., and V.S.; investigation, A.K.G., R.M., and V.S.; resources, A.K.G., R.M., and V.S.; writing—original draft preparation, S.S.R., A.K.G., R.M., and V.S.; writing—review and editing, A.K.G., R.M., and V.S.; supervision, A.K.G., R.M., and V.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**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.

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

## References

1. Box, G.E.P.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*; Holden-Day: San Francisco, CA, USA, 1976.
2. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *J. Econom.* **1986**, *31*, 307–327. [\[CrossRef\]](#)
3. Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. Predicting stock market index using fusion of machine learning techniques. *Expert Syst. Appl.* **2015**, *42*, 2162–2172. [\[CrossRef\]](#)
4. Huang, W.; Nakamori, Y.; Wang, S.Y. Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.* **2005**, *32*, 2513–2522. [\[CrossRef\]](#)
5. Kyoung-jae, K. Financial time series forecasting using support vector machines. *Neurocomputing* **2003**, *55*, 307–319. [\[CrossRef\]](#)
6. Barua, M.; Kumar, T.; Raj, K.; Roy, A.M. Comparative Analysis of Deep Learning Models for Stock Price Prediction in the Indian Market. *FinTech* **2024**, *3*, 551–568. [\[CrossRef\]](#)
7. Granger, C.W. Outline of forecast theory using generalized cost functions. *Span. Econ. Rev.* **1999**, *1*, 161. [\[CrossRef\]](#)
8. Christoffersen, P.F.; Diebold, F.X. Optimal prediction under asymmetric loss. *Econom. Theory* **1997**, *13*, 808–817. [\[CrossRef\]](#)
9. Michańkó, J.; Sakowski, P.; Ślepaczuk, R. Mean Absolute Directional Loss as a New Loss Function for Machine Learning Problems in Algorithmic Investment Strategies. *arXiv* **2023**, arXiv:2309.10546. [\[CrossRef\]](#)
10. Michańkó, J.; Sakowski, P.; Ślepaczuk, R. Generalized Mean Absolute Directional Loss as a Solution to Overfitting and High Transaction Costs in Machine Learning Models Used in High-Frequency Algorithmic Investment Strategies. *arXiv* **2024**, arXiv:2412.18405. [\[CrossRef\]](#)
11. Dessain, J. Improving the Prediction of Asset Returns With Machine Learning by Using a Custom Loss Function. *Adv. Artif. Intell. Mach. Learn.* **2023**, *3*, 1640–1653. [\[CrossRef\]](#)
12. Kontopoulou, V.I.; Panagopoulos, A.D.; Kakkos, I.; Matsopoulos, G.K. A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks. *Future Internet* **2023**, *15*, 255. [\[CrossRef\]](#)
13. Kumar, M.; Thenmozhi, M. Forecasting stock index returns using ARIMA-SVM, ARIMA-ANN, and ARIMA-random forest hybrid models. *Int. J. Bank. Account. Financ.* **2014**, *5*, 284. [\[CrossRef\]](#)
14. Nelson, D.M.; Pereira, A.C.; Oliveira, R.A.D. Stock market's price movement prediction with LSTM neural networks. In Proceedings of the International Joint Conference on Neural Networks, Anchorage, AK, USA, 14–19 May 2017. [\[CrossRef\]](#)

15. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [[CrossRef](#)]
16. Ghosh, P.; Neufeld, A.; Sahoo, J.K. Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. *Financ. Res. Lett.* **2022**, *46*, 102280. [[CrossRef](#)]
17. Chandola, D.; Mehta, A.; Singh, S.; Tikkiwal, V.A.; Agrawal, H. Forecasting Directional Movement of Stock Prices using Deep Learning. *Ann. Data Sci.* **2023**, *10*, 1361–1378. [[CrossRef](#)] [[PubMed](#)]
18. Elliott, G.; Komunjer, I.; Timmermann, A. Estimation and Testing of Forecast Rationality under Flexible Loss. *Rev. Econ. Stud.* **2005**, *72*, 1107–1125. [[CrossRef](#)]
19. Markov, S.; Tan, M. Loss Function Asymmetry and Forecast Optimality: Evidence from Individual Analysts' Forecasts. *Account. Rev.* **2006**. Available online: [https://ink.library.smu.edu.sg/soa\\_research/165](https://ink.library.smu.edu.sg/soa_research/165) (accessed on 1 August 2025).
20. Patton, A.J.; Timmermann, A. Properties of optimal forecasts under asymmetric loss and nonlinearity. *J. Econom.* **2007**, *140*, 884–918. [[CrossRef](#)]
21. Hsu, C.; Tai, L. Exploring the Impact of Magnitude- and Direction-based Loss Function on the Profitability using Predicted Prices from Deep Learning. *Int. J. Eng. Manag. Res.* **2020**, *10*, 1–20. Available online: <https://ssrn.com/abstract=3560007> (accessed on 1 August 2025). [[CrossRef](#)]
22. Yin, H. Enhancing Directional Accuracy in Stock Closing Price Value Prediction Using a Direction-Integrated MSE Loss Function. In Proceedings of the 1st International Conference on Data Analysis and Machine Learning (DAML), Kuala Lumpur, Malaysia, 2023; pp. 119–126. Available online: <https://www.scitepress.org/Link.aspx?doi=10.5220/0012810200003885> (accessed on 1 August 2025). [[CrossRef](#)]
23. Liao, H.; Hu, Y.; Yuan, L. Time Series Forecasting with Trend Loss Function. 2024. Available online: <https://ssrn.com/abstract=4960118> (accessed on 1 August 2025).
24. Guen, V.L.; Thome, N. Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models. *arXiv* **2019**, arXiv:1909.09020. [[CrossRef](#)]
25. Lee, H.; Lee, C.; Lim, H.; Ko, S. TILDE-Q: A Transformation Invariant Loss Function for Time-Series Forecasting. *arXiv* **2024**, arXiv:2210.15050.
26. Pei, Z.; Yan, J.; Yan, J.; Yang, B.; Li, Z.; Zhang, L.; Liu, X.; Zhang, Y. A Stock Price Prediction Approach Based on Time Series Decomposition and Multi-Scale CNN using OHLCT Images. *arXiv* **2024**, arXiv:2410.19291. [[CrossRef](#)]

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