International Journal of Data Science and Analytics (2026) 21:29
https://doi.org/10.1007/s41060-025-00915-8

REGULAR PAPER

Adv-SHNets: a stock movement prediction framework based on hierarchy LSTM networks

Quannan Zu[1] · Nannan Wu[2] · Wei Yu[2,4] · Huaming Wu[3] · Yanxia Zhao[5] · Kehan Long[6] · Sibo Yang[1] · Lei Wang[2]

Received: 1 June 2025 / Accepted: 8 October 2025

Abstract

Stock data are inherently time-series data, characterized by complex structures and high levels of randomness. This makes forecasting stock prices a challenging problem in machine learning. The non-stationary nature of stock indices in financial markets, coupled with randomness and uncertainty in the data, and the simple hierarchical structure and lack of semantic information in the temporal dimension pose significant challenges to traditional time-series forecasting models. This study therefore proposes a novel stock price forecasting framework based on a hierarchical LSTM model to address these issues, which is known as Adv-SHNets. The model's core concept involves learning the temporal features of stock data through two layers of LSTMs to fully capture its hierarchical structure. The framework processes stock time series via overlapping segmentation in the lower LSTM layer to preserve temporal continuity and enrich single-step semantics. Subsequently, a segmented attention mechanism and upper LSTM capture multi-scale hierarchical dependencies. A simple yet effective adversarial loss function significantly enhances the model's robustness against data uncertainty. Furthermore, experiments on two real-world stock datasets demonstrate that Adv-SHNets achieves a relative improvement of 12.97% over comparable state-of-the-art methods, providing thorough validation of the effectiveness of this approach.

Keywords Stock movement prediction · CNN · Multi-level LSTM · Segmentation attention · Adversarial learning

1 Introduction

Stock market forecasts can help investors make decisions and minimize risks as much as possible, thereby obtaining investment returns. For example, when the market becomes bearish, investors can sell stocks to maintain relatively low losses. Moreover, forecasting the future condition of a stock has consistently captivated the attention of numerous participants. As we know, the exact price of a stock is usually unpredictable [1, 2], research endeavors have concentrated on forecasting the movement of stock prices, e.g., determining whether the price will rise or fall, or if the price fluctuations

✉ Wei Yu
weiyu@tju.edu.cn

✉ Huaming Wu
whming@tju.edu.cn

Quannan Zu
quannan_zu@tju.edu.cn

Nannan Wu
nannan.wu@tju.edu.cn

Yanxia Zhao
20212015@zyufl.edu.cn

Kehan Long
longkhan@connect.hku.hk

Sibo Yang
yangsibo@tju.edu.cn

Lei Wang
wanglei2019@tju.edu.cn

1 College of Management and Economics, Tianjin University, Tianjin 300072, Tianjin, China

2 College of Intelligence and Computing, Tianjin University, Tianjin 300072, Tianjin, China

3 Center for Applied Mathematics, Tianjin University, Tianjin 300072, Tianjin, China

4 College of International Business, Zhejiang Yuexiu University, Shaoxing 312000, Zhejiang, China

5 School of Public Administration, Zhejiang Gongshang University, Hangzhou 310018, Zhejiang, China

6 Department of Computer Science, Faculty of Engineering, The University of Hong Kong, Pok Fu Lam 999077, Hong Kong S.A.R., China

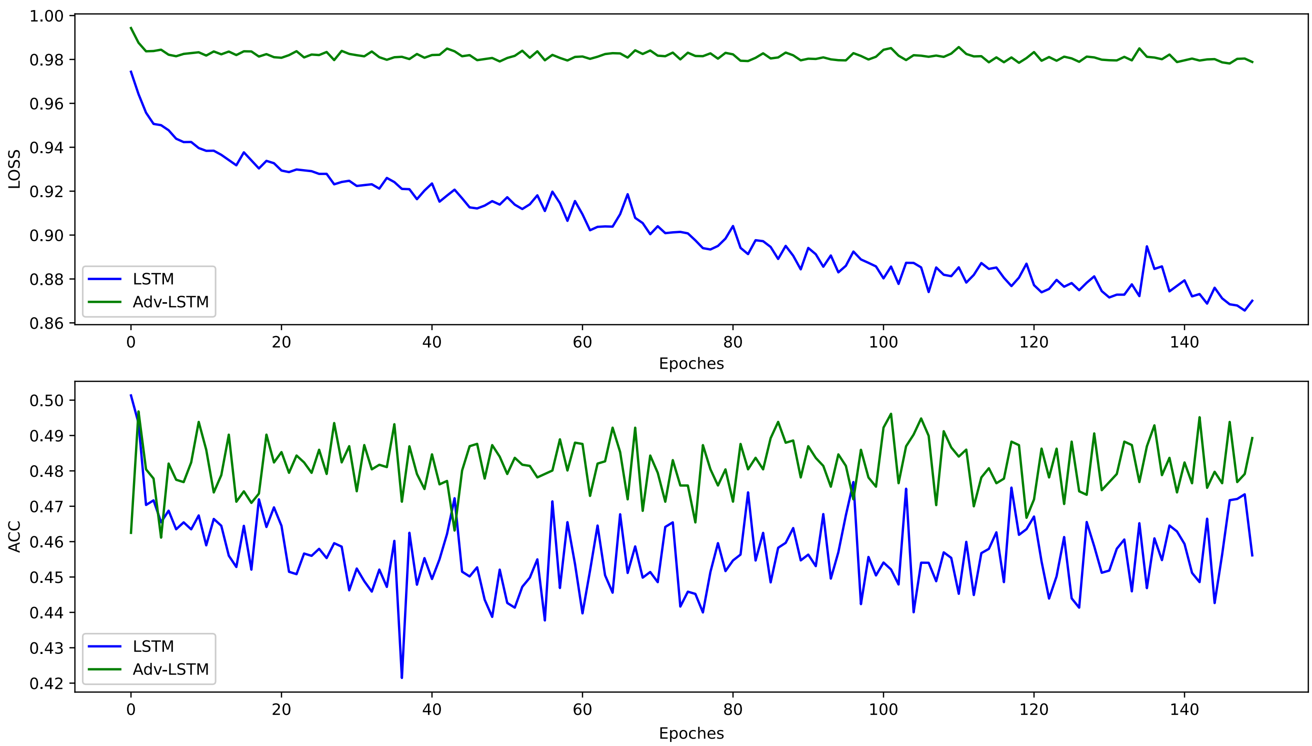
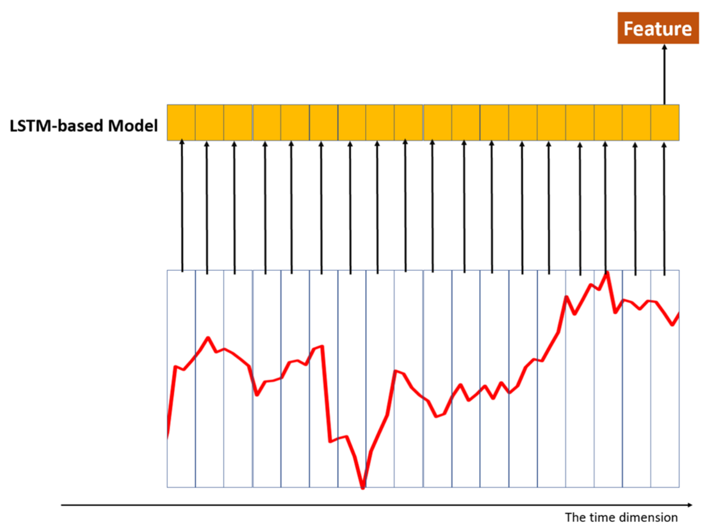


Fig. 1 The training trajectories of LSTM and Adv-ALSTM on the training dataset ACL18. The top figure shows the training Hinge Loss [18] on the training set as the training progresses, while the bottom figure shows the accuracy (ACC) as its training progresses. It can be seen from it that the model convergences are not very ideal, especially after 150 epochs, and on the validation sets, the recognition rates do not achieve the desired accuracy as the training progresses

Fig. 2 The schematic diagram of the LSTM-based feature extraction process with a single-step temporal information as its basic input cell



will surpass a specific threshold, is easier than predicting actual stock prices [3].

Stock movement prediction (SMP) is essentially a classification problem based on time series. The stock market is a complex system that involves multiple factors and influences, resulting in many difficulties, such as the significant unpredictability in the market, the emotional impact of the external economic environment, and the impact of economic policies. In this work, our focus lies in forecasting binary time-series movements, enabling investors to generate profits by making accurate long or short positions in a financial asset.

In recent years, the prediction of stock movement has mainly focused on the latest advanced models based on deep learning techniques [4–6], as [3, 7–10] for examples, which are data-driven and do not require professional knowledge in stock finance. Some studies have shown that deep learning-based models outperform traditional statistical models on SMP task. For example, the lecture [7] proposed the CMI autoregressive method, which used the idea of comparative learning to represent and learn stock time-series data by deep neural networks for price movement prediction. The Adv-ALSTM in [3] proposed adversarial sample learning [11] and combined it with the attention [12] LSTM to model stock sequence data and predict movements.

However, the high randomness and complexity of the stock market, as well as the characteristics of noisy market information and temporally dependent prediction, make the prediction a challenging problem [13]. Firstly, there is market uncertainty, and the stock movements are influenced by different types of factors, including economic, political, and social factors. The references [13–15] utilized the social media information to enhance the stock prediction, which requires manual collection of stock price time-series data. However, in reality, this additional information is often difficult to obtain; secondly, deep learning-based prediction methods often lack available training data [7]. The works in [3] and [7] addressed this issue by using adversarial samples and comparative learning, respectively. In addition, from the perspective of model structure, SMP is often related to the time information of stock prices. The existing literature usually uses LSTM-based methods [3, 7, 16, 17] for modeling, ignoring the complex hierarchical structure of non-stationary sequences in the temporal dimension. The statistics of single-step data show significant changes over time. Failure to address this non-stationarity explicitly at either the model architecture or loss function level makes model training challenging and results in suboptimal prediction accuracy.

The LSTM-based models LSTM and Adv-ALSTM [3] training trajectories are as shown in Fig. 1 (see the experimental section for the dataset and its detailed process). It can be found that as the training progresses, the model convergences are not very ideal, especially after 150 epochs, the minimum losses of the models on the training set are greater than 0.85, while on the validation set, the recognition rates do not achieve the desired accuracy with the training progresses with the 50% ACC. The existing models do not take into account the randomness and complex hierarchy of the temporal dimension in their modeling, which leads to weak feature representation ability in model extraction. As shown in Fig. 2, the existing LSTM-based time dimension models input the single-step temporal information of the stock time-series data into the LSTM cells. However, the semantics of the single-step temporal information are not as rich as the word vectors in NLP [19], and the features obtained by these methods have strong locality in other words. Then, we need to address these hierarchical features in the time-series data related to stocks.

In summary, the SMP problem is complex and with strong randomness and hierarchical temporal dimensions. A good SMP model should obtain this hierarchical information and be robust to the randomness in the stock data.

In order to tackle these concerns, a SMP framework based on adversarial training learning and hierarchy LSTM networks, abbreviated as Adv-SHNets is, proposed. Our primary consideration is that the semantic structure contained in stock time-series data is hierarchical, and the contribution of each level's structure to its movement classification is different. To implement this idea, Adv-SHNets, proposed in this work, firstly obtains the original features of stock information and calculates its covariates, including the time and representation vector of each stock, which are beneficial for obtaining the characteristics of each stock itself. Secondly, the original features are segmented in a temporal dimension and input to the low-level LSTM temporal model with a segmentation attention mechanism for low-level feature fusion. To ensure the semantic continuity of low-level features, the segmentation of time dimensions has overlap. Thirdly, the fused low-level features are input into the high-level LSTM to obtain hierarchical semantic information. And finally, the feature embedding vector calculates the training loss by the Hinge loss function. At the same time, in order to make Adv-SHNets robust to the uncertainty of stock data, we train the adversarial samples by simply and efficiently calculating the adversarial loss. The main contributions can be outlined as follows:

- **Novel end-to-end SMP framework:** A novel SMP framework named Adv-SHNets is proposed, which relies solely on stock numerical data, enhanced with preprocessing such as differencing to remove trends and address non-stationarity. This ensures the solution's practical applicability across diverse financial datasets.
- **Innovative hierarchical LSTM design:** We introduce overlapping temporal segmentation and a multi-level LSTM. This is the first SMP to effectively capture hierarchical dependencies. Preprocessing is used to handle non-stationarity, which significantly improves feature representation and model convergence.
- **Efficient adversarial loss function:** We have designed a simple adversarial loss function to strengthen the robustness of the model against randomness and non-stationary shifts in stock data. This enhances generalization without the need for additional data.
- **Empirical validation and performance gains:** When evaluated on the non-stationary datasets ACL18 and KDD17, Adv-SHNets achieved a relative accuracy improvement of 12.97% over state-of-the-art methods, as well as absolute gains of between 1.2% and 4.7%. This demonstrates its effectiveness in addressing non-stationarity and market complexities.

In addition, the structure of our paper is as follows: Section 2 gives a summary of related works on SMP approaches. Section 3 provides details about the Adv-SHNets framework. The evaluation of Adv-SHNets' predictive performance, encompassing prediction results and perturbation, is discussed in Sect. 4. Finally, Sect. 5 shows the conclusion and future work.

2 Related work

SMP is a fundamental application in financial data mining, garnering increased attention within the data mining and machine learning communities. This heightened interest is attributed to its significant influence on financial markets [20–22]. Considerable advancements have been made in predicting stock movement, particularly in handling complex nonlinear data, through extensive research, particularly in the realm of deep learning. The key developments pertinent to this paper are outlined as follows:

2.1 Basic categories of The SMP

Depending on the type of input data, the SMP can be classified into two main categories [21]: a) approaches that solely rely on past stock prices for forecasting and b) approaches incorporating supplementary text data for prediction.

The methods using only historical prices attempt to obtain patterns solely from historical data with lower data requirements. For example, the method Adv-ALSTM predicted the movement only by raw stock time-series data using adversarial training. AttLSTM [23] introduces attention mechanism into recurrent neural networks (RNNs) and demonstrated outstanding performance in the task of sequential predictions. Constrained deep neural network forecasting algorithm [24] focuses on a constrained deep neural network approach for accurate short-term stock price index movement prediction. Advanced deep learning for short-term price forecasting [25] presents an advanced deep learning model designed for short-term forecasting of US natural gas prices. The methods in [26–28] used the transfer learning strategy to inference the movement from the stock price. However, these methods lack hierarchical modeling for the historical prices. The proposed Adv-SHNets consider stock data as a time series and focus on improving predictive ability by the hierarchical structure of stock temporal dimensions, and it belongs to this category of methods.

While methods that incorporate additional textual data operate on the premise that stock movements are influenced by public sentiment, they typically integrate text feature extraction models with time-series techniques, thereby significantly enhancing prediction accuracy [29–33]. For example, SLOT [33] improves the accuracy of SMP based on the self-supervised learning strategy to get the embedding vectors of tweets and stocks. In StockNets [13], a deep generative model combining text content and price signal is proposed. However, such methods often require collecting lots of textual information, which makes their practical application more difficult. The focus of Adv-SHNets' consideration is to use the time-series structure of the stock itself for movement prediction; therefore, it does not belong to this type.

2.2 LSTM models in SMP

The stock data inherently constitute a time-sequential series, displaying evident temporal dependence. Recently, RNN has predominantly found applications in handling sequential data types, including time series, speech, images, text, and other data [34]. The long short-term memory (LSTM) network is an important RNN model structure that can overcome the long-term dependency problem of the original RNN [35] and is therefore widely used in stock data mining. In [36], trade indicators based on market microstructures were employed as input for an RNN, utilizing Graves' LSTM, to conduct price predictions for algorithmic stock trading. In the literature [23], the Attention LSTM (AttLSTM) model was pitted against the LSTM model in predicting the movement of Hong Kong stocks. The findings further validate the efficacy of the attention mechanism in LSTM-based prediction methods. A novel hybrid model, SA-DLSTM, introduced in [10], was devised for predicting the stock market and simulating trading. It combines an emotion-enhanced convolutional neural network, denoising autoencoder models and LSTM. Sonani et al. [37] developed a hybrid LSTM-GNN model for predicting stock prices, combining time-series analysis with graph-based methods. Using LSTM to capture temporal dynamics and GNN to model nonlinear stock relationships, the model outperformed benchmarks. Pilla et al. [38] used LSTM to forecast the S&P 500, comparing it with ARIMA using historical data to highlight its strength in handling financial volatility.

However, the basic unit processed by these LSTM-based methods is a single temporal step, and they have not considered the problem of insufficient semantic information of a single time in the stock temporal series and its temporal hierarchical nature, which can be seen from the basic LSTM Eq. $h_t = LSTM(x_t, h_{t-1})$, in which, h_t is the hidden representation, x_t is a single-time step data and $LSTM(.)$ is the arithmetic unit of the LSTM.

2.3 Adversarial learning in SMP

The randomness of stock data makes deep learning-based methods be difficult to obtain sufficient samples to improve the generalization ability of the model. Therefore, it is necessary to use adversarial methods that introduce noise to regularize parameters. The adversarial learning is by adding a disturbance Δx to the original input sample x and obtaining the adversarial sample, which is then used for training. And these processes can be abstracted into $\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} [\max_{\Delta x \in \Omega} L(x + \Delta x, y; \theta)]$ [39], where y is the ground truth of sample x and θ is the parameters of the model. The adversarial samples in Adv-ALSTM [3] were by $\boldsymbol{e}_{adv} = \boldsymbol{e} + \boldsymbol{r}_{adv}, \boldsymbol{r}_{adv} = \arg\max_{\boldsymbol{r}, \|\boldsymbol{r}\| \leq \epsilon} l\left(y, \hat{y}_{adv}\right)$, where e was the final latent representation of the stock, $\boldsymbol{r}_{adv}$ was the associated adversarial perturbation, and ϵ was a hyperparameter to explicitly control the scale of perturbation. STLAT [27] used $\Gamma^{\text{adv}} = \text{KL}\left(Y^{\text{adv}}, Y^s\right) = \text{KL}\left[p\left(\cdot \mid e^d; \hat{\theta}\right) \| p\left(\cdot \mid e^d + r^d_{\text{adv}}; \hat{\theta}\right)\right]$ to generate the adversarial samples. Deshpande et al. [40] proposed the use of quantum GANs for predicting stock index prices. This approach uses an adversarial generator-discriminator enhanced by quantum computing to generate synthetic data. Sun et al. [41] referenced ALSTM as a baseline and integrated adversarial training with attention LSTM to model stock time series, thereby improving prediction accuracy.

In summary, current adversarial approaches to predicting stock movements, such as the Adv-ALSTM, STLAT and attention-boosted LSTM models, struggle with the randomness and layered temporal structures inherent in stock data. These techniques often rely on complex perturbation strategies that hinder training, overlook deeper temporal layers, and result in unstable convergence and poor performance in volatile markets. Many also rely on additional data or complex setups, which limits their real-world applicability.

By contrast, our Adv-SHNets framework addresses these issues with an innovative hierarchical LSTM setup. It divides time series into overlapping segments to ensure continuity, extracts features at basic and advanced levels to improve distinction, and incorporates a straightforward adversarial training method to enhance resilience—all without additional complexity. The model handles everything end-to-end using only core stock numbers. This represents a significant advance in understanding the complex timing of stocks and delivers robust results across various scenarios.

3 Methods

3.1 Problem statement

We aim to improve the accuracy of SMP by capturing the hierarchical temporal dimensions within the stock data. And all the vectors of this paper are in column form if not otherwise specified.

SMP involves solving a binary classification problem. Following the literatures [13, 30], the formalization of the stock movement is defined according to the difference in price $s \in S = \{s_0, \cdots, s_{|S|}\}$ between two consecutive days d and $d-1$. Essentially given the sequential feature of the stock with T history trading days $X^s = [x^s_0, \cdots, x^s_{T-1}] \in \mathbb{R}^{T\times D}$, which is a feature matrix representing the sequential input features (e.g., open and close prices), and then, the SMP is defined as the following formula:

$$y^s = f_\Theta(X^s, V^s) = \begin{cases} -1 & p^s_d > p^s_{d+1}, \\ 1 & p^s_d < p^s_{d+1}, \end{cases} \tag{1}$$

which denotes the stock movement at the trading time point $T+h$, p^s_d represents the *adjusted closing price* of the stock s [3, 30, 42], $x^s_i \in \mathbb{R}^D$ and D is the variable dimension, V^s is the external covariates, such as trading time points, stock codes, and f_Θ is the prediction model function with the parameters Θ.

3.2 Overview of Adv-SHNets

As shown in Fig. 3, the proposed Adv-SHNets are structured with three main layers, namely the feature representation layer, the feature extraction layer, and the adversarial training layer. Especially, the feature representation layer obtains and fuses the basic features by Inception [43] 1-dimensional convolutional structure (Incept 1DCNN), and fuses the time and stock covariates of the stock. The feature extraction layer segments the stock time-series data along the time dimension and inputs these segmentation data into the different LSTM levels to extract the hierarchical temporal dependency information of the stock data. And the adversarial layer trains the whole model by the adversarial learning methods and label data.

From the process of Adv-SHNets shown in Fig. 3, it can be observed that Adv-SHNets does not require additional data other than temporal information, and the entire framework is trained end-to-end. The rest of this section provides a detail introduction of the essential components of Adv-SHNets.

3.3 Feature representation layer

The presentation layer in Adv-SHNets maps the original stock data into the feature latent space by the fusion and encoding methods, including the trading weekly encoder, stock encoder, and price feature encoder.

3.3.1 Trading weekly encoder

The stock trading day is usually from Monday to Friday every week. On Saturdays, Sundays, and statutory holidays, the stock market is closed. The input of the model f_Θ is the trading day window sequence data, which cannot reflect the stock trading day. Therefore, we need to encode the time information of the trading window.

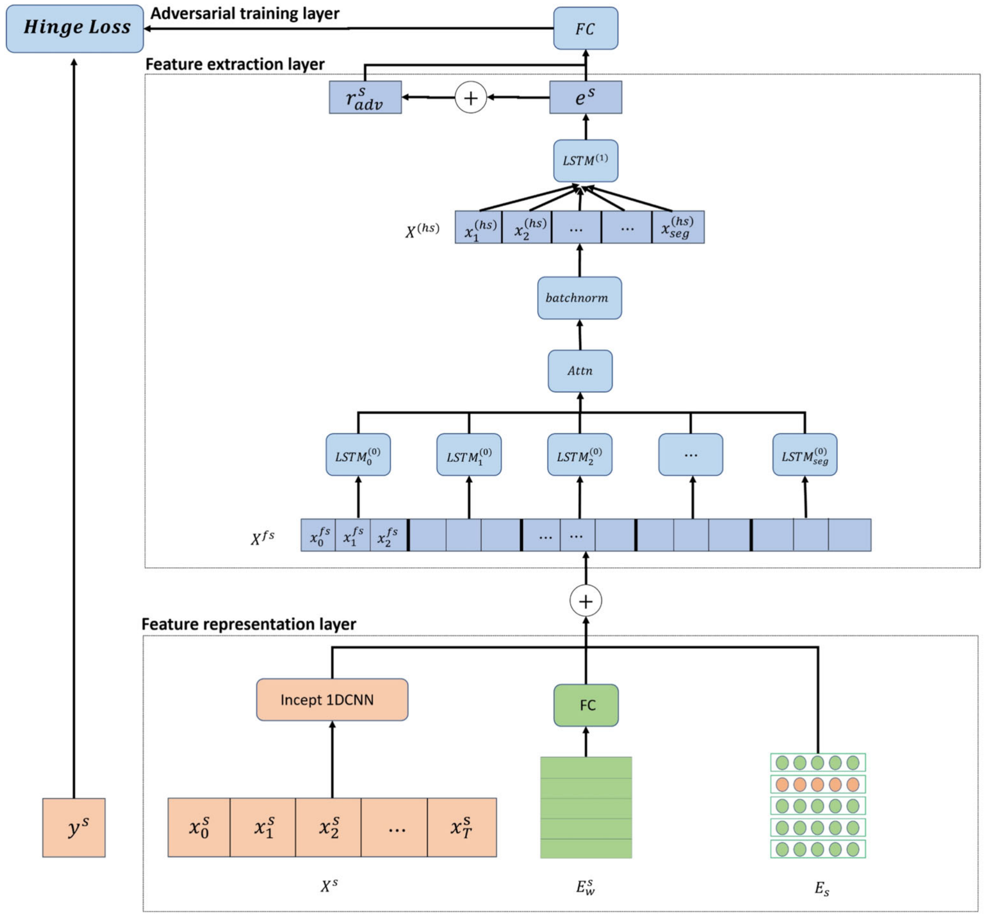


Fig. 3 The overview of the proposed Adv-SHNets Framework

We use the one-hot encoding $e^s_w = \{0, \cdots, 1, \cdots, 0\} \in \mathbb{R}^5\}$ to represent the weekly information of X^s transactions, and then, a neural network is used to map week codes to weekly embedding vectors E^s_w, i.e.,

$$E^s_w = \sigma(FC(e^s_w)) \in \mathbb{R}^{T \times D}, \tag{2}$$

where $FC(.)$ is a fully connected neural network, $\sigma = \tanh$ is the active function, and D is the dimension of the latent feature space. The weekly embedding vectors E^s_w are beneficial for the model to obtain weekly information on the prediction window.

3.3.2 Stock encoder

There are various types of stocks. As exemplified in [13, 44], the data are classified into 9 sectors: Basic Materials (BM), Consumer Goods (CG), Healthcare (HE), Services (SE), Utilities (UT), Conglomerates (CO), Financial (FI), Industrial Goods (IG), and Technology (TE). Different types of stocks have different patterns, and further, the patterns of changes in different stocks of the same type are also different. There are various types of stocks, such as finance, energy, real estate, etc. Different types of stocks have different patterns, and further, the patterns of changes in different stocks of the same type are also different. Based on this consideration, we believe that the model should be able to identify different stocks. Adv-SHNet uses the method of stock encoding to learn the dictionary embedding representation of stock representations by its training. Let $S = \{s_0, \cdots, s_{|S|-1}\}$ learnable dictionary embeddings be represented as $E_S \in \mathbb{R}^{|S| \times D}$, and then, we can get the encoding of the stock s, namely

$$E^s = E_S[s_i] \in \mathbb{R}^D, \tag{3}$$

where D is the same with it in Eq. (1), meaning the feature latent space dimensions.

3.3.3 Price feature encoder

The stock price data essentially constitutes a time series of trading days, encompassing information such as the *open, high, low, and closing* prices, which respectively represent the opening, highest, lowest, and closing prices for each day.

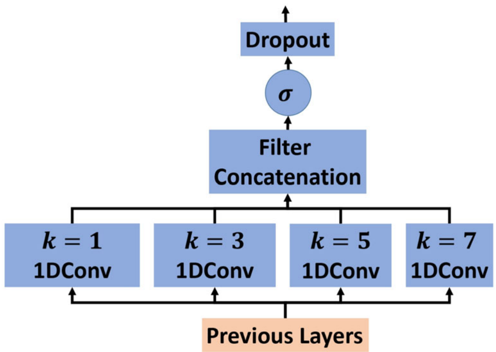


Fig. 4 The flowchart of the Inception structure used in Adv-SHNets, which can capture different features in stock data through different perspectives

Following [3, 33], we get the three price feature groups $X^s \in \mathbb{R}^{T\times 11}$ from the original to describe the daily trend of a stock $s \in S$, including price movements in a day, price movements between days and long-term movements (detailed in Table 1).

To fuse the price time information and maintain the temporal order of the sequence, 1D convolution (1DConv) is adopted to fuse the information of X^s in Adv-SHNets. And considering the temporal complexity of stock time-series data, we refer to the Inception structure [43] and use convolutional operators with different kernel sizes to fuse its temporal information. The other layer is followed by a tanh activation function and a dropout function to increase the generalization ability. The flow chart of the Inception structure-based operators is shown in Fig. 4, and we can use the following formula to describe this process:

$$X^{sm} = dropout(\sigma(\underset{k\in\{1,3,5,7\}}{concat}\ 1DConv_k(X^s))) \in \mathbb{R}^{T\times D}, \tag{4}$$

where $concat$ represents channel connection operation, $1DConv_k$ is the 1D convolution with kernel size k, the input channels $c_{in} = 11$, the output channels $c_{out} = \frac{D}{4}$, and $dropout(.)$ is used to prevent the overfitting. From Eq. (4) and Fig. 4, it can be seen that the practicality of the Inception structure enables Adv-SHNets to fuse different features in stock data through different perspectives.

The price feature encoder maps the stock price time-series data into the latent feature space with dimension D. The fusion of Eqs. (1), (3), and (4) will get the latent space features of the feature presentation layer, that is,

$$X^{fs} = E^s_w + E^s + X^{sm} \in \mathbb{R}^{T\times D}. \tag{5}$$

3.4 Feature extraction layer

The feature extraction layer extracts stock features based on the semantics and structure of the stock data, thereby obtaining the final semantic embedding of the stocks. Stock data belong to the category of time-series data, and employing time-dependent modeling is crucial for accurately predicting its movement. The hierarchy LSTM-based methods are employed to extract their hierarchical information in our proposed framework.

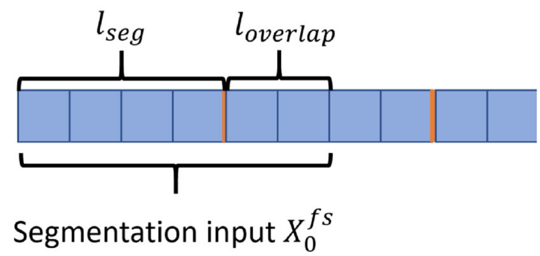


Fig. 5 The diagram of the overlapping segmentation method

3.4.1 Time segmentation LSTM

Considering the problem of insufficient information in a single time step, we have performed overlapping segmentation along the time dimension of X^{fs} and input each segmentation into the corresponding LSTM to extract the low-level time-dependent features as shown in Fig. 3. The data segmentation method is shown in Fig. 5 and can be formulated as follows:

$$X^{fs}_k = X^{fs}[kl_{seg} : (k+1)l_{seg} + l_{overlap}], k = 0, \cdots, \lceil\frac{T}{l_{seg}}\rceil, \tag{6}$$

where l_{seg} is the segmentation length, $l_{overlap}$ is the overlapping length, $\lceil . \rceil$ represents rounding up, and in the case of nondivisibility, we perform zero padding on the original sequence. And then, we input X^{fs}_k to a LSTM model, i.e.,

$$X^{ls}_k = LSTM^{(0)}_k(X^{fs}_k) \in \mathbb{R}^{(l_{seg}+l_{overlap})\times D}. \tag{7}$$

To fuse the segmentation time information within a segment, X^{ls}_k is input to a 1D convolution with a kernel size of $k = l_{seg} + l_{overlap}$, i.e., $X^{cs}_k = 1DConv_{l_{seg}+l_{overlap}}(X^{ls}_k) \in \mathbb{R}^{1\times D}$. Finally, we concatenate the segmentation features to get the first-level features, namely

$$X^{ls} = \underset{k=0,\cdots,\lceil\frac{T}{l_{seg}}\rceil}{concat} X^{cs}_k \in \mathbb{R}^{\frac{T}{l_{seg}}\times D}. \tag{8}$$

It can be seen from Eq. (8) that X^{ls} still maintains the original temporal order, and due to overlapping, it has a certain degree of temporal continuity.

Table 1 The characteristics of a stock's daily trend

Feature groups	Features	Calculation
Price movements in a day	c_open, c_high, c_low	$e.g., c\_open = open_t/close_t - 1$
Price movements between days	n_close, n_adj_close	$e.g., n\_close = close_t/close_{t-1} - 1$
Long-term movements	$5\_day, 10\_day, 15\_day,$ $20\_day, 25\_day, 30\_day$	$e.g., 5\_day = \frac{\sum_{i=0}^{4} adj\_close_{t-i}/5}{adj_c close_t} - 1$

3.4.2 The segmentation attention

Different time segmentations have different contributions to the stock movement classification. Based on this idea, the attention mechanisms are used in our proposed framework Adv-SHNets, along the segmentation dimension of X^{ls}. The segmentation attention mechanism fuses information by learning different weight coefficients from the fragments' representation in the latent space, which can be formulated as:

$$Q = W_q X^{ls}, \tag{9}$$

$$K = W_k X^{ls}, \tag{10}$$

$$V = W_v X^{ls}, \tag{11}$$

$$H_h = V \text{ softmax}\left(\frac{K^T Q}{\sqrt{D}}\right), \tag{12}$$

where W_q, W_k, and W_v are the learning parameters. To integrate multiple attention relationships, Adv-SHNets utilizes a multi-head attention mechanism, namely

$$X^{as} = \text{batchnorm}\left(\underset{h}{concat}(W_h H_h)\right), \tag{13}$$

where W_h is the learning parameter of the kth head of the attention's output H_k and batchnorm(.) [45] is the batch norm operator, which plays a role in regularizing the distribution of output and facilitating the convergence of training.

In summary, the hierarchical LSTM approach effectively addresses the limitations of flat LSTMs by segmenting data into overlapping low-level units and fusing them at a higher level. Flat LSTMs struggle to capture long-term hierarchies, which often results in localized features and poor generalization. In contrast, the hierarchical structure improves the representation of non-stationary patterns, leading to faster convergence, as demonstrated by experimental results.

3.5 The high-level LSTM

In order to obtain high-level temporal semantic information, we perform high-level LSTM feature extraction after the segmenting LSTM layer to obtain the final stock representation embedding e^s, namely

$$e^s = LSTM^{(1)}(X^{as}) \in \mathbb{R}^D. \tag{14}$$

It can be seen from Eqs. (8) and (14) that the input of $LSTM^{(1)}$ is essentially semantic information after segmentation, which still has temporal order and overcomes the problem of semantic inadequacy in time steps.

With e^s, a fully connected operator is used to estimate the classification confidence as a prediction function:

$$\hat{y^s} = FC(e^s) \in [-1, 1]. \tag{15}$$

And one should note that the final prediction is $sign(\hat{y^s})$.

3.6 Adversarial training

Similar to the other binary learning tasks, Adv-SHNets uses the Hinge function [18] as the loss function for clean training samples, that is, the objective function of movement of Adv-SHNets is

$$Loss_n = \sum_s h(FC(e^s), l^s), \tag{16}$$

where $h(y^s, \hat{y^s}) = \max(0, 1 - y^s \hat{y^s})$ is the Hinge function, Fc is the fully connected operator which is the same as in Eq. (15) and l^s is the ground-truth label of the stock sample s.

Considering the randomness of stock data, we adopt adversarial training [46] as in reference [3, 27], where the training samples include clean and adversarial samples simultaneously. The adversarial training usually adds perturbations to the clean samples to learn a robust model. Follow the works in [3, 27] Adv-SHNets add the perturbations r^s_{adv} into the final stock representation embedding e^s in Eq. (14), and thus, the perturbations is $e^s_{adv} = e^s + r^s_{adv}$. Considering the simplicity of calculation, adversarial training adopts the following cost function:

$$Loss_{adv} = |\min(\max(FC(e^s + r^s_{adv}), -\gamma), \gamma)|, \tag{17}$$

where FC is the fully connected neural network as in Eq. (15) and γ is the hyperparameter. The meaning of Eq. (17) is that the predicted output range between $[-\gamma, \gamma]$ contributes to the loss. Finally, the perturbations of Adv-SHNets are obtained by the fast gradient approximation methods, namely

$$e^s_{adv} = e^s + \beta_{adv} r^s_{adv}, r^s_{adv} = \arg \max_{\|r^s_{adv}\| < 1} Loss_{adv}, \qquad (18)$$

where β_{adv} is a hyperparameter that adjusts the intensity of the perturbations sample.

The simple perturbation strategy used in adversarial training greatly enhances the model's resilience to the randomness inherent in stock data. This approach outperforms more complex methods, which can complicate training. This approach effectively mitigates uncertainty by integrating clean and adversarial losses, resulting in higher accuracy scores compared to non-adversarial baselines.

3.7 The whole loss of Adv-SHNets

The whole loss of Adv-SHNets is the combination of the clean sample and perturbation sample training loss, i.e.,

$$Loss = \sum_{s \in S} (h(FC(e^s), l^s) + \alpha h(FC(e^s_{adv}), l^s)), \qquad (19)$$

where α is a balance parameter.

3.8 The algorithm of Adv-SHNets

The training phase algorithm for Adv-SHNets is shown in Algorithm 1. In detail, the input of the proposed Adv-SHNets includes the stock set S, the date of stock data, the time-series original feature data of stock prices $X^s, s \in S$, and the true label of S. As shown in Algorithm 1, the process of model training requires a large number of calculations according to the previous equations.

Algorithm 1 The training phase of Adv-SHNets

Require: The stock set S; The date of stock data; The time series original feature data of stock prices X^s, $s \in S$; The true label of S
Ensure: the learned model
1: Calculate the trading week code, stock code, and stock value code respectively according to Eqs. (1), (3), and (4);
2: Calculate the latent space features according to Eq. (14);
3: Calculate the low-level temporal features using overlapping segmentation LSTM Method according to Eqs. (7) and (8);
4: Calculate the high-level temporal features to get the final stock representation embedding e^s according to Eq. (14);
5: Calculate the classification confidence by Eq. (15);
6: Calculate the perturbation sample according to Eq. (18);
7: Calculate the final training loss of Adv-SHNets according to Eq. (19).
8: Propagate the network backward according to loss L and update the model parameters.

4 Performance analysis

Here, we design a number of experiments to address the 3 research questions as follows:

- RQ1: How does the performance of our proposed methods, Adv-SHNets, compare to various state-of-the-art methods?
- RQ2: According to the discussion in the introduction of this paper, Adv-SHNets adopts a segmentation hierarchical structure that conforms to stock time-series data, making it easier to obtain features. How does it accelerate the training process?
- RQ3: What impact do different components have on the resilience of our proposed Adv-SHNets?

4.1 Experimental setup

4.1.1 Datasets

The experiments conducted in this paper employed the ACL18 [13], KDD17 [47], shangha_50 and shenzhen_50 datasets for evaluation. These datasets were chosen because they are publicly available and widely used in studies related to stock market applications [3, 18, 27, 28, 44]. They also represent diverse stock types and market conditions, which helps to validate the robustness of Adv-SHNets across varying economic environments.

The ACL18 data comprises historical data spanning from 2014-01-01 to 2016-01-01 for 88 high-trade-volume stocks from various sectors such as technology, finance, and consumer goods in the NASDAQ and NYSE markets, covering a period of relative post-crisis stability. Following [3], the ACL18 data are partitioned into training (2015-01-01), validation (2015-08-01 to 2015-10-01), and testing (2015-10-01 to 2017-01-01) sets along the temporal dimension. On the other hand, KDD17 includes a longer history from 2007-01-01 to 2016-01-01 for 50 high-volume stocks from similar US markets, encompassing volatile conditions such as the 2008 financial crisis and subsequent recovery phases, thus including bull, bear, and crisis market scenarios. It is split along the temporal dimension into training (2015-01-01 to 2015-01-01), validation (2015-01-01 to 2017-01-01), and testing (2016-01-01 to 2017-01-01) sets.

Meanwhile, the Shanghai_50 and Shenzhen_50 datasets comprise historical price data from 2019 to 2025 for 50 representative stocks listed on the Shanghai and Shenzhen stock exchanges in China, sourced from EastMoney. These datasets contain approximately 1,500 data points.

Following the works in [3, 27], we used the features listed in Table 1, and the negative or positive label is calculated according to Eq. (1).

4.1.2 Metrics

Consistent with prior studies on stock movement prediction [3, 27], we employ two standard metrics to assess the prediction performance as follows:

- Matthews Correlation Coefficient (MCC):

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fp)(tn + fp)(tn + fn)}},$$

- The Harmonic Mean of Precision and Recall (F1 Score):

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}},$$

- The Area Under the Receiver Operating Characteristic Curve (ROC_AUC):

$$\text{TPR} = \frac{\text{tp}}{\text{tp} + \text{fn}}, \text{FPR} = \frac{\text{fp}}{\text{fp} + \text{tn}},$$

- Accuracy (ACC):

$$Acc = \frac{tp + tn}{tp + fn + tn + fp},$$

where tp, tn, fp and fn represent the true positives, true negatives, false positives, and false negatives on test examples, respectively. Precision $= \frac{tp}{tp+fp}$, Recall $= \frac{tp}{tp+fn}$. The higher the values of MCC, F1 Score, ROC_AUC and ACC are, the better the prediction effect is.

4.1.3 Baselines

We benchmark Adv-SHNets against the following baseline methods for SMP.

- LSTM [48] is a very basic model for SMP. We selected it because Adv-SHNets is an LSTM-based model.
- ALSTM [20] is an LSTM-based model and uses the temporal attention mechanism, which is also used in our proposed methods.
- LSTM_GCN [49] is a model for stock price prediction using knowledge-incorporated graphs to capture stock relationships and capital flows.
- StockNet [13] used variational autoencoders to encode stock movement and the media text information as latent probabilistic vectors, which is a relatively new method.
- StockFormer [50] used wavelet transforms and self-attention to predict stock returns and trends with enhanced accuracy in volatile markets.
- Adv-ALSTM [3] trains ALSTM with adversarial data to enhance the robustness of SMP. It is chosen as a baseline due to the shared utilization of adversarial data in Adv-SHNets.
- IDTLA-2 S [44] is a recently proposed transfer learning model incorporating an attention mechanism, serving as another baseline method for comparison.

4.1.4 Other settings

We implement Adv-SHNets using PyTorch and optimize it with mini-batch Adam. The batch size and the initial learning rate are 1024 and 0.01, respectively, which follow the approach used in Adv-ALSTM [3]. In order to make the results fair and credible, all experiments are conducted during the training process on the validation set so that the model with the best results is saved and used for testing on the test set. The hyperparameter in Eq. (18) is $\beta_{adv} = 1.0$. The loss function hyperparameter in Eq. (19) is $\alpha = 0.1$. The latent dimension is $D = 16$ in ADV-SHNets. And the hyperparameter γ in Eq. (19) is 0.6. For the history length T and segmentation number $N_{seg} = \frac{T}{l_{seg}}$, overlapping length $l_{overlap}$ in Eq. (6) are selected via grid-search within the ranges of [10, 30], [2, 4] and [1, 3], respectively. We can find that the Adv-SHNets model is sensitive to these parameters, as shown in Fig. 6.

4.2 Prediction performance comparison RQ1

Table 2 presents a comparison of various methods based on Acc and MCC metrics, with bolder fonts indicating superior performance. It is evident from Table 2 that:

- Adv-SHNets obtains the best results on both datasets under all evaluation metrics, and compared to Adv-ALSTM testing on database ACL18 (KDD17), it improved the accuracy by 12.07% and 203.35% (5.97% and 307.35%), both of which were modeled by time-based LSTM and adversarial training. This fully proves the effectiveness of the method proposed in this article.
- Compared with the latest method IDTLA-2 S, Adv-SHNets is also competitive, with an accuracy improvement of 2.48% and 13.04% on the ACL18 database.
- ALSTM, LSTM, and Adv-SHNets models are all based on LSTM methods. However, in some cases, ALSTM accuracy does not have an advantage compared to LSTM, while Adv-SHNets has a significant improvement compared to those, indicating that attention needs to be obtained by semantics, which does not have an advantage for single-time step attention. This fully demonstrates that a single stock time step does not have sufficient information content.
- Adv-SHNets and Adv-ALSTM exhibit notable enhancements compared to StockNet, which incorporates the

Table 2 Performance comparison on the ACL18, KDD17, Shanghai_50 and Shenzhen_50 datasets. Bolded metrics indicate the best performance, while underlined metrics indicate the second-best performance

Methods	ACL18				kDD17			
	MCC	F1 score	ROC AUC	ACC	MCC	F1 score	ROC AUC	ACC
LSTM	0.0022	0.0212	0.5243	0.6550	0.1763	0.4155	0.6179	0.6050
ALSTM	0.0142	0.0299	0.5254	0.6554	0.1783	0.4325	0.6235	0.6057
LSTM_GCN	0.0635	0.2264	0.5354	0.6384	0.1851	0.4586	**0.6368**	**0.6170**
StockNet	0.0187	0.1045	0.5113	0.5283	0.0183	0.4513	0.5109	0.5193
StockFormer	0.0677	**0.4852**	**0.5485**	0.5339	0.0961	0.4165	0.4289	0.4537
Adv-ALSTM	0.0626	0.4486	0.5301	0.5304	0.0204	0.4131	0.3972	0.5094
IDTLA-2 S	0.1680	0.3059	0.4891	0.5800	0.1566	0.3841	0.5103	0.5455
Adv_SHNets	**0.1899**	0.4542	0.5262	**0.6587**	**0.1925**	**0.4603**	0.6319	0.6114

Methods	Shanghai_50				Shenzhen_50			
	MCC	F1 score	ROC AUC	ACC	MCC	F1 score	ROC AUC	ACC
LSTM	0.1016	0.5939	0.5641	0.5546	0.0928	0.6175	0.5302	0.5604
ALSTM	0.1017	0.5948	0.5625	0.5548	0.0972	0.6234	0.5406	0.5556
LSTM_GCN	0.1287	0.6221	0.5852	0.5701	0.1072	0.7103	0.5673	0.5749
StockNet	0.0130	0.3649	0.5047	0.5215	0.0850	0.4279	0.4973	0.5049
StockFormer	0.0584	0.5166	0.5420	0.5291	0.0570	0.4682	0.5295	0.5328
Adv-ALSTM	0.0518	0.4967	0.5364	0.5029	0.0532	0.4363	0.5123	0.4986
IDTLA-2 S	0.0626	0.4326	0.4262	0.4084	0.0269	0.3411	0.4577	0.4367
Adv_SHNets	**0.1301**	**0.6443**	**0.5926**	**0.5730**	**0.1169**	**0.7388**	**0.5922**	**0.5858**

stochastic nature of stock inputs with VAE and Adv-ALSTM. This improvement is attributed to the consideration of the inherent randomness in stock data by both methods.

In summary, the superior ACC and MCC of Adv-SHNets result from its hierarchical LSTM and adversarial training. These address the limitations of baseline models by capturing multi-scale dependencies and enhancing robustness. The hierarchical LSTM effectively handles non-stationarity, achieving faster loss reduction, and demonstrates significant performance drops when the high-level LSTM is removed. Adversarial training improves the MCC by perturbing the embeddings to counteract randomness; this is in contrast to non-adversarial methods. This synergy ensures consistent performance gains across datasets, thus validating the contributions of these components to improved convergence and generalization.

4.3 Hyperparameter sensitivity experiment RQ1

To evaluate the robustness of the model, we performed a sensitivity analysis on the key hyperparameters N_{seg} and overlap. We assessed their impact on ACC using the ACL18 dataset. This helps to identify optimal ranges and potential vulnerabilities.

As shown in Fig. 6, when T is fixed at 20 and overlap is set to 3, the accuracy (ACC) increases from 0.55 at $N_{seg} = 1$ (single-layer LSTM) to 0.58 at $N_{seg} = 2$, remains at 0.57 for $N_{seg} = 3$, and decreases to 0.54 at $N_{seg} = 4$. Moderate segmentation (2–3) captures hierarchical dependencies, whereas excessive segmentation (>3) fragments the data, disrupting continuity and upper LSTM fusion.

With T = 20 and $N_{seg} = 2$ fixed, the ACC increases from 0.53 at overlap = 0 (continuity loss) to 0.58 at Overlap = 3, with values of 0.55–0.57 for overlap = 1–2. Adequate Overlap ensures smooth segment transitions and stronger low-level features, whereas insufficient overlap reduces robustness.

Overall, Adv-SHNets is sensitive to temporal parameters, reflecting the complexity of stock data. However, it remains robust within grid-searched optima. Performance gradually deteriorates beyond this, which can inform deployment via prioritized temporal tuning on non-stationary data. Referring to Fig. 6 provides the detailed sensitivity analysis of hyperparameters N_{seg} and overlap.

4.4 Training trajectories of Adv-SHNets RQ2

This section will verify the impact of time hierarchical structure on the model training process, by comparing LSTM and Adv-LSTM models based on the same LSTM structure. The loss of the models on the training set and the accuracy on the validation set as the training progresses are shown in Fig. 7

From Fig. 7, we can see that as the training progresses, all models converge and their accuracy on the validation set gradually improves. However, the training losses of LSTM

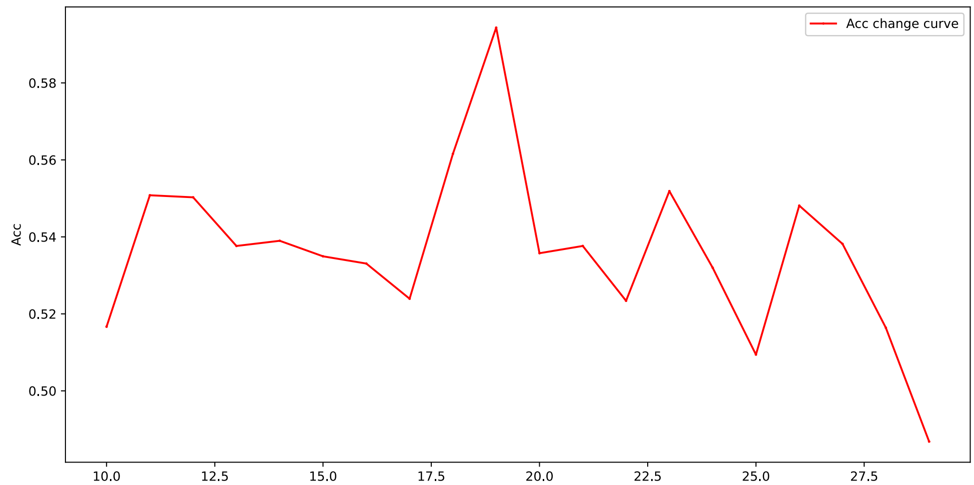


Fig. 6 When we fix $N_{seg} = 2$ and $Overlap = 3$, the change of Acc with history trade window length on ACC18 dataset

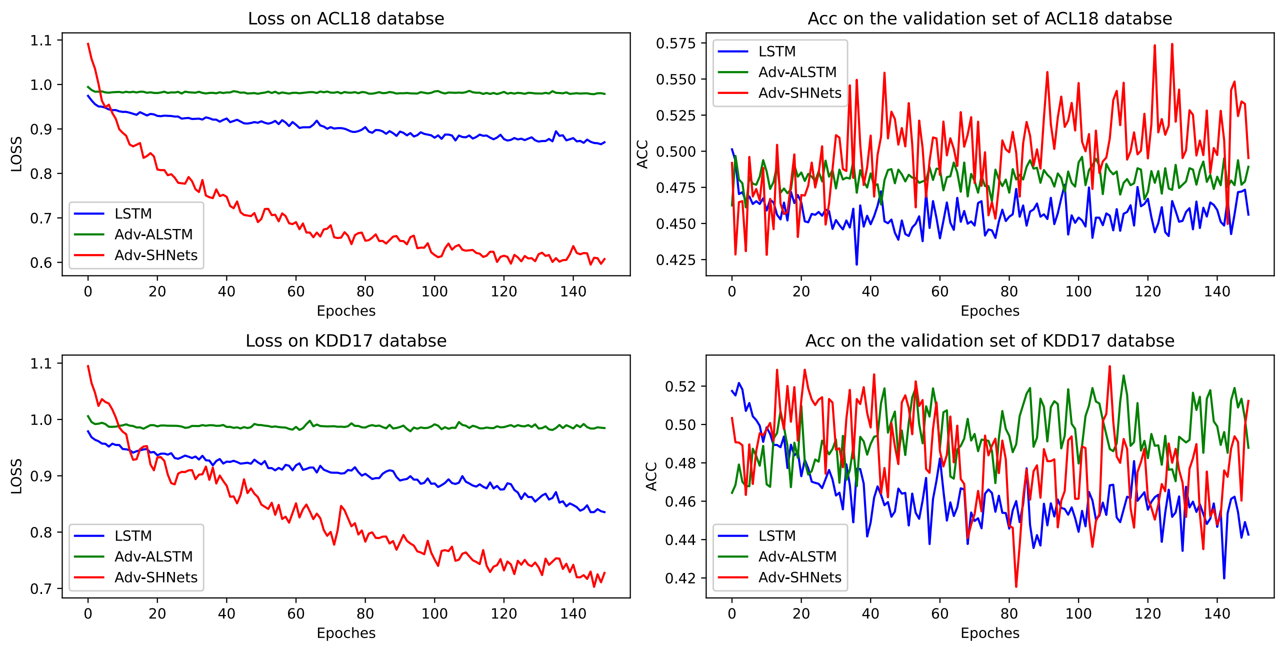


Fig. 7 The training trajectories of LSTM [20], Adv-ALSTM [3], and Adv-SHNets on the dataset ACL18 and KDD17. The left two figures show the training Hinge Loss [18] on the training set, while the right two figures show the accuracy on the validation set as training progresses

and Adv-ALSTM models on the training set no longer continue to decrease when they decrease to 0.89−0.85. This situation has been greatly improved in Adv-SHNets, where training losses have decreased throughout the training process, from 1.1 to 0.73. We believe this is because Adv-SHNets can more fully obtain information from stock data, especially the hierarchical structure information in stock data.

4.5 Ablation experiment RQ3

Ablation experiments are conducted in this section to assess the impact of each component in the Adv-SHNets framework through five variants:

- wo overlap: Adv-SHNets without overlap in Eq. (6), which means that the temporal correlation of features obtained through the underlying LSTM is weakened;
- wo HLSTM: Adv-SHNets without high-level LSTM, which means that we change Eq. (14) to $e^s = FC(X^{as})$;
- wo Adv: Adv-SHNets without the adversarial training;
- wo attn: Ad-SHNets without the segmentation attention mechanism, which means that we change Eq. (13) to $X^{as} = \text{batchnorm}(concat_k\, W_k X^{ls})$.

In this segment of the experiment, we evaluate it on the ACL18 and KDD17 databases, maintaining the same experimental parameters as detailed in subsection 4.2. The experimental results are shown in Fig. 8. It can be seen from

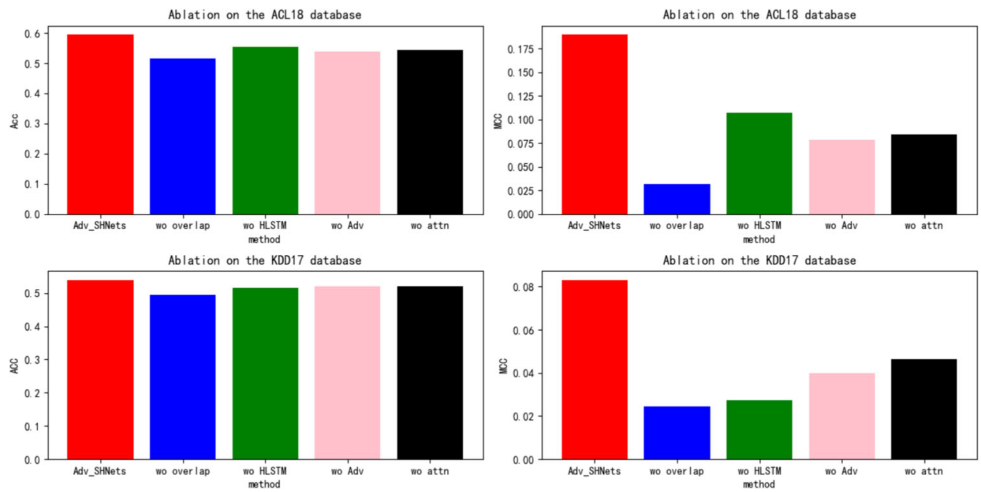


Fig. 8 Recognition accuracy of different methods on the dataset ACL18 and KDD17

Fig. 8 that adv-SHNets outperform wo overlap, wo HLSTM, wo Adv, and wo attn in most cases, which indicates the effectiveness of these components of our proposed methods, and upon removing the overlap in the segmentation LSTM, there is a significant decline in the model's performance. This underscores the importance of preserving the continuity of low-level LSTM temporal features; when the segmentation attention mechanism is removed, the adv-SHNets obviously decrease, which fully illustrates the imbalance of low-level temporal feature information; and when both adversarial training and the high-level LSTM are omitted, there is a noticeable decrease in performance. This substantiates that our adversarial training and the design of multi-layer LSTM are indeed robust to stock noise, contributing to enhanced accuracy in SMP.

5 Conclusion and future work

This paper introduces Adv-SHNets, a novel, deep learning-based approach for predicting stock movements. The proposed method effectively processes temporal information using a hierarchical LSTM and overlapping segmentation, while incorporating adversarial learning techniques. The hierarchical LSTM excels at capturing multi-scale dependencies and addressing semantic limitations and non-stationarity, resulting in faster convergence and higher accuracy. Adversarial training enhances robustness to randomness, as demonstrated by ablation studies and training trajectories. This enables consistent outperformance on real-world datasets without the need for additional data. Adv-SHNets can solve the problem of previous SMPs neglecting insufficient information in a single step when extracting features and can fully extract time hierarchy information.

To improve the effectiveness of the proposed method, we will enhance its adaptability to more complex stock market prediction scenarios by incorporating multiple data sources and refining the model architecture. Specifically, we intend to integrate textual information, such as financial news and social media sentiment, using existing SMP methods to improve the accuracy of stock movement predictions. Additionally, we plan to explore advanced attention mechanisms to better capture temporal and contextual dependencies in volatile market conditions. Another approach is to develop hybrid models combining hierarchical LSTMs and graph neural networks to model inter-stock relationships and market dynamics more effectively. Finally, we will investigate applying reinforcement learning techniques to enable the model to dynamically adjust its decision-making process, thus optimizing predictions in real-time trading environments.

Acknowledgements This research received partial support from the National Key Research & Development Plan Project of China (2021YFC2600502), the Key Research & Development Transformation Plan of Qinghai Province (2022-QY-218), and the National Natural Science Foundation of China (62272311).

Author Contributions Q.Z. was mainly responsible for data collating and writing. N.W. and Y.Z. got the data and the results verified. W.Y. and H.W. mainly modified and polished the article. K.L., S.Y. and L.W. primarily designed the thesis program.

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

Ethical Approval This article does not contain any studies with animals performed by any of the authors.

References

1. Walczak, S.: An empirical analysis of data requirements for financial forecasting with neural networks. J. Manag. Inf. Syst. **4**, 17 (2001)
2. Thien, H.: Nguyen, Kiyoaki, Shirai, Julien, Velcin: Sentiment analysis on social media for stock movement prediction. Expert Syst. Appl. **42**(24), 9603–9611 (2015)
3. Feng, F., Chen, H., He, X., Ding, J., Chua, T.S.: Enhancing stock movement prediction with adversarial training. In: Twenty-eighth international joint conference on artificial intelligence IJCAI-19 (2019)
4. Tian, L., Li, X., Ye, Y., Xie, P., Li, Y.: A generative adversarial gated recurrent unit model for precipitation nowcasting. IEEE Geosci. Remote Sens. Lett. **17**(4), 601–605 (2019)
5. Li, T., Liu, W., Zeng, Z., Xiong, N.N.: Drlr: A deep-reinforcement-learning-based recruitment scheme for massive data collections in 6g-based iot networks. IEEE Internet Things J. **9**(16), 14595–14609 (2021)
6. Sun, C., Zhang, C., Xiong, N.: Infrared and visible image fusion techniques based on deep learning: a review. Electronics **9**(12), 2162 (2020)
7. Wu, H., Gattami, A., Flierl, M.: Conditional mutual information-based contrastive loss for financial time series forecasting. arXiv e-prints (2020)
8. Sawhney, R., Agarwal, S., Wadhwa, A., Shah, R.R.: Deep attentive learning for stock movement prediction from social media text and company correlations. In: Empirical methods in natural language processing (2020)
9. Envelope, Y.M.A., Envelope, R.M.B., Envelope, Q.L.C., Envelope, P.W.D., Envelope, E.C.B.P.: Multi-source aggregated classification for stock price movement prediction. Information Fusion **91**, 515–528 (2022)
10. Zhao, Y., Yang, G.: Deep learning-based integrated framework for stock price movement prediction. Appl. Soft Comput. **133**, 109921 (2023)
11. Cheng, J., Yang, Y., Tang, X., Xiong, N., Zhang, Y., Lei, F.: Generative adversarial networks: a literature review. KSII Trans. Internet Inf. Syst. **14**(12), 4625–4647 (2020)
12. Shen, Y., Fang, Z., Gao, Y., Xiong, N., Zhong, C., Tang, X.: Coronary arteries segmentation based on 3d fcn with attention gate and level set function. Ieee Access **7**, 42826–42835 (2019)
13. Xu, Y., Cohen, S.B.: Stock movement prediction from tweets and historical prices. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1970–1979 (2018)
14. Ma, Y., Mao, R., Lin, Q., Wu, P., Cambria, E.: Multi-source aggregated classification for stock price movement prediction. Inf. Fusion **91**, 515–528 (2023)
15. Das, N., Sadhukhan, B., Chatterjee, T., Chakrabarti, S.: Effect of public sentiment on stock market movement prediction during the covid-19 outbreak. Soc. Netw. Anal. Min. **12**(1), 92 (2022)
16. Zhang, Y., Tan, H., Yang, J., Kim, T., Bae, J.: Stock price movement prediction based on re-extract feature lstm. ICIC Express Letters ICIC International, pp. 187â **194** (2022)
17. Choi, J., Yoo, S., Zhou, X., Kim, Y.: Hybrid information mixing module for stock movement prediction. IEEE Access **11**, 28781–28790 (2023)
18. Gentile, C., Warmuth, M.K.: Linear hinge loss and average margin. In: Advances in neural information processing systems **11** (1998)
19. Yang, B., Bender, G., Le, Q.V., Ngiam, J.: Condconv: conditionally parameterized convolutions for efficient inference. Advances in neural information processing systems. **32**, (2019)
20. Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., Cottrell, G.: A dual-stage attention-based recurrent neural network for time series prediction. Preprint at arXiv:1704.02971 (2017)
21. Yoo, J., Soun, Y., Park, Y.-c., Kang, U.: Accurate multivariate stock movement prediction via data-axis transformer with multi-level contexts. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining, pp. 2037–2045 (2021)
22. Hagenau, M., Liebmann, M., Neumann, D.: Automated news reading: Stock price prediction based on financial news using context-capturing features. Decis. Support Syst. **55**(3), 685–697 (2013)
23. Chen, S., Ge, L.: Exploring the attention mechanism in lstm-based hong kong stock price movement prediction. Quant. Financ. **19**(9), 1507–1515 (2019)
24. Livieris, I.E., Kotsilieris, T., Stavroyiannis, S., Pintelas, P.: Forecasting stock price index movement using a constrained deep neural network training algorithm. Intell. Decis. Technol. **14**(3), 313–323 (2020)
25. Livieris, I.E., Pintelas, E., Kiriakidou, N., Stavroyiannis, S.: An advanced deep learning model for short-term forecasting us natural gas price and movement. In: IFIP international conference on artificial intelligence applications and innovations, pp. 165–176. Springer, Cham (2020)
26. Nguyen, T.-T., Yoon, S.: A novel approach to short-term stock price movement prediction using transfer learning. Appl. Sci. **9**(22), 4745 (2019)
27. Li, Y., Dai, H.-N., Zheng, Z.: Selective transfer learning with adversarial training for stock movement prediction. Connect. Sci. **34**(1), 492–510 (2022)
28. Pal, S.S., Kar, S.: Fuzzy transfer learning in time series forecasting for stock market prices. Soft. Comput. **26**(14), 6941–6952 (2022)
29. Kumar, D., Sarangi, P.K., Verma, R.: A systematic review of stock market prediction using machine learning and statistical techniques. Mater. Today Proceed. **49**, 3187–3191 (2022)
30. Sawhney, R., Agarwal, S., Wadhwa, A., Shah, R.: Deep attentive learning for stock movement prediction from social media text and company correlations. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp. 8415–8426 (2020)
31. Rajendiran, P., Priyadarsini, P.: Survival study on stock market prediction techniques using sentimental analysis. Mater. Today Proceed. **80**, 3229–3234 (2023)
32. Bouadjenek, M.R., Sanner, S., Wu, G.: A user-centric analysis of social media for stock market prediction. ACM Trans. Web **17**(2), 1–22 (2023)
33. Soun, Y., Yoo, J., Cho, M., Jeon, J., Kang, U.: Accurate stock movement prediction with self-supervised learning from sparse noisy tweets. In: 2022 IEEE international conference on big data (Big Data), pp. 1691–1700. IEEE (2022)
34. Ozbayoglu, A.M., Gudelek, M.U., Sezer, O.B.: Deep learning for financial applications: A survey. Appl. Soft Comput. **93**, 106384 (2020)
35. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
36. Karaoglu, S., Arpaci, U., Ayvaz, S., et al: A deep learning approach for optimization of systematic signal detection in financial trading systems with big data. Int. J. Intell. Syst. Appl. Eng. 2017(Special Issue), 31–36 (2017)
37. Sonani, M.S., Badii, A., Moin, A.: Stock price prediction using a hybrid lstm-gnn model: Integrating time-series and graph-based analysis. Preprint at arXiv:2502.15813 (2025)
38. Pilla, P., Mekonen, R.: Forecasting s&p 500 using lstm models. Preprint at arXiv:2501.17366 (2025)
39. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. Preprint at arXiv:1706.06083 (2017)

40. Deshpande, S., Dahale, G.R., Morapakula, S.N., Wad, U.: Prediction of stocks index price using quantum gans. Preprint at arXiv:2509.12286 (2025)
41. Sun, W., Mei, J., Liu, S., Yuan, C., Zhao, J.: Research on deep learning model for stock prediction by integrating frequency domain and time series features. Sci. Rep. **15**(1), 30386 (2025)
42. Yang, L., Ng, T.L.J., Smyth, B., Dong, R.: Html: Hierarchical transformer-based multi-task learning for volatility prediction. In: Proceedings of the web conference 2020, pp. 441–451 (2020)
43. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9 (2015)
44. He, Q.-Q., Siu, S.W.I., Si, Y.-W.: Instance-based deep transfer learning with attention for stock movement prediction. Appl. Intell. **53**(6), 6887–6908 (2023)
45. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp. 448–456. pmlr (2015)
46. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. Preprint at arXiv:1412.6572 (2014)
47. Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., Yuille, A.: Adversarial examples for semantic segmentation and object detection. In: Proceedings of the IEEE international conference on computer vision, pp. 1369–1378 (2017)
48. Nelson, D.M., Pereira, A.C., De Oliveira, R.A.: Stock market's price movement prediction with lstm neural networks. In: 2017 international joint conference on neural networks (IJCNN), pp. 1419–1426. Ieee (2017)
49. Shi, Y., Wang, Y., Qu, Y., Chen, Z.: Integrated gcn-lstm stock prices movement prediction based on knowledge-incorporated graphs construction. Int. J. Mach. Learn. Cybern. **15**(1), 161–176 (2024)
50. Gao, S., Wang, Y., Yang, X.: Stockformer: learning hybrid trading machines with predictive coding. In: IJCAI, pp. 4766–4774 (2023)