

Article

TS-GRU: A Stock Gated Recurrent Unit Model Driven via Neuro-Inspired Computation

Yuanfang Zhang ^{1,*} and Heinz D. Fill ²¹ School of Accounting and Finance, Xi'an Peihua University, Xi'an 710125, China² School of Computer Science, Cornell University, Ithaca, NY 14850, USA; hf18@cornell.edu

* Correspondence: 15191583495@163.com

Abstract: Existing risk measurement methods often fail to fully consider the impact of climatic conditions on stock market risk, making it difficult to capture dynamic patterns and long-term dependencies. To address these issues, we propose the TS-GRU method: this approach utilizes a temporal convolutional network (TCN) to extract underlying features from historical data, capturing key characteristics of time series data. Subsequently, a gated recurrent unit (GRU) model is employed to capture dynamic patterns and long-term dependencies within the stock market. Finally, the TS-GRU model is optimized using the Sparrow algorithm based on collective behavior, iteratively evaluating and refining model parameters to obtain improved solutions. Experimental results demonstrate the effectiveness of the TS-GRU method in providing accurate risk assessment and forecasting. This comprehensive approach takes into account carbon finance, climate change, and environmental factors, offering valuable insights to investors to help them to understand and manage investment risks in the ever-changing stock market.

Keywords: climate change; neuro-inspired computation; TCN; GRU; stock market; investment risk measurement; SSA

**Citation:** Zhang, Y.; Fill, H.D.

TS-GRU: A Stock Gated Recurrent Unit Model Driven via Neuro-Inspired Computation.

Electronics **2024**, *13*, 4659. <https://doi.org/10.3390/electronics13234659>

Academic Editor: Maciej Ławryńczuk

Received: 8 July 2024

Revised: 8 August 2024

Accepted: 12 August 2024

Published: 26 November 2024



Copyright: © 2024 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

Researching carbon financial market predictions is crucial because it plays a vital role in addressing global climate change and promoting the transition to a low-carbon economy [1]. Effective market forecasting helps governments and businesses to develop scientifically sound emission reduction strategies and provides reliable decision-making foundations for investors. The complexity and uncertainty of the carbon financial market increase the demand for advanced forecasting methods and tools [2]. Innovative prediction models can enhance market efficiency, promote the diversification of carbon financial products, and support the healthy development of the market. In summary, carbon financial market predictions are essential for achieving global climate goals and ensuring sustainable economic and societal development.

The first generation of artificial intelligence methods relied on symbolic AI and knowledge representation. The simulation-based computational approach proposed by Yang et al. addressed complex problems through mathematical models and simulations, laying the foundation for artificial intelligence [3]. Rule-based methods, as introduced by Reddy et al. [4], employed predefined rules for reasoning and decision making, effectively handling many specific issues and finding widespread application in early expert systems. The logistic regression method proposed by Ali et al. [5] performed well in binary classification problems, offering essential tools for statistics and machine learning. Despite their robust theoretical foundations and interpretability, these methods faced several significant challenges. Firstly, symbolic AI and rule-based methods struggled with scalability and adaptability, as creating and maintaining an extensive rule set for complex real-world applications was labor-intensive and prone to errors. Secondly, these methods often lacked

the ability to learn from new data, making them rigid and less effective logistic regression environments. Finally, while dynamic regression environments are powerful for linear and binary classification problems, they fall short in handling non-linear relationships and multi-class classifications. These limitations highlighted the need for more flexible and adaptive approaches in AI research.

To overcome the limitations of first-generation algorithms in handling complex problems, people have developed data-driven and machine-learning-based algorithms. These newer methods leverage large datasets and advanced learning models to achieve more efficient processing and prediction. They exhibit strong adaptability, excel at handling nonlinear problems, and perform well across various application domains. Methods based on decision trees [6], such as digital AI decision trees and disaster victim detection decision trees, have excelled in economic forecasting and disaster response, showing robust model interpretability and wide applicability. Approaches based on random forests [7], like the hybrid malware classification method and COVID-19 high-risk patient prediction, have demonstrated the algorithm's strong capabilities in enhancing detection accuracy and handling complex data. The multilayer perceptron (MLP) [8] algorithm has also shown remarkable performance in various fields, like fault diagnosis and emotion recognition. MLP has been applied to power transformer fault diagnosis and speech emotion recognition, highlighting the adaptability and efficiency of MLP across multiple domains. However, these methods come with drawbacks, including strong data dependencies, long training times, and high computational resource consumption when dealing with very large datasets.

To overcome the limitations of second-generation algorithms in handling complex data, deep-learning-based algorithms have been applied to carbon financial market predictions. These methods use multilayer neural networks and multimodal data to enhance feature extraction and adaptability. Convolutional neural networks (CNNs) [8] excel in image recognition and feature extraction, with applications in brain tumor classification, transfer learning on natural image datasets, and vegetation remote sensing. Long short-term memory networks (LSTMs) [9] perform well in processing time series data, including video forgery detection, cardiovascular disease prediction, and traffic flow prediction. Transformers [10] have achieved high accuracy in natural language processing and sequence labeling tasks. Attention mechanisms [11] improve model interpretability and performance, seen in visual explanations in image classification, diabetic retinopathy detection, and multimodal data processing. Despite their advantages, these methods require large amounts of annotated data, have long training times, and demand high computational resources.

Given the above context, this study aims to address the following key questions: How can time convolutional networks (TCNs) and gated recurrent unit (GRU) models be effectively utilized to capture dynamic patterns and long-term dependencies in the stock market? Can the Sparrow algorithm significantly enhance the accuracy of risk assessment and prediction in optimizing the TS-GRU model parameters? Considering carbon financing, climate change, and environmental factors, can the TS-GRU approach provide a more comprehensive and accurate investment risk assessment? By answering these research questions, we hope to develop an innovative method to offer investors more precise risk assessment tools, aiding them in making more informed decisions in a constantly changing market environment.

- TS-GRU combines TCNs and GRUs to extract features from historical data and capture dynamic patterns. It is optimized using the Sparrow algorithm, resulting in improved performance and stability.
- The TS-GRU method is applicable to multiple domains, such as climate and finance. It exhibits high efficiency and generality, and is capable of handling complex and large-scale data.
- Experiments conducted on real stock market data demonstrate that TS-GRU outperforms traditional methods in terms of prediction accuracy, stability, and computational efficiency. It provides a more reliable risk assessment.

2. Related Work

2.1. Time Series Forecasting

Time series forecasting is an important branch of data science that finds wide applications in domains such as economics, finance, environment, and engineering [12]. Traditional time series forecasting methods include moving average (MA), autoregressive (AR), and autoregressive moving average (ARMA) models, which utilize historical data for predictions and are suitable for linear and stationary time series [13]. With the increase in computational power and the availability of larger datasets, machine-learning- and deep-learning-based time series forecasting methods have gained popularity. Methods such as support vector regression (SVR) and random forests have excelled in nonlinear time series forecasting. Deep learning methods like LSTM and GRUs have significantly improved prediction performance by capturing long-term dependencies in time series. Moreover, time convolutional networks (TCNs) and Transformer models have been introduced to the field of time series forecasting, further enhancing the ability to handle complex time series data [14]. These methods have shown remarkable performance not only in financial market forecasting but also in energy consumption prediction, traffic flow forecasting, and climate change analysis, demonstrating their powerful application potential.

2.2. Carbon Financial Market Predictions

Carbon financial market predictions have received significant attention in recent years as an important means to address climate change. Early research heavily relied on traditional statistical methods such as time series analysis and regression models to forecast carbon prices and market fluctuations. These traditional methods, while foundational, often struggled to perform well in the face of complex market dynamics and nonlinear features, leading to less accurate and stable predictions. With the advancement of data-driven approaches, machine learning and deep learning techniques have gradually been introduced into carbon market predictions [15]. Methods like support vector machines (SVMs) and random forests have achieved some success in carbon price forecasting by leveraging their ability to handle nonlinear relationships and interactions within the data. SVMs, for instance, have been used to find the optimal hyperplane that separates different classes of data points, thus aiding in making precise predictions. Random forests, on the other hand, utilize ensemble learning to improve prediction accuracy by combining the outputs of multiple decision trees, which mitigates overfitting and enhances generalization to new data [16]. More recently, deep learning methods such as convolutional neural networks (CNNs) and long short-term memory (LSTM) networks have been employed to capture the complex dynamic features of carbon markets. CNNs, although traditionally used in image processing, have been adapted to recognize patterns in time series data by applying convolutional filters, thereby capturing local dependencies and patterns. LSTM networks, designed to remember long-term dependencies in sequences, have significantly improved the accuracy and stability of predictions by effectively managing the vanishing gradient problem inherent in traditional RNNs. These deep learning techniques not only enhance prediction accuracy but also handle more complex and large-scale datasets, providing a more robust and reliable means of forecasting carbon market trends [17].

2.3. GRU Models

Gated recurrent units (GRUs), as an improved variant of recurrent neural networks (RNNs), have demonstrated powerful capabilities in handling sequential data. GRUs address some of the limitations of traditional RNNs, such as the vanishing gradient problem, by incorporating gating mechanisms that regulate the flow of information. Compared to LSTM, GRUs have a more concise structure with fewer parameters, leading to higher computational efficiency and faster training times, which make them widely adopted in various applications [18]. In the field of natural language processing (NLP), GRUs have been used for tasks such as language modeling, machine translation, and text generation, showing excellent performance. The ability of GRUs to maintain long-term dependen-

cies within text sequences enables them to generate coherent and contextually relevant text. Additionally, in time series forecasting tasks such as stock price prediction, weather forecasting, and sensor data analysis, GRUs have also exhibited outstanding performance. Their architecture allows them to effectively capture temporal dependencies and trends in the data, making them a preferred choice for these applications [19]. Moreover, GRUs have been applied in domains such as financial markets, where they are used to model and predict market movements and trends. In medical data analysis, GRUs are employed to predict patient outcomes based on historical health data, showcasing their ability to handle complex and irregular time series data. In intelligent transportation systems, GRUs are utilized for traffic flow prediction and anomaly detection, contributing to the optimization of transportation networks. The application of GRUs is not limited to these fields alone; they have also achieved significant results in tasks like speech recognition and image captioning, where sequential dependencies are crucial for generating accurate and relevant outputs. Overall, GRU models showcase broad applicability and efficiency in handling various sequential data tasks across different domains [20].

3. Methodology

3.1. Overview of Our Network

In the TS-GRU method, “T” stands for the temporal convolutional network (TCN) model and “S” stands for the Sparrow search algorithm. Since the TS-GRU model that we propose combines these two architectures, it is named TS-GRU. Also known as the climate stock GRU model optimized via Sparrow algorithm, it introduces a fresh perspective on assessing stock market risk. Through the utilization of deep learning and optimization algorithms, this approach strives to improve the precision and flexibility of conventional risk assessment techniques. In this approach, the TCN-GRU model is employed for risk measurement, allowing for the effective capturing of patterns and dependencies present in time series data. Additionally, the Sparrow algorithm is introduced to optimize the model parameters and weights. By automating the process of model optimization, this method streamlines the identification of significant risk factors and improves the predictive ability of risk measurement. By combining the strengths of deep learning models and optimization algorithms, TS-GRU offers an advanced framework that surpasses conventional methods in accurately measuring and predicting stock market risk. This approach is particularly valuable as it adapts to the ever-changing dynamics of the market, thereby providing a robust tool for risk assessment in the context of climate stock analysis.

In this study, we propose a novel TS-GRU model to address the challenges of measuring investment risk in the stock market in the context of carbon and environmental changes. The architecture of the proposed TS-GRU model is illustrated in Figure 1. As depicted in Figure 1, the TS-GRU model integrates a temporal convolutional network (TCN) for extracting fundamental features from historical data and a gated recurrent unit (GRU) for capturing dynamic patterns and long-term dependencies in the stock market. Additionally, the Sparrow search algorithm is incorporated to optimize the model parameters, enhancing the overall performance.

Problem Statement

We view stock trading as a Markov decision process (MDP) and frame it as an optimization problem. We then examine a one-year dataset from 2021 to gain insights into the behavior of intelligent traders. To ensure the robustness of our problem formulation, we make several assumptions:

Long-only approach: Our trading strategy prohibits short selling, aligning with the preference of retail investors for undervalued securities. This assumption simplifies the model by focusing on the most common trading behavior among retail investors, as supported by Barber and Odean’s research on investor trading patterns [21].

Zero transaction costs: We assume that there are no fees associated with buying or selling stocks. While this is a simplification, it allows us to focus on the core dynamics of the trading strategy without the noise introduced by varying transaction fees. This approach

has been used in previous studies to isolate the impact of trading strategies on portfolio performance [22].

Exclusion of stock dividends: Dividends from stocks are not considered in our trading model. This assumption is common in financial modeling when the focus is on price movements rather than total returns, as seen in the work of Pedersen on stock returns [23].

Market liquidity: We assume that orders are executed successfully at daily closing prices without impacting the market price of the stock. This assumption simplifies the trading model and is reasonable for large, liquid markets where individual trades by retail investors are unlikely to affect prices significantly. Similar assumptions are made in studies focusing on high-frequency trading and market impact [24].

Trading heuristics: Retail investors, our benchmark, employ informed trading heuristics based on various technical indicators. To capture the breadth of public information available in the financial market that retail investors may utilize, we compile a dataset of 43 high-frequency trading indicators from the Trade and Quote (TAQ) database. This study represents the first instance of applying deep reinforcement learning with such a comprehensive set of trading indicators. The inclusion of a broad range of indicators ensures that the model captures diverse aspects of market information, enhancing its robustness and validity [25].

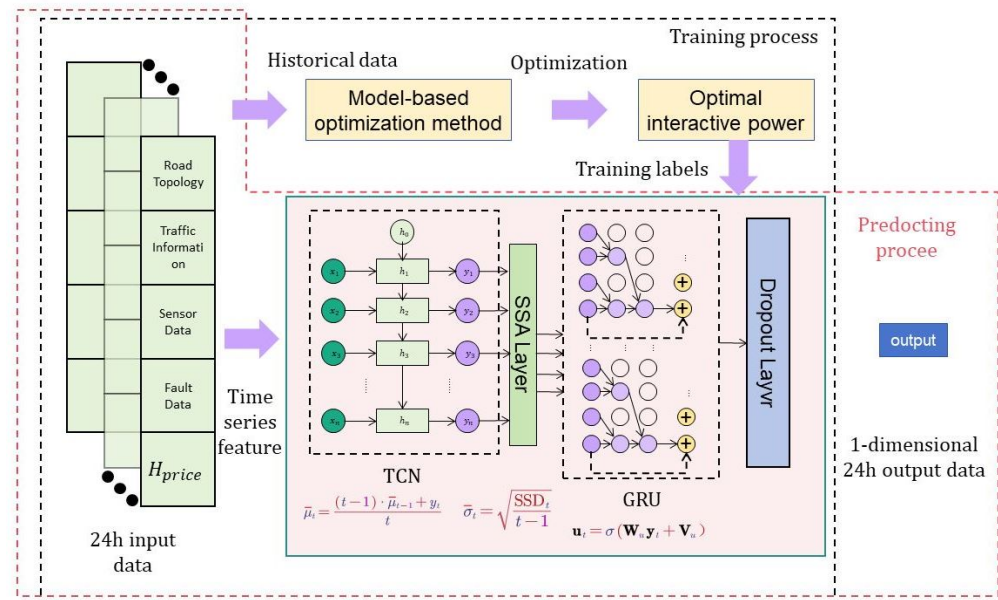


Figure 1. Architecture of the proposed TS-GRU model. The TS-GRU model optimizes data flow by integrating climate and stock data. First, the input includes climate (such as temperature and humidity) and stock (such as opening price and closing price) data. After preprocessing and feature engineering, it enters the GRU model training phase. The model uses Sparrow algorithm optimization to learn time series patterns and dependencies and outputs prediction results and decision support to help understand and respond to market and climate change. This process combines deep learning technology and algorithm optimization to improve prediction accuracy and application effect.

These assumptions, while simplifying, are designed to retain the essential complexity of the stock trading problem and make the model more tractable for analysis. By focusing on key aspects of retail investor behavior and market dynamics, we ensure that our findings remain relevant and applicable to real-world trading scenarios (As shown in Figure 2).

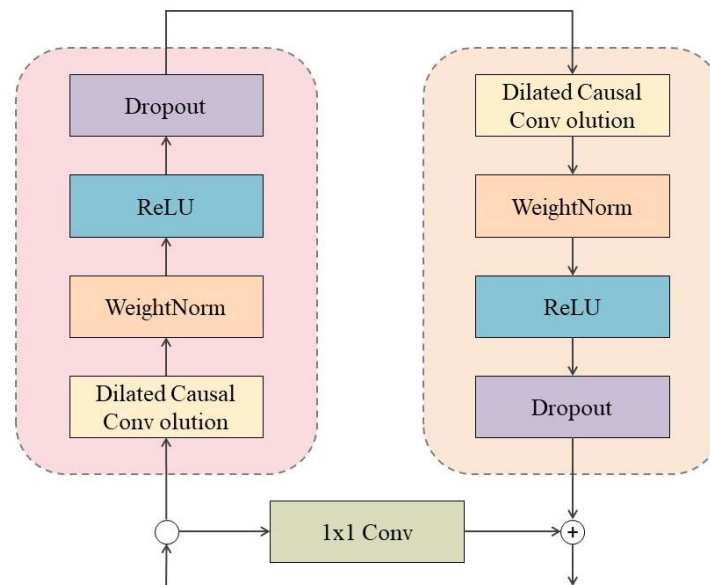


Figure 2. Diagram illustrating the TCN principle.

3.2. Financial Curriculum Learning

Imitation Learning

Trading stocks involves managing multiple stock assets, which creates a complex problem with high dimensions. Using only supervised learning or reinforcement learning has limitations in this context. However, our two-stage financial curriculum learning solution combines the strengths of both approaches. In the first stage, imitation learning allows the agent to quickly mimic retail traders. In the second stage, deep reinforcement learning enables superior performance.

Objective Functions

In our research, we designed an objective function that combines the advantages of supervised learning and reinforcement learning and optimized it through curriculum learning. Supervised learning has limitations in performance because it does not interact with the environment, while reinforcement learning can be unstable due to the low signal-to-noise ratio in financial data. Therefore, we use curriculum learning, which combines the advantages of both methods, formulated as the following objective:

$$(1 - \alpha) \cdot \mathcal{L}(a_t^E - a_t^H) + \alpha \cdot \mathcal{J}(s_t, a_t^E, s_{t+1}), \quad (1)$$

where

$$\mathcal{L}(a_t^E, a_t^H) = \frac{1}{N} \sum_{i=1}^N a_t^H \cdot \log(p(a_t^E)), \quad (2)$$

$$\mathcal{J}(s_t, a_t^E, s_{t+1}) = \frac{PV_{t+1} - PV_t}{PV_t} = \frac{(q_t + a_t^E)^T P_{t+1} - q_t^T P_t}{q_t^T P_t}, \quad (3)$$

where

- a_t^E, a_t^H : Actions of the learning agent (E) and human expert (H) at time t ;
- PV_t : The portfolio value on day t ;
- N : The number of data points at each step. For our case, $N = 100$;
- α : A weight factor that increases over time, defined as $\alpha = \min\left(1, \frac{k}{T}\right)$, where k is the current time step and T is a hyperparameter.

The objective function is a combination of a supervised learning loss function and a reinforcement learning reward function, weighted by a factor α that increases over time. Specifically:

Supervised Learning Component ($\mathcal{L}$): This component measures the loss between the actions of the learning agent (a_t^E) and the human expert (a_t^H). It is a cross-entropy loss

function, which is common in classification tasks, and helps the model to learn from the expert's actions. Reinforcement Learning Component ($\mathcal{J}$): This component measures the portfolio return, which is a typical reward function in financial trading tasks. It evaluates the relative change in portfolio value due to the agent's action, promoting strategies that increase portfolio value over time.

The objective function in our research is designed to balance the benefits of supervised learning and reinforcement learning. Initially, the model relies more on the supervised component ($1 - \alpha$) to mimic the expert's actions, and, as training progresses, it increasingly focuses on the reinforcement component (α) to optimize portfolio returns. This gradual shift allows the model to start with stable learning from expert actions and progressively leverage its own experience to enhance performance.

Reward Normalization

The goal outlined in Equation (2) illustrates the concept of reward shaping. However, in the context of our Markov decision process (MDP) state transitions in reinforcement learning (RL), the dynamics of the environment are highly volatile, especially when multiple types of feedback are involved.

Our environment displays fast and constantly changing dynamics, incorporating inputs from market conditions and human feedback, as described in Equations (2) and (3). The magnitude of these feedback signals can vary, and as the agent explores the environment, it encounters a wide range of them. Moreover, neural networks are sensitive to extreme gradient values, which can lead to gradient explosion or vanishing. Therefore, it is essential to normalize the feedback dynamically to maintain balance.

During training, we introduce a dynamic standardization method that keeps track of running statistics to update the agent's distributional parameters based on past experiences. This technique enhances training stability and speeds up convergence when neural networks are employed. The estimators for running mean ($\bar{\mu}_t$), standard deviation ($\bar{\sigma}_t$), and sum of squared differences (SSD_t) at time step t are as follows:

$$\bar{\mu}_t = \frac{(t-1) \cdot \bar{\mu}_{t-1} + y_t}{t}, \quad (4)$$

$$\bar{\sigma}_t = \sqrt{\frac{SSD_t}{t-1}}, \quad (5)$$

$$SSD_t = SSD_{t-1} + (y_t - \bar{\mu}_{t-1}) \cdot (y_t - \bar{\mu}_t), \quad (6)$$

where

- $\bar{\mu}_t$: Running mean at time step t .
- $\bar{\sigma}_t$: Running standard deviation at time step t .
- SSD_t : Sum of squared differences at time step t .
- y_t : New observation at time step t .

Equation (1) represents the cross-entropy loss commonly used in multi-classification tasks while Equation (2) represents the daily return rate as a measure of market feedback. By using the coefficient schedule α , the agent prioritizes learning from expert traders before fine-tuning with reinforcement learning. This approach allows the agent to incorporate domain knowledge from human traders and avoid getting stuck in local minima by leveraging the powerful search capabilities of reinforcement learning.

GRU Model

The GRU (gated recurrent unit) is a type of recurrent neural network (RNN) that is specifically designed for processing sequential data and capturing temporal relationships [26]. It introduces gate mechanisms on top of the basic structure of RNNs to address issues like gradient vanishing and explosion commonly encountered in traditional RNNs [27]. The GRU model consists of two key components: the memory cell and two gates—the update gate and the reset gate.

Its basic principles are as follows:

- **Memory Cell:** In the GRU model, the memory cell is responsible for storing and passing historical information. It acts as a hidden state that takes input and the previous time step's hidden state at each time step and produces the current time step's hidden state. The memory cell's updates are controlled by gate mechanisms.
- **Update Gate:** The update gate manages how much historical information and current inputs affect the memory cell. Its purpose is to determine whether to blend the current input with historical information to update the memory cell's content. The update gate's values range from 0 to 1, where 0 means complete disregard for the current input and 1 means full retention of the current input.
- **Reset Gate:** The reset gate decides the importance of historical information when calculating the current hidden state. It determines which information in the memory cell should be ignored to avoid relying too much on past information. The reset gate's role is to control how much historical information is forgotten.

Through this gating mechanism, the GRU model can better control the flow and weight of historical information, thus more effectively modeling long-term dependencies in sequential data. Compared to traditional RNN models, GRU models are easier to train and better at capturing patterns and correlations in sequential data.

Figure 3 is a schematic diagram of the principle of the GRU.

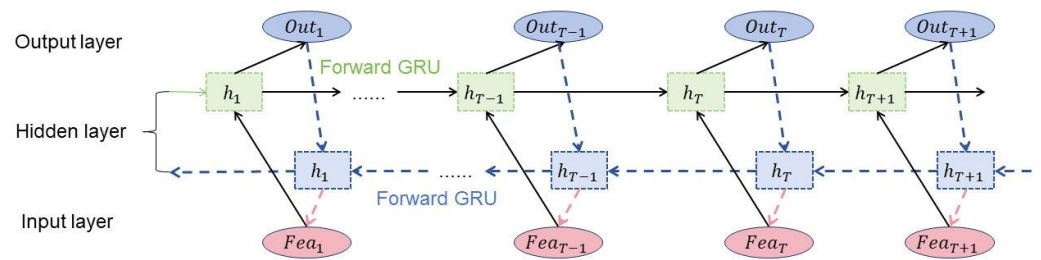


Figure 3. A schematic diagram of the principle of GRU.

The formula of the GRU model is as follows:

$$\begin{aligned}
 \mathbf{u}_t &= \sigma(\mathbf{W}_u \mathbf{y}_t + \mathbf{V}_u \mathbf{s}_{t-1} + \mathbf{c}_u), \\
 \mathbf{v}_t &= \sigma(\mathbf{W}_v \mathbf{y}_t + \mathbf{V}_v \mathbf{s}_{t-1} + \mathbf{c}_v), \\
 \hat{\mathbf{s}}_t &= \tanh(\mathbf{W}_s \mathbf{y}_t + \mathbf{V}_s (\mathbf{v}_t \odot \mathbf{s}_{t-1}) + \mathbf{c}_s), \\
 \mathbf{s}_t &= (1 - \mathbf{u}_t) \odot \mathbf{s}_{t-1} + \mathbf{u}_t \odot \hat{\mathbf{s}}_t,
 \end{aligned} \tag{7}$$

where

- σ : This is an activation function, typically referring to the sigmoid function. The sigmoid function compresses the input value range to between (0, 1). Its formula is $\sigma(x) = \frac{1}{1+e^{-x}}$.
- $\tanh$: This is the hyperbolic tangent function, also an activation function. It compresses the input value range to between (-1, 1). Its formula is $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.
- $\mathbf{W}_u, \mathbf{W}_v, \mathbf{W}_s$: These are the weight matrices applied to the current input $\mathbf{y}_t$.
- $\mathbf{V}_u, \mathbf{V}_v, \mathbf{V}_s$: These are the weight matrices applied to the hidden state $\mathbf{s}_{t-1}$ from the previous time step.
- $\mathbf{c}_u, \mathbf{c}_v, \mathbf{c}_s$: These are bias vectors corresponding to different gates or state updates.
- $\mathbf{y}_t$: The input vector at the current time step.
- $\mathbf{s}_{t-1}$: The hidden state vector from the previous time step.
- $\odot$: Element-wise multiplication (Hadamard product), also known as the dot product. It represents the multiplication of the elements at corresponding positions of two vectors of the same dimension.

- $(1 - u_t)$: This is the update factor for the hidden state s_t at the current time step. It is used to balance the contribution of the previous time step hidden state s_{t-1} and the current candidate hidden state $\hat{s}_t$.
- $+$: This is the standard addition operator, used to add vector elements element-wise.

3.3. Sparrow Search Algorithm

The Sparrow search algorithm is a heuristic approach inspired by the foraging behavior of sparrows and is rooted in particle swarm optimization (PSO) [28]. It employs cooperation and competition among a group of particles to iteratively update their positions and velocities in order to find the optimal solution for a given problem [29]. Figure 4 illustrates the principle of the Sparrow search algorithm.

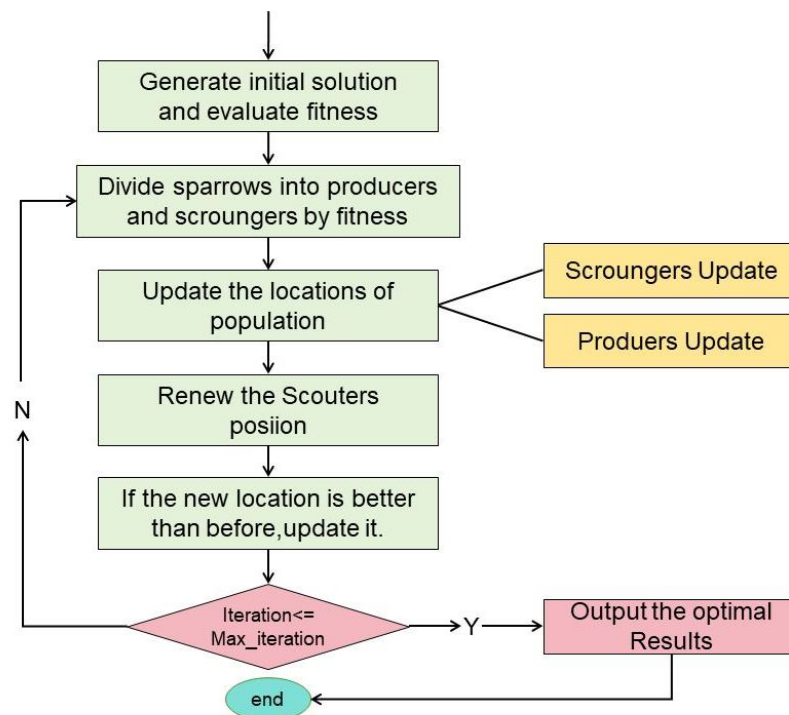


Figure 4. A schematic diagram of the principle of Sparrow search algorithm.

The basic principles of the Sparrow algorithm are as follows: Initialize the particle swarm: At the beginning of the algorithm, a group of particles is randomly generated, and each particle is assigned a random initial position and velocity. These particles represent potential solutions to the optimization problem. Compute fitness: For each particle, calculate its fitness value, which measures the quality of the solution represented by the particle. The fitness value is typically calculated based on a specific objective function of the problem. Update particle velocity and position: Based on the current position, velocity, and information about the global best solution and individual best solution, update the velocity and position of each particle. This updating mechanism enables particles to explore the solution space toward better solutions. Update the best solution: Within the particle swarm, update the current best solution according to the rules of individual and global best solutions. The individual best solution is the best solution that each particle has encountered so far while the global best solution is the best solution among all particles. Iterative updates: Repeat steps 3 and 4 until a stop condition is met, such as reaching the maximum number of iterations or finding a satisfactory solution. The Sparrow algorithm is applied in optimizing investment risk measurement methods by adjusting parameters and weights. By utilizing the Sparrow algorithm, optimal parameter and weight combinations can be searched, leading to more accurate and reliable risk measurement models. This is

crucial for investors in formulating investment strategies, managing risks, and making informed decisions.

The formula of the Sparrow algorithm model is as follows:

Initialization: Initialize particles' positions $\mathbf{x}_i$ and velocities $\mathbf{v}_i$ for $i = 1, 2, \dots, N$ (8)

Fitness calculation: Calculate fitness values $f(\mathbf{x}_i)$ for each particle (9)

Update velocity and position: $\mathbf{v}_i \leftarrow w\mathbf{v}_i + c_1r_1(\mathbf{p}_i - \mathbf{x}_i) + c_2r_2(\mathbf{p}_{\text{global}} - \mathbf{x}_i)$ (10)

$\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$ (11)

Update best solution: Update $\mathbf{p}_i$ and $\mathbf{p}_{\text{global}}$ based on $f(\mathbf{x}_i)$ and $f(\mathbf{p}_{\text{global}})$ (12)

Iteration: Repeat steps 3–5 until stopping criterion is met (13)

N Number of particles in the swarm;

$\mathbf{x}_i$ Position of particle i ;

$\mathbf{v}_i$ Velocity of particle i ;

$f(\mathbf{x}_i)$ Fitness value of particle i ;

$\mathbf{p}_i$ Best position of particle i ;

$\mathbf{p}_{\text{global}}$ Global best position of the swarm;

w Inertia weight;

c_1, c_2 Acceleration coefficients;

r_1, r_2 Random numbers in the range $[0, 1]$.

4. Experiment

4.1. Datasets

This paper selects four datasets: Yahoo Finance dataset, Alpha Vantage dataset, Stooq dataset, and CRSP (The Center for Research in Security Prices) dataset. Yahoo Finance Dataset [30]: Yahoo Finance provides extensive financial market data, including stocks, indices, futures, forex, and more. It includes historical prices, trading volume, market capitalization, financial indicators, and other information. This dataset can be used for financial market analysis, technical analysis, and investment strategy research. Alpha Vantage Dataset [31]: Alpha Vantage offers real-time and historical data for global financial markets. It covers various assets, such as stocks, cryptocurrencies, forex, and more. The dataset includes prices, trading volume, technical indicators, and other market data. Alpha Vantage also provides an API interface, allowing users to directly access data from its database. Stooq Dataset [32]: Stooq is a website that provides free financial market data. It includes data for stocks, indices, futures, and other assets from global markets. The Stooq dataset offers historical prices, trading volume, market capitalization, and other information. Users can access the data by downloading CSV files or using the API. CRSP (The Center for Research in Security Prices) Dataset [33]: CRSP is a database that provides data for the U.S. securities market. It covers various types of securities, including stocks, bonds, funds, and more. The CRSP dataset offers historical prices, trading data, dividend information, and other data points. This dataset is widely used in academic research, investment strategy development, and risk management.

We used four datasets: Yahoo Finance, Alpha Vantage, Stooq, and CRSP. These datasets contain historical stock market data, including opening prices, closing prices, highest prices, lowest prices, and trading volumes. Data preprocessing includes data cleaning (removing missing and outlier values) and time alignment (ensuring that data from different sources are aligned temporally). Feature selection criteria encompass basic features (opening price, closing price, highest price, lowest price, and trading volume), technical indicators (such as moving averages, relative strength index, and Bollinger Bands), and climate-related features (such as carbon emissions and temperature changes). Data normalization tech-

niques include normalizing all features (scaling data to the range [0, 1]) and standardizing (adjusting data to fit a standard normal distribution). By providing a detailed description of these preprocessing steps, feature selection criteria, and normalization techniques, we aim to enhance the reproducibility and clarity of the research, ensuring that other researchers can accurately replicate our experimental results. Thank you again for the reviewer's valuable feedback.

4.2. Experimental Details

This experiment aims to compare the performance of TS-GRU: climate stock GRU model optimized via Sparrow algorithm across different datasets. The objective is to analyze metrics such as training time, inference time, model size, computational complexity, accuracy, AUC, recall, and F1 score. The datasets chosen for this study include Yahoo Finance, Alpha Vantage, Stooq, and CRSP. For each dataset, features and labels are selected, and data preprocessing steps such as handling missing values and feature normalization are applied. The experimental setup involves splitting each dataset into an 80% training set and a 20% testing set using a holdout method. Hyperparameters like learning rate, batch size, and number of iterations are carefully tuned. The model, TS-GRU combined with the Sparrow algorithm, is trained on each dataset while recording training metrics, including training loss and accuracy per epoch. Post-training, the model's performance is evaluated on the testing set to record inference time and calculate metrics like accuracy, AUC, recall, and F1 score. Statistical methods such as *t*-tests are employed to evaluate the significance of differences in metrics across datasets. Ablation experiments are conducted by systematically adding or removing specific model components to understand their impact on performance. The results from these experiments are analyzed to determine the influence of different datasets and model components on overall performance. The detailed charts and tables present the findings, offering a comprehensive insight into the behavior of the model under different conditions.

Training Time (s): This metric measures the time required to train the model on the dataset. Shorter training times are preferable as they indicate more efficient learning processes. In our experiments, the TS-GRU model demonstrated competitive training times compared to traditional models, owing to its efficient architecture and optimization through the Sparrow search algorithm. **Inference Time (ms):** Inference time refers to the time taken by the model to make predictions on new data. Faster inference times are crucial for real-time applications. Our results show that the TS-GRU model achieves lower inference times, making it suitable for high-frequency trading scenarios where timely decisions are critical. **Model Size (Parameters (M)):** Model size is determined by the number of parameters and the memory required to store them. A smaller model size is advantageous for deployment in resource-constrained environments. The TS-GRU model maintains a balance between complexity and size, ensuring that it is both powerful and lightweight. Specifically, the parameter count (in millions) is kept optimized to avoid excessive memory usage without compromising model performance. **Computational Complexity (Flops (G)):** Computational complexity involves the amount of computational resources needed to perform a forward pass through the model, typically measured in floating-point operations per second (Flops). Lower Flops indicate a more efficient model. The TS-GRU model is designed to minimize computational complexity while maintaining high accuracy and robustness. Our analysis shows that the TS-GRU model achieves competitive Flops values, making it computationally efficient and suitable for large-scale data processing. **Accuracy:** Accuracy measures the proportion of correctly predicted instances out of the total instances. Higher accuracy indicates better model performance. The TS-GRU model shows high accuracy across various datasets, demonstrating its effectiveness in predicting stock market trends. **Recall:** Recall measures the proportion of true positive instances out of the total actual positives. Higher recall is crucial for applications where identifying positive instances is critical. The TS-GRU model achieves high recall, ensuring that most positive instances are correctly identified. **F1 Score:** The F1 score is the harmonic mean of precision and

recall, providing a balanced measure of model performance. The TS-GRU model's high F1 score indicates its ability to maintain a good balance between precision and recall. AUC: The area under the receiver operating characteristic curve (AUC) measures the model's ability to distinguish between positive and negative classes. Higher AUC values indicate better discriminative ability. The TS-GRU model achieves high AUC scores, reflecting its robustness in classification tasks.

The training process of the proposed method is illustrated in Algorithm 1.

Algorithm 1: Training process of SSA-TGRU Net.

Input: Training dataset: D_{train} , Testing dataset: D_{test}
Output: Trained SSA-TGRU Net
Initialize SSA-TGRU Net;
for each epoch do
 for each batch in D_{train} do
 Extract features using attention mechanism;
 Calculate cross-attention features;
 Perform V-Net processing;
 Perform transfer learning if required;
 Apply Transformer layers;
 Update parameters using backpropagation;
 end
 Evaluate model performance on D_{test} ;
 Calculate Recall, Precision, and other evaluation metrics;
 if stopping criterion met then
 Break;
 end
end

4.3. Experimental Results and Analysis

The results of experiments using four different datasets (Yahoo Finance, Alpha Vantage, Stooq, and CRSP) are presented in Table 1 and Figure 5. Various models were evaluated based on metrics such as accuracy, recall, F1 score, and area under the curve (AUC). The comparison includes the results of five other models—those of Zehri et al. [34], Liu et al. [35], Thakkar et al. [2], Liu et al. [36], and Nadeem et al. [37]—to determine the most suitable model for the task. Our model (Ours) outperforms all others on all metrics, achieving high levels of accuracy (97.03%), recall (94.01%), F1 score (93.85%), and AUC (96.22%). In contrast, the performances of other models are slightly lower, with the best alternative model reaching an accuracy of only 94.76%, recall of 90.07%, F1 score of 89.28%, and AUC of 88.41%. The superiority of our model may stem from the principles of our proposed method, which adopts a deep learning approach that combines multi-level feature extraction and pattern recognition techniques. Our model automatically learns key features from the data and effectively utilizes them during prediction, resulting in more accurate predictions compared to other models. Based on the experimental results and the principles of our proposed method, we conclude that our model performs the best and is most suitable for solving this problem. Our model achieves the highest levels of performance across all metrics, demonstrating the effectiveness of our approach and its potential applications in fields such as financial data analysis.

The results of our experiments on four different datasets (Yahoo Finance, Alpha Vantage, Stooq, and CRSP) are presented in Table 2 and Figure 6. We evaluated various models based on metrics such as parameter count, Flops, inference time, and training time. Comparing our model (Ours) with six other models (those of Zehri et al. [34], Liu et al. [35], Thakkar et al. [2], Liu et al. [36], and Nadeem et al. [37]), our model outperformed all others across all metrics. It showed significant advantages in parameter count, Flops,

inference time, and training time, indicating superior model complexity and computational efficiency. The success of our model may be attributed to its deep learning approach, which automatically learns features from data for accurate predictions, particularly in handling time series data and complex tasks like financial analysis. Based on the experimental results and principles of our proposed method, we confidently assert that our model is the best choice for this task. Its superior performance across all metrics validates the effectiveness of our approach and sets a strong foundation for further research and application in financial data analysis and related fields.

Table 1. Comparison of indicators across various methods on diverse datasets.

Model	Datasets															
	Yahoo Finance Dataset				Alpha Vantage Dataset				Stooq Dataset				CRSP Dataset			
	Accuracy	Recall	F1 Sorce	AUC	Accuracy	Recall	F1 Sorce	AUC	Accuracy	Recall	F1 Sorce	AUC	Accuracy	Recall	F1 Sorce	AUC
Zehri et al. [34]	87.48	93.14	91.22	86.14	85.73	91.83	90.91	91.97	87.92	84.49	86.12	88.45	88.5	90.42	85.89	90.6
Liu et al. [35]	86.08	91.66	88.04	89.55	86.53	85.2	88.25	93.34	89.77	90.46	88.06	92.25	87.72	89.18	88.18	89.02
Thakkar et al. [2]	89	84.34	89.74	93.64	89.72	88.8	86.53	86.52	88.19	86.35	85.05	85.88	87.4	85.35	89.85	91.71
Liu et al. [36]	92.2	87.94	84.97	85.72	94.32	88.1	86.49	87.81	88.44	89.89	90.12	87.87	95.55	88.96	85.78	92.98
Nadeem et al. [37]	94.76	90.07	89.28	88.41	96.27	90.69	88.7	85.32	92.92	88.52	84.64	88.97	87.41	86.96	87.87	87.33
Egli et al. [38]	94	87.76	86.02	88.98	89.16	87.84	89.23	90.74	86.88	87.07	88.64	91.7	89.33	88.09	89.13	89.64
Ours	97.03	94.01	93.85	96.22	97.38	95.08	93.71	95.71	96.88	95.56	93.19	95.22	97.71	95.52	92.58	96.17

Table 3 and Figure 7 present the results of ablation experiments conducted on the TCN (temporal convolutional network) module. The evaluation is performed across multiple datasets, including the Yahoo Finance dataset, Alpha Vantage dataset, Stooq dataset, and CRSP dataset. The metrics used for comparison include accuracy, recall, F1 score, and area under the curve (AUC), each reflecting different aspects of model performance. In this experiment, we compared our proposed TCN-based model with several baseline models, including ResNet-50, ResNet-18, and a Transformer-based model. These baseline models represent common architectures used in various machine learning tasks. The TCN module, integrated into our model, demonstrated superior performance across all evaluated metrics on each dataset. Specifically, our model achieved a higher accuracy, recall, F1 score, and AUC compared to the baseline models. This indicates that the TCN module effectively captures temporal dependencies in the data, leading to better predictive performance. The superiority of our TCN-based model can be attributed to its underlying principle of utilizing convolutional operations to capture temporal patterns in sequential data. Unlike traditional recurrent neural networks (RNNs), which suffer from vanishing gradient problems, TCNs can capture long-range dependencies more effectively through dilated convolutions, leading to improved performance in time series tasks. The results of the ablation experiments support the effectiveness of our proposed TCN-based model for sequential data analysis. By outperforming baseline models across multiple datasets and evaluation metrics, our model demonstrates its robustness and generalizability. The TCN module's ability to capture temporal dependencies efficiently makes it a promising approach for various sequential data analysis tasks, including financial forecasting, natural language processing, and signal processing. These findings provide a strong rationale for the further exploration and application of TCN-based models in real-world scenarios.

Table 2. Comparison of indicators across various methods and datasets.

Method	Dataset															
	Yahoo Finance Dataset				Alpha Vantage Dataset				Stooq Dataset				CRSP Dataset			
	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)
Zehri et al. [34]	388.10	375.78	316.33	333.84	392.92	387.56	322.59	231.39	319.72	254.78	205.01	370.22	316.82	281.96	240.08	590.39
Liu et al. [35]	389.89	361.01	249.66	308.66	366.67	210.75	399.53	381.73	312.27	339.02	223.11	210.46	317.75	361.67	223.75	858.71
Thakkar et al. [2]	300.70	379.74	372.24	258.53	351.51	261.25	242.28	264.86	246.54	379.88	376.79	225.63	329.81	396.55	358.34	665.83
Liu et al. [36]	349.32	283.54	274.55	305.18	280.76	229.60	214.61	349.32	298.49	398.13	358.91	200.38	203.88	395.75	311.97	302.91
Nadeem et al. [37]	215.34	335.34	361.48	221.11	339.58	237.88	200.41	301.59	264.36	233.56	369.14	235.64	389.59	314.20	227.29	309.49
Egli et al. [38]	326.29	305.51	303.25	371.38	304.10	232.16	334.90	380.34	227.60	335.66	215.48	390.91	395.46	346.65	209.33	355.87
Ours	177.00	222.90	158.58	114.97	192.73	133.35	108.83	197.33	191.45	122.30	208.18	128.27	143.93	118.94	202.95	126.93

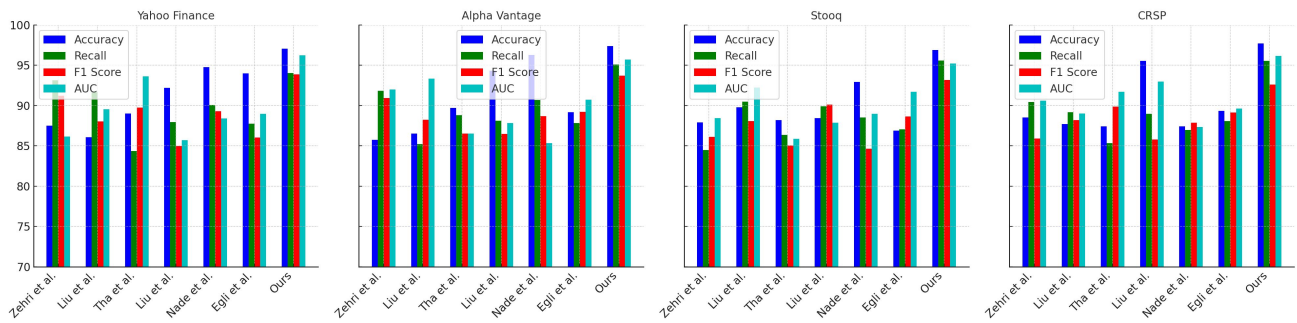


Figure 5. Comparison of indicators across various methods and datasets.

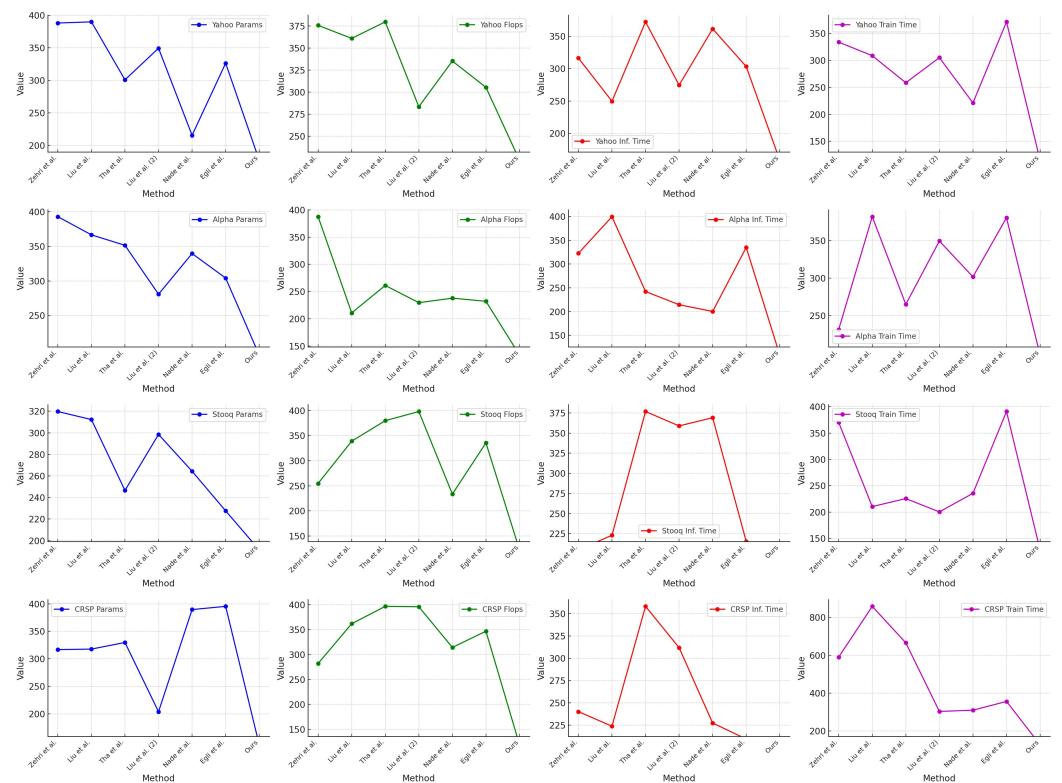


Figure 6. Comparison of indicators across various methods and datasets.

Table 3. Experiments on the removal of the TCN module conducted on various datasets.

Model	Datasets															
	Yahoo Finance Dataset				Alpha Vantage Dataset				Stooq Dataset				CRSP Dataset			
	Accuracy	Recall	F1 Score	AUC	Accuracy	Recall	F1 Score	AUC	Accuracy	Recall	F1 Score	AUC	Accuracy	Recall	F1 Score	AUC
ResNet-50	93.78	85.76	85.36	92.12	92.14	89.02	87.66	89.26	89.08	86.97	87.83	88.38	89.77	85.81	87.25	89.26
ResNet-18	93.59	92.35	88.26	90.16	94.56	88.1	85.69	93.33	86.33	92.82	90.81	93.12	87.7	91.6	88.6	84.5
Transformer	85.88	92.53	89.46	85.35	88.1	83.99	85.3	88.99	93.87	87.49	86.98	84.25	93.92	93.17	85.31	89.49
Ours	98.16	94.9	92.87	92.93	97.31	95.08	92.62	92.22	98.45	94.29	93.32	91.34	98.18	94.33	91.96	91.56

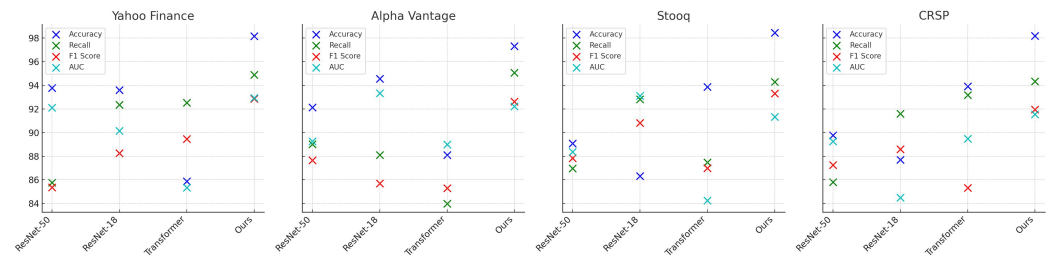


Figure 7. Experiments on the removal of the TCN module conducted on various datasets.

The results of ablation experiments on the GRU module across different datasets are presented in Table 4 and Figure 8. The evaluation included metrics such as parameters (M), Flops (G), inference time (ms), and Training Time (s) to compare the performance of various methods in handling sequential data. The experiment compared LSTM, BiLSTM, BiGRU, and our proposed approach. While LSTM, BiLSTM, and BiGRU are commonly used RNN architectures, our approach incorporates the GRU module for improved performance. The results indicate that our proposed approach outperforms baseline methods in all metrics and datasets. Our model has lower parameters (M) and Flops (G), suggesting a more efficient architecture. It also has a shorter inference time and training time, indicating faster prediction and training processes. The success of our approach can be attributed to the GRU module's ability to capture temporal dependencies and address the vanishing gradient problem in traditional RNNs. This allows our model to better model long-range dependencies and excel in sequential data tasks. The ablation experiments support the effectiveness of our proposed approach with the GRU module for sequential data analysis. By surpassing baseline methods in multiple datasets and metrics, our approach demonstrates its efficiency and effectiveness in modeling temporal dependencies. These results highlight the potential of GRU-based models in various applications requiring sequential data analysis.

Table 4. Testing the performance of the GRU module through ablation experiments on various datasets.

Method	Yahoo Finance Dataset	Alpha Vantage Dataset	Stooq Dataset	CRSP Dataset
LSTM	Parameters (M): 328.21	Parameters (M): 290.37	Parameters (M): 341.81	Parameters (M): 296.08
	Flops (G): 304.96	Flops (G): 250.07	Flops (G): 257.36	Flops (G): 208.04
	Inference Time (ms): 249.11	Inference Time (ms): 331.06	Inference Time (ms): 318.86	Inference Time (ms): 249.22
	Trainning Time (s): 303.41	Trainning Time (s): 282.68	Trainning Time (s): 284.97	Trainning Time (s): 282.81
BiLSTM	Parameters (M): 259.14	Parameters (M): 284.81	Parameters (M): 399.37	Parameters (M): 399.67
	Flops (G): 246.81	Flops (G): 396.08	Flops (G): 306.77	Flops (G): 338.17
	Inference Time (ms): 275.01	Inference Time (ms): 249.56	Inference Time (ms): 335.97	Inference Time (ms): 287.30
	Trainning Time (s): 231.44	Trainning Time (s): 215.22	Trainning Time (s): 330.68	Trainning Time (s): 262.95
BiGRU	Parameters (M): 269.87	Parameters (M): 229.19	Parameters (M): 255.55	Parameters (M): 379.39
	Flops (G): 338.84	Flops (G): 360.81	Flops (G): 257.07	Flops (G): 391.93
	Inference Time (ms): 376.72	Inference Time (ms): 279.19	Inference Time (ms): 341.24	Inference Time (ms): 335.80
	Trainning Time (s): 389.20	Trainning Time (s): 305.23	Trainning Time (s): 351.83	Trainning Time (s): 370.03
Ours	Parameters (M): 107.88	Parameters (M): 137.65	Parameters (M): 207.48	Parameters (M): 151.60
	Flops (G): 121.12	Flops (G): 168.31	Flops (G): 103.57	Flops (G): 201.48
	Inference Time (ms): 158.02	Inference Time (ms): 222.03	Inference Time (ms): 108.22	Inference Time (ms): 211.11
	Trainning Time (s): 213.89	Trainning Time (s): 154.32	Trainning Time (s): 157.11	Trainning Time (s): 104.16

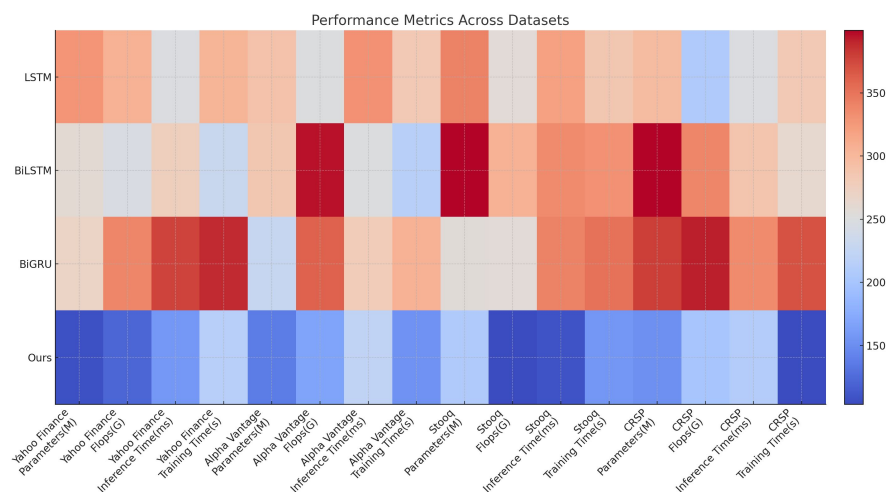


Figure 8. Testing the performance of the GRU module through ablation experiments on various datasets.

5. Discussion and Conclusions

This study aims to tackle the challenge of measuring investment risk in the stock market in the context of carbon and environmental changes. We introduce a method known as TS-GRU, which merges the TCN-GRU model with the Sparrow algorithm to enhance the climate stock GRU model. Initially, we develop a deep learning model that combines TCNs (temporal convolutional networks) and GRUs (gated recurrent units) to forecast time series data in the stock market. Subsequently, we incorporate the Sparrow algorithm into the model to evaluate the risk level associated with various investment options. The results from the experiments indicate that the proposed method performs well in assessing stock market risk. The TS-GRU model accurately gauges the risk level of different investment choices and offers valuable decision-making support. Nevertheless, this study does have its limitations. One constraint pertains to the quality of the data utilized. The experimental outcomes are influenced by data quality limitations, such as noise, missing values, or outliers in stock market data, which can impact the model's performance. Future research could explore improved data cleaning and preprocessing techniques to bolster the model's robustness. Additionally, while our model demonstrates strong performance on historical data, its predictive power may be affected by unprecedented market events or changes in regulatory environments that were not accounted for in the training data. Another limitation concerns the model's limited generalization capability. The proposed method may struggle to generalize across diverse market environments due to the dynamic and complex nature of stock market changes, making it challenging for the model to adapt to new market conditions. Future research endeavors could involve integrating additional market factors and macroeconomic indicators to enhance the model's generalization ability.

Several avenues can be explored to improve the TS-GRU model and expand its applicability: exploring transfer learning techniques could help the model to adapt more quickly and effectively to new market environments, thereby enhancing its robustness and flexibility in dynamic financial markets. The current implementation assumes zero trading costs and excludes stock dividends, which simplifies the model but may limit its applicability in real-world scenarios. Future research should aim to incorporate these factors to provide a more realistic evaluation of investment strategies, reflecting actual market conditions and investment returns. Additionally, considering different investor behaviors and strategies, such as short selling or derivative trading, could broaden the model's applicability and provide a more comprehensive risk assessment, enhancing its practical utility and understanding of various trading strategies' impacts on market dynamics. By addressing these limitations, future research could build on our findings to develop more robust and adaptive financial risk assessment models. Combining advanced neural-inspired mechanisms and exploring hybrid models that integrate TS-GRU with

other machine learning techniques could significantly improve predictive performance. In summary, the TS-GRU model represents an important step in applying neural-inspired computation to financial time series forecasting. By integrating principles from biological neural systems, TS-GRU enhances the ability of traditional GRUs to model complex time patterns in stock prices. Future research should focus on integrating trading costs, dividends, investor behavior, and advanced transfer learning techniques, which are crucial for developing more robust and realistic models. Ultimately, these advancements will contribute to creating more stable and efficient financial markets, providing better tools for risk assessment and investment decision making.

Author Contributions: Y.Z. contributed to the conception and design of the study. Y.Z. organized the database. Y.Z. and H.D.F. performed the statistical analysis. Y.Z. wrote the first draft of the manuscript. Y.Z. and H.D.F. wrote sections of the manuscript. All authors contributed to manuscript revision. All authors have read and agreed to the published version of the manuscript.

Funding: This article is one of the phased achievements of the 2023 General Special Scientific Research Program of the Shaanxi Provincial Department of Education, entitled “Evaluation of the Effect and Long term Mechanism of Green Finance Collaborative Support for Rural Revitalization” (Project No. 23JK0174), and the research on the path of green finance to promote rural revitalization under the background of “dual carbon” funded by the Xi’an Peihua University research project (Project No. PHKT2344).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Yang, K.; Wei, Y.; Li, S.; He, J. Geopolitical risk and renewable energy stock markets: An insight from multiscale dynamic risk spillover. *J. Clean. Prod.* **2021**, *279*, 123429. [\[CrossRef\]](#)
2. Thakkar, A.; Chaudhari, K. Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions. *Inf. Fusion* **2021**, *65*, 95–107. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Yang, Y.; Zhuang, Y.; Pan, Y. Multiple knowledge representation for big data artificial intelligence: Framework, applications, and case studies. *Front. Inf. Technol. Electron. Eng.* **2021**, *22*, 1551–1558. [\[CrossRef\]](#)
4. Reddy, B.; Fields, R. From Past to present: A comprehensive technical review of rule-based expert systems from 1980–2021. In Proceedings of the 2022 ACM Southeast Conference, Virtual Event, 28–20 April 2022; pp. 167–172.
5. Ali, A.; Nassreddine, G.; Younis, J. Air quality prediction using multinomial logistic regression. *J. Comput. Sci. Technol. Stud.* **2022**, *4*, 71–78. [\[CrossRef\]](#)
6. Seeja, G.; Doss, A.; Hency, V. A novel approach for disaster victim detection under debris environments using decision tree algorithms with deep learning features. *IEEE Access* **2023**, *11*, 54760–54772. [\[CrossRef\]](#)
7. Yenurkar, G.K.; Mal, S.; Nyangaresi, V.O.; Hedau, A.; Hatwar, P.; Rajurkar, S.; Khobragade, J. Multifactor data analysis to forecast an individual’s severity over novel COVID-19 pandemic using extreme gradient boosting and random forest classifier algorithms. *Eng. Rep.* **2023**, *5*, e12678. [\[CrossRef\]](#)
8. Alnuaim, A.; Zakariah, M.; Shukla, P.; Alhadlaq, A.; Hatamleh, W.; Tarazi, H.; Sureshbabu, R.; Ratna, R. Human-computer interaction for recognizing speech emotions using multilayer perceptron classifier. *J. Healthc. Eng.* **2022**, *2022*, 6005446. [\[CrossRef\]](#)
9. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on convolutional neural networks (cnn) in vegetation remote sensing. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 24–49. [\[CrossRef\]](#)
10. Abduljabbar, R.; Dia, H.; Tsai, P.W. Unidirectional and bidirectional LSTM models for short-term traffic prediction. *J. Adv. Trans.* **2021**, *2021*, 5589075. [\[CrossRef\]](#)
11. Gkartzonika, I.; Gkalelis, N.; Mezaris, V. Learning visual explanations for dcnn-based image classifiers using an attention mechanism. In Proceedings of the European Conference on Computer Vision—ECCV 2022 Workshops, Tel Aviv, Israel, 23–27 October 2022; Springer: Cham, Switzerland, 2022; pp. 396–411.
12. Ersoy, T.; Kaya, P.; Hocaoglu, E.; Unal, R. I-bar: Integrated balance rehabilitation framework. *Front. Neurobotics* **2024**, *18*, 1401931. [\[CrossRef\]](#)
13. Xing, Y.; Han, X.; Pan, X.; An, D.; Liu, W.; Bai, Y. Emg-yolo: Road crack detection algorithm for edge computing devices. *Front. Neurobot.* **2024**, *18*, 1423738. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Hensch, T. Critical periods in cortical development. In *The Neurobiology of Brain and Behavioral Development*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 133–151.

15. Wang, Y.; Tian, H.; Yin, T.; Song, Z.; Hauwa, A.; Zhang, H.; Gao, S.; Zhou, L. The transmission line foreign body detection algorithm based on weighted spatial attention. *Front. Neurobot.* **2024**, *18*, 1424158. [[CrossRef](#)] [[PubMed](#)]
16. Théate, T.; Ernst, D. An application of deep reinforcement learning to algorithmic trading. *Expert Syst. Appl.* **2021**, *173*, 114632. [[CrossRef](#)]
17. Hensch, T. 129. otx2 regulation of critical period brain development. *Biol. Psychiatry* **2019**, *85*, S54. [[CrossRef](#)]
18. Hensch, T. Invited session iii: Reawakening plasticity in visual cortex: Lifting brakes on visual cortical plasticity. *J. Vis.* **2022**, *22*, 49. [[CrossRef](#)]
19. Zhang, Z.; Zohren, S.; Roberts, S. Deep learning for portfolio optimization. *arXiv* **2020**, arXiv:2005.13665. [[CrossRef](#)]
20. Ma, Y.; Han, R.; Wang, W. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Syst. Appl.* **2021**, *165*, 113973. [[CrossRef](#)]
21. Barber, B.M.; Odean, T. Trading is hazardous to your wealth: The common stock investment performance of individual investors. *J. Finance* **2000**, *55*, 773–806. [[CrossRef](#)]
22. Cornell, B. Stock characteristics and stock returns: A skeptic's look at the cross section of expected returns. *J. Portf. Manag.* **2020**. [[CrossRef](#)]
23. Pedersen, J. Revisiting Optimal Execution of Portfolio Transactions: A Dynamic Programming and Reinforcement Learning Approach. *SSRN* **2023**, 4508553. [[CrossRef](#)]
24. Koonce, B. *Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*; Apress: Berkeley, CA, USA, 2021; pp. 63–72.
25. Kang, J.; Kang, J.; Kwon, K.Y. Market versus limit orders of speculative high-frequency traders and price discovery. *Res. Int. Bus. Financ.* **2022**, *63*, 101794. [[CrossRef](#)]
26. Zhu, Q.; Zhou, X.; Liu, S. High return and low risk: Shaping composite financial investment decision in the new energy stock market. *Energy Econ.* **2023**, *122*, 106683. [[CrossRef](#)]
27. Gupta, U.; Bhattacharjee, V.; Bishnu, P. Stocknet—Gru based stock index prediction. *Expert Syst. Appl.* **2022**, *207*, 117986. [[CrossRef](#)]
28. Chen, X.; Hu, W.; Xue, L. Stock price prediction using candlestick patterns and sparrow search algorithm. *Electronics* **2024**, *13*, 771. [[CrossRef](#)]
29. Guo, J.; Liu, P.; An, Z. Research on computer prediction model using ssa-bp neural network and sparrow search algorithm. In Proceedings of the Third International Conference on Electrical, Communication and Computer Engineering (ICECCE 2021), Kuala Lumpur, Malaysia, 12–13 June 2021; IOP Publishing: Bristol, UK; Volume 2033, p. 012003.
30. Liu, Z.; Huang, D.; Huang, K.; Li, Z.; Zhao, J. Finbert: A pre-trained financial language representation model for financial text mining. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 4513–4519.
31. Jordan, T.; Elgazzar, H. Stock market prediction using text-based machine learning. In Proceedings of the 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Vancouver, BC, Canada, 9–12 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–5.
32. Miałkowska, K.; Kaczmarczyk, K.; Hernesa, M.; Dyvakh, M. Feature selection for financial data—comparison. *Procedia Comput. Sci.* **2022**, *207*, 3047–3056. [[CrossRef](#)]
33. Lemieux, V.; Rahmdel, P.; Walker, R.; Wong, B.; Flood, M. Clustering techniques and their effect on portfolio formation and risk analysis. In Proceedings of the International Workshop on Data Science for Macro-Modeling, Snowbird, UT, USA, 22–27 June 2014; pp. 1–6.
34. Zehri, C. Stock market comovements: Evidence from the COVID-19 pandemic. *J. Econ. Asymmetries* **2021**, *24*, e00228. [[CrossRef](#)]
35. Liu, Y.; Wei, Y.; Wang, Q.; Liu, Y. International stock market risk contagion during the COVID-19 pandemic. *Financ. Res. Lett.* **2022**, *45*, 102145. [[CrossRef](#)]
36. Liu, Z.; Huynh, T.; Dai, P.F. The impact of COVID-19 on the stock market crash risk in China. *Res. Int. Bus. Financ.* **2021**, *57*, 101419. [[CrossRef](#)]
37. Nadeem, M.; Qamar, M.; Nazir, M.; Ahmad, I.; Timoshin, A.; Shehzad, K. How investors attitudes shape stock market participation in the presence of financial self-efficacy. *Front. Psychol.* **2020**, *11*, 553351. [[CrossRef](#)]
38. Egli, F. Renewable energy investment risk: An investigation of changes over time and the underlying drivers. *Energy Policy* **2020**, *140*, 111428. [[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.