Trade When Opportunity Comes: Price Movement Forecasting via Locality-Aware Attention and Iterative Refinement Labeling

Liang Zeng[1] , **Lei Wang**[1] , **Hui Niu**[1] , **Ruchen Zhang**[2] , **Ling Wang**[2] and **Jian Li**[1]
[1]Institute for Interdisciplinary Information Sciences (IIIS), Tsinghua University, China
[2]Huatai Securities Co., Ltd, China
{zengl18, wanglei20, niuh17}@mails.tsinghua.edu.cn; {zhangruchen, wangling}@htsc.com
lijian83@mail.tsinghua.edu.cn

Abstract

Price movement forecasting, aimed at predicting financial asset trends based on current market information, has achieved promising advancements through machine learning (ML) methods. Most existing ML methods, however, struggle with the extremely low signal-to-noise ratio and stochastic nature of financial data, often mistaking noises for real trading signals without careful selection of potentially profitable samples. To address this issue, we propose LARA, a novel price movement forecasting framework with two main components: Locality-Aware Attention (LA-Attention) and Iterative Refinement Labeling (RA-Labeling). (1) LA-Attention, enhanced by metric learning techniques, automatically extracts the potentially profitable samples through masked attention scheme and task-specific distance metrics. (2) RA-Labeling further iteratively refines the noisy labels of potentially profitable samples, and combines the learned predictors robust to the unseen and noisy samples. In a set of experiments on three real-world financial markets: stocks, cryptocurrencies, and ETFs, LARA significantly outperforms several machine learning based methods on the Qlib quantitative investment platform. Extensive ablation studies confirm LARA's superior ability in capturing more reliable trading opportunities.

1 Introduction

Price movement forecasting, a crucial yet challenging task in quantitative finance, is notoriously difficult largely due to the financial market's inherently stochastic, dynamic, and volatile nature [De Prado, 2018]. The classical Efficient Market Hypothesis (EMH) [Fama, 1960; Lo, 2019] states that in an informationally efficient market, price changes must be unforecastable if all relevant information is reflected immediately. However, perfect efficiency is impossible to achieve in practice [Campbell *et al.*, 1997] due to informational asymmetry and "noisy" behaviors among the traders [Bloomfield *et al.*, 2009]. Indeed, hundreds of abnormal returns have been discovered in the literature (see the classical review by [Grossman and Stiglitz, 1980] and the recent list [Harvey *et al.*, 2016]). As the more recent endeavor to "beat the market" and achieve excess returns, machine learning based solutions, taking advantage of the nonlinear expressive power, have become increasingly popular and achieved promising results [De Prado, 2018; Gu *et al.*, 2018; Zhang *et al.*, 2020; Xu *et al.*, 2020; Wang *et al.*, 2019a].

The straightforward approach to utilizing machine learning techniques for financial forecasting can be distilled into two primary steps [Gu *et al.*, 2018]. First, mine effective factors (either manually or automatically [Li *et al.*, 2019]) correlated/predictive with asset returns. Second, feed these factors as features into some off-the-shelf machine learning algorithms to generate trading signals. Recently, much effort has been devoted to designing new machine learning algorithms in the second step to make more accurate predictions [Zhang *et al.*, 2020; Xu *et al.*, 2020; Zhang *et al.*, 2017; Wang *et al.*, 2019b; Yang *et al.*, 2020; De Prado, 2018]. Most works mentioned above trains the model over the entire set of training data (typically spans a consecutive period of time)[1]. However, it is well known that financial time-series data has an extremely low signal-to-noise ratio [De Prado, 2018], to the degree that modern machine algorithms are prone to pick up patterns of noise instead of real profitable signals (*a.k.a.,* overfit the training set). As depicted in Fig. 1, samples are mixed together in the feature space, making it challenging to clearly distinguish between them. Without careful selection of potentially profitable samples from the entire set of training data, tuning the models to generalize well can be very challenging [Zhang *et al.*, 2017; Kim *et al.*, 2021]. Even for the most experienced investment experts, asset prices often exhibit behaviors akin to unpredictable random walks, with profitable opportunities occurring only sporadically. In light of this fact, we argue that it can be more effective and robust to focus on specific samples that are potentially more predictable and profitable (Fig. 1), as the proverb says, *trade when opportunity comes*. Consequently, our goal is to develop a powerful and generalizable model for price movement forecasting over highly noisy data. However, this realistic financial scenario brings two unique technical challenges, which we will tackle in this paper.

[1]Note that in financial time-series forecasting problems, one should not split the training and testing set randomly, since the data points are not i.i.d., but strongly and temporally correlated.

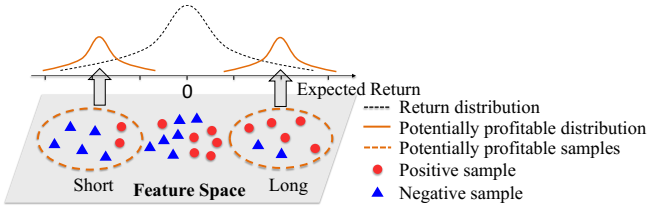

Figure 1: Illustration of the potential profitable samples. The half top figure represents the probability density function (PDF) of expected return over corresponding samples (best viewed in color).

Challenge 1: How to effectively extract potentially profitable samples?

Solution 1: *Locality-aware attention* (LA-Attention). We design two main modules for LA-Attention: a *metric learning module* for constructing a better metric, and a *localization module* for explicitly extracting potentially profitable samples. Specifically, many algorithms rely critically on being given a good metric over their inputs due to the complex internal relations among different features of financial data [Xing *et al.*, 2002]. It motivates us to introduce the *metric learning module* to assist the following *localization module* by learning a more compatible distance metric [De Vazelhes *et al.*, 2020]. The *localization module* intends to model an auxiliary relation between input samples and their labels by locally integrating label information to extract the desired potentially profitable samples.

Challenge 2: How to adaptively refine the labels of potentially profitable samples?

Solution 2: *Iterative refinement labeling* (RA-Labeling). We empirically identify that in the actual financial market, two samples may have similar features but yield completely opposite labels because of the stochastic and chaotic nature of financial data [De Prado, 2018; Lo and MacKinlay, 2011]. Such *noisy samples*, *i.e.,* similar samples with the opposite labels, hamper the model generalization in practical financial applications [Song *et al.*, 2022]. In response, we propose the novel RA-Labeling method to adaptively distinguish noisy labels of potentially profitable samples according to their training losses, iteratively refine their labels with multiple predictors, and combine the learned predictors to enhance the overall performance.

In short, our main **contributions** are as follows:

- We propose the two-step LA-Attention method to first locally extract potentially profitable samples by attending to label information and then construct a more accurate classifier based on these selected samples. Moreover, we utilize a metric learning module to learn a well-suited distance metric to assist LA-Attention.
- We introduce the RA-Labeling method to adaptively refine the noisy labels of potentially profitable samples. We obtain a noise-robust model with our proposed combination schemes to integrate the multiple learned predictors and boost generalization ability.
- We conduct comprehensive experiments on three real-world financial markets. LARA significantly outperforms a set of machine learning competitors on the Qlib quantitative investment platform [Yang *et al.*, 2020]. Extensive ablation studies and experiments demonstrate the effectiveness of each component in LARA and suggest that LARA indeed captures more reliable trading opportunities.

2 Related Work

Price Movement Forecasting. There are two main research threads in price movement forecasting based on machine learning methods. (1) [Li *et al.*, 2019; Achelis, 2001; Sawhney *et al.*, 2021; Xu *et al.*, 2020] focus on improving prediction by introducing extra financial data sources. [Gu *et al.*, 2018] used sentiment analysis to extract supplementary financial information from various textual data sources. Besides, there also exists much literature (*e.g.*, [Li *et al.*, 2019; Achelis, 2001]) studying new methods to construct more informative features for financial data. (2) [Zhang *et al.*, 2020; Wu *et al.*, 2021; Jiang *et al.*, 2022; Zhang *et al.*, 2017] aim to develop specialized prediction models to enhance prediction accuracy. [Zhang *et al.*, 2020] proposed a new ensemble method based on the sample reweighting and feature selection method for stock price prediction. Despite increasing efforts in price movement forecasting, few of them explicitly incorporate auxiliary label information in an extremely noisy market. We propose the two-step LA-Attention to effectively extract potentially profitable samples.

Handling Label Noise. Learning with noisy labels is a common challenge in many real-world applications [Bloomfield *et al.*, 2009; Zhang *et al.*, 2020; Song *et al.*, 2022]. [Chen *et al.*, 2021] proposed a new labeling method called SEAL (Self-Evolution Average Label) to tackle instance-dependent label noise. Apart from the labeling method, [Kim *et al.*, 2021] proposed filtering noisy instances via their eigenvectors (FINE), while [Xia *et al.*, 2021] dynamically selected training samples by analyzing the prediction values between two models. However, in the context of quantitative investment, these methods may not suffice due to extreme noise levels, as illustrated in Fig. 1. Simple label modifications or noisy sample removal can inadvertently increase uncertainty or eliminate the potentially profitable opportunities. In contrast, our RA-Labeling method is designed to adaptively refine the noisy labels of potentially profitable samples to build a noise-robsut model.

3 Preliminary

Problem Setup. Consider a time series of the asset trading price over T time-steps $\{\text{Price}_t | t = 1, 2, \ldots, T\}$. Typically, the price movement trend is defined as the future change (return) of the asset price [De Prado, 2018]. In this paper, we adopt a fixed-time horizon method to define the asset price trend. Concretely, the label y_t indicates whether the return of the asset price over a time horizon Δ exceeds a certain predefined threshold λ. For long positions:

$$y_t = \begin{cases} 1, & \text{Price}_{t+\Delta}/\text{Price}_t - 1 > \lambda \\ 0, & \text{otherwise} \end{cases} . \quad (1)$$

Similarly, for short positions, we set the label 1 if $\text{Price}_{t+\Delta}/\text{Price}_t - 1 < -\lambda$ and 0 otherwise. At the train-

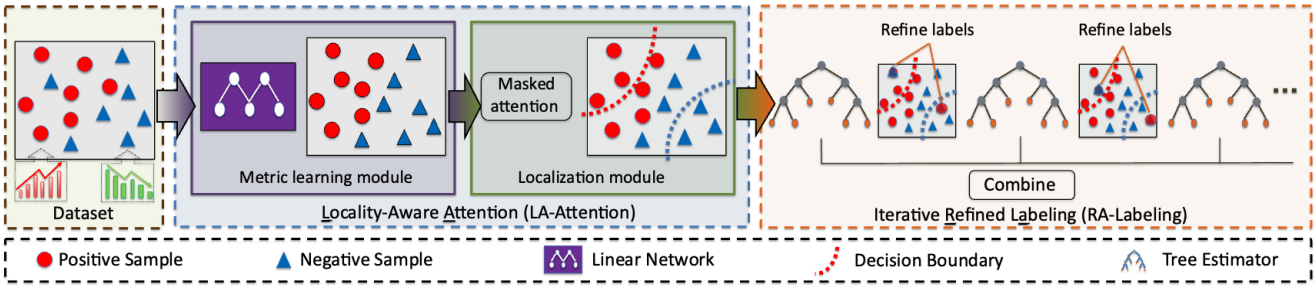

Figure 2: The workflow of our proposed LARA framework. LARA first extracts potentially profitable samples from the noisy market and then refines their labels. LARA consists of two sequential components: LA-Attention and RA-Labeling.

ing step t, we calculate the feature vector $\boldsymbol{x}_t \in \mathcal{X} \subseteq \mathbb{R}^d$ based on price and volume information up to time t, and its label $y_t \in \mathcal{Y} = \{0, 1\}$. We denote the training set as $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]^T \in \mathbb{R}^{N \times d}$ associated with its label $\boldsymbol{y} = [y_1, \cdots, y_N]^T \in \mathbb{R}^N$, where N is the number of time steps and d is the feature dimension. Similarly, $\boldsymbol{X}_{\text{test}} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{N_{\text{test}}}]^T \in \mathbb{R}^{N_{\text{test}} \times d}$ denotes the testing set.

Goal. *Price movement forecasting* aims to build a (binary) parameterized classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$, where $f_\theta(\boldsymbol{x}_t) = Pr(y_t = 1)$ represents the probability of whether the price trend y_t exceeds the pre-defined threshold λ. In order to prevent the data leakage issue of financial data, $f_\theta(\boldsymbol{x}_t)$ makes a prediction on the asset price trend y_t of Δ-step ahead $(t + \Delta)$ only based on price information up to t.

Samples With High $p_{\boldsymbol{x}}$. Because we consider the binary classification, we assume the predicted label $y_t \sim \text{Bernoulli}(p_{\boldsymbol{x}_t})$ conditional on known information $\boldsymbol{x}_t$ at the t time step. It means that we observe the positive class with the probability $p_{\boldsymbol{x}_t}$ based on the current market condition. A larger $p_{\boldsymbol{x}_t}$ indicates a stronger tendency and lower uncertainty of the asset price, and thus the prediction at this time step can be more reliable. In order to make a more accurate prediction, we gather the *samples with high* $p_{\boldsymbol{x}_t}$, where $p_{\boldsymbol{x}_t}$ is greater than a given threshold $(thres)$, *i.e.,* $p_{\boldsymbol{x}_t} > thres$. However, it is impossible to obtain the exact underlying probability $p_{\boldsymbol{x}_t}$ from the input data since we can only observe one sample of the distribution $y_t \sim \text{Bernoulli}(p_{\boldsymbol{x}_t})$ at one certain time-step. Thus, we introduce *locality-aware attention* (Sec. 4.1) to give an approximate estimation $\hat{p}_{\boldsymbol{x}_t}$ of $p_{\boldsymbol{x}_t}$.

4 LARA: The Proposed Framework

In this section, we present our LARA framework, which consists of two main components—*locality-aware attention* (LA-Attention, Sec. 4.1) and *iterative refinement labeling* (RA-Labeling, Sec. 4.2). The key idea of LARA is to extract potentially profitable samples from the noisy market. LA-Attention extracts the *samples with high* $p_{\boldsymbol{x}}$ and RA-Labeling refines the labels of *noisy samples* to boost the performance of predictors. The workflow of LARA is shown in Fig. 2.

4.1 Locality-Aware Attention (LA-Attention)

LA-Attention is a two-step method for its training and testing phases. In the training phase, we first extract the *samples with high* $p_{\boldsymbol{x}_t}$ satisfying $\hat{p}_{\boldsymbol{x}_t} > thres$ ($thres$ denotes the desired probability that one sample yields the positive class), which implicitly denote the potentially profitable samples. Then we train a more robust model on these selected samples according to the corresponding supervised losses. In the testing phase, we first gather the *samples with high* $p_{\boldsymbol{x}_t}$ by the $\hat{p}_{\boldsymbol{x}_t} > thres$ criterion, and then we only make predictions on these selected samples. Thanks to this two-step modular design, LA-Attention can largely improve the precision of the prediction model on empirical evaluations. The complete algorithm is summarized in Algorithm 1. In what follows, we first introduce the localization module to estimate $\hat{p}_{\boldsymbol{x}_t}$ and then introduce the metric learning module to assist the *localization module* by learning a well-suited distance metric.

Localization Module. It is hard to recover the label distribution $y_t \sim \text{Bernoulli}(p_{\boldsymbol{x}_t})$ from the noisy asset price dataset, as we can only observe one sample of the distribution at the t-th time-step. Suppose the parameter $p_{\boldsymbol{x}_t}$ of the probability distribution is continuous with respect to known information $\boldsymbol{x}_t$ (*e.g.,* the asset price). We design a *localization module* attending to other observed samples in the dataset to obtain an approximate estimation $\hat{p}_{\boldsymbol{x}_t}$ of $p_{\boldsymbol{x}}$. In light of the concepts of attention mechanism [Vaswani *et al.*, 2017], we denote $(\boldsymbol{x}_i, y_i)$ as the key-value pair, where y_i is the label of $\boldsymbol{x}_i$. Let $\boldsymbol{x}_t$ be the query sample. The *localization module* can be formulated as follows:

$$\hat{p}_{\boldsymbol{x}_t} = \sum_{1 \le i \le N} y_i \cdot \frac{k(\boldsymbol{x}_i, \boldsymbol{x}_t)}{\sum_{1 \le j \le N} k(\boldsymbol{x}_j, \boldsymbol{x}_t)} \quad , \tag{2}$$

where N is the size of training data, and $k(\boldsymbol{x}_i, \boldsymbol{x}_t)$ is the attention weight between $\boldsymbol{x}_i$ and $\boldsymbol{x}_t$. However, in the most general attention mechanisms, the model allows each sample to attend to every other sample. Intuitively, closer neighbors of a query sample have greater *similarity* than neighbors which are further away. We inject the locality structure into the module by performing the *masked attention* scheme and merely paying attention to the neighbors $\mathcal{N}(\boldsymbol{x}_t)$ of $\boldsymbol{x}_t$. Then we reformulate the *localization module* as follows:

$$\hat{p}_{\boldsymbol{x}_t} = \sum_{\boldsymbol{z}_i \in \mathcal{N}(\boldsymbol{x}_t)} y_{\boldsymbol{z}_i} \cdot \frac{k(\boldsymbol{z}_i, \boldsymbol{x}_t)}{\sum_{\boldsymbol{z}_j \in \mathcal{N}(\boldsymbol{x}_t)} k(\boldsymbol{z}_j, \boldsymbol{x}_t)} \quad . \tag{3}$$

We propose two schemes (detailed in Appendix) to define the neighbors $\mathcal{N}(\boldsymbol{x}_t)$ in the *localization module*: 1) the k-nearest

neighbors of the query point (K-Neighbor); 2) the k-nearest neighbors within a moderate radius (R-Neighbor). Moreover, there exist several reasonable similarity metrics to implement the attention weight in Eq.(3). We recommend two practical approaches: 1) the identical weight; 2) the reciprocal of Mahalanobis distance $d_{\boldsymbol{M}}(\boldsymbol{z}, \boldsymbol{x}_t)$ (introduced below), detailed as follows:

$$k(\boldsymbol{z}, \boldsymbol{x}_t) := \begin{cases} k_I(\boldsymbol{z}, \boldsymbol{x}_t) = 1 \\ k_R(\boldsymbol{z}, \boldsymbol{x}_t) = 1/d_{\boldsymbol{M}}(\boldsymbol{z}, \boldsymbol{x}_t) \end{cases} . \tag{4}$$

Metric Learning Module. As introduced above, we need a reasonable measurement of the distance between data representations on the embedding space to select *samples with high* p_{x_t}. However, in the context of quantitative investment, different investors have different understandings of the financial market and prefer completely different types of factors, *e.g.,* value investors prefer fundamental factors, while technical analysts may choose technical factors. Due to the multimodal nature of financial data, we adopt the metric learning technique to mitigate the pressure of manually searching for the suitable distance measurement and enhance the flexibility of LARA. The goal of metric learning is to learn a Mahalanobis distance metric $\boldsymbol{M}$ to pull the similar samples close and push the dissimilar samples away. A straightforward solution is to minimize the dissimilarity of samples within the same class. We first define the indicator function as $K_{i,j} = 1$ if $y_i = y_j$ and 0 otherwise. Given two embedding vectors $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$, we optimize the (squared) Mahalanobis matrix $\boldsymbol{M} \in \mathbb{R}^{d \times d}$ [Kaya and Bilge, 2019] as follows:

$$\begin{aligned} &\min_{\boldsymbol{M}} \quad \frac{1}{2} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}} d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) K_{i,j} \\ =&\min_{\boldsymbol{M}} \quad \frac{1}{2} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}} (\boldsymbol{x}_i - \boldsymbol{x}_j)^T \boldsymbol{M} (\boldsymbol{x}_i - \boldsymbol{x}_j) K_{i,j} \\ =&\min_{\boldsymbol{M}} \sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}} (\boldsymbol{x}_i^T \boldsymbol{M} \boldsymbol{x}_i - \boldsymbol{x}_i^T \boldsymbol{M} \boldsymbol{x}_j) K_{i,j} \quad , \end{aligned} \tag{5}$$

where d is the feature dimension, and $d_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ denotes the Mahalanobis distance. Metric learning module can better leverage the similarity and dissimilarity among samples by learning a new embedding space. Thus, LARA adopts the *metric learning module* to automatically learn a well-suited distance metric in the embedding space, assisting the *localization module* in extracting potentially profitable samples.

LA-Attention Has a Proper Estimation of $p_{\boldsymbol{x}_t}$. We use a running example to illustrate why LA-Attention helps extract potentially profitable samples and further understand the effectiveness of the *metric learning module* incorporated into the LA-Attention method, as shown in Fig. 3. We search for potentially profitable *samples with high* $p_{\boldsymbol{x}_t}$ via the $\hat{p}_{\boldsymbol{x}_t} > thres$ criterion ($thres$ equals 0.5 in this demo). We aim to verify whether $\hat{p}_{\boldsymbol{x}_t}$ calculated by LA-Attention is a proper estimation of the predefined $p_{\boldsymbol{x}_t}$. In Fig. 3 (Left), We plot embeddings of randomly sampled 1000 points in the ETF dataset using t-SNE [Van der Maaten and Hinton, 2008]. We find that the positive and negative points are mixed together and there are only 174 positive samples. It exhibits the return distribution with almost the zero mean, and $\hat{p}_{\boldsymbol{x}_t} \approx 0.174$ is far away from the desired $p_{\boldsymbol{x}_t}$. After introducing the *localization module* (Fig. 3 (Middle)), we also randomly sample 1000 points with $\hat{p}_{\boldsymbol{x}_t} > thres$ criteria, and find that there are 279 positive samples. The mean return of potentially profitable samples is obviously larger than zero, which indicates the effectiveness of the *localization module*. However, the negative ones still account for a large part of the samples and the estimated $\hat{p}_{\boldsymbol{x}_t}$ (0.279) is much lower than desired $thresh = 0.5$. In Fig. 3 (Right), further incorporated into the *metric learning module*, the number of the positive samples (424) accounts for almost half of all samples, which is in line with expectations that $\hat{p}_{\boldsymbol{x}_t}$ equals 0.5 and is a proper estimation of $p_{\boldsymbol{x}_t}$. We also find that the learned embedding has a more compact structure and clearer boundaries. It achieves the $8.42E$-4 mean return, which is much larger than zero with a greater margin compared with the other two methods (note that $8.42E$-4 is a reasonable and effective return in high-frequency trading and it is close to our target return of $\lambda = 1E$-3). Hence, this learned distance metric can assist the LA-Attention method in distinguishing potentially profitable *samples with high* $p_{\boldsymbol{x}_t}$.

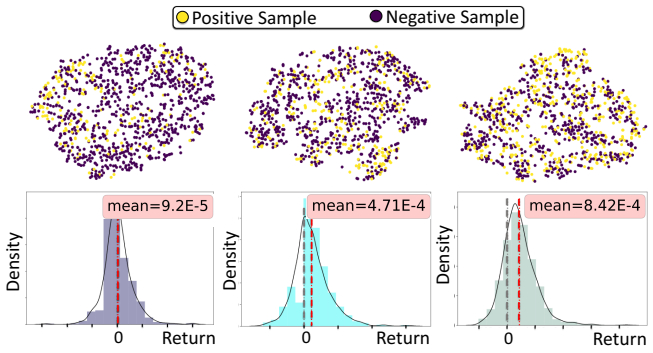

Figure 3: Visualization of randomly sampled 1000 points with t-SNE and their corresponding return distributions. **Left:** the input dataset. **Middle:** *samples with high* $p_{\boldsymbol{x}_t}$ selected by *localization module*. **Right:** *samples with high* $p_{\boldsymbol{x}_t}$ selected by LA-Attention (*localization module + metric learning module*).

4.2 Iterative Refinement Labeling (RA-Labeling)

Price movement is notoriously difficult to forecast because financial markets are complex dynamic systems [De Prado, 2018; Lo and MacKinlay, 2011]. Indeed, we observe that the complicated financial markets sometimes mislead ML systems to produce opposite prediction results with extremely high confidence. For example, some training samples get a high (low) probability of the positive prediction $Pr(y = 1)$, but their labels are negative (positive). Such *noisy samples* hamper the model generalization in price movement forecasting [Li *et al.*, 2018]. Thus, LARA should be robust to noisy labels of potentially profitable samples, and we adopt the label refurbishment technique introduced by [Song *et al.*, 2022] to learn a noise-robust prediction model. Formally, given the training data $\boldsymbol{X} \in \mathbb{R}^{N \times d}$, our method runs in an iterative manner to obtain adaptively refined labels $\{\boldsymbol{y}^k\}_{0 \le k \le K}$, where N is the number of training data and K is the number

Algorithm 1: LARA framework

Input: Training data $(\boldsymbol{X}, \boldsymbol{y}) \in \mathbb{R}^{N\times d} \times \mathbb{R}^N$, testing data $\boldsymbol{X}_{\text{test}} \in \mathbb{R}^{N_{\text{test}}\times d}$, the model f.
Parameter: Threshold $thres$, combination scheme C, hyper-parameters K.
Output: $Y_{out} \in \mathbb{R}^N$.
▷ **Training phase:**
Init *emtpy set* $\boldsymbol{X}', \boldsymbol{y}'$.
for $(\boldsymbol{x}, y)$ ***in*** $(\boldsymbol{X}, \boldsymbol{y})$ **do**
  Estimate $\hat{p}_{\boldsymbol{x}}$ via Eq. 3 with the query sample $\boldsymbol{x}$ and the key-value pair $(\boldsymbol{x}_i, y_i) \in (\boldsymbol{X}, \boldsymbol{y})$.
  if $\hat{p}_x > thres$ **then** $\boldsymbol{X}' \cup \{\boldsymbol{x}\}$; $\boldsymbol{y}' \cup \{y\}$.
F = RA-Labeling $(\boldsymbol{X}', \boldsymbol{y}', f, K, C)$; */*Algorithm 2*/*
▷ **Testing phase:**
Init *emtpy set* $\boldsymbol{X}'_{\text{test}}$.
for $\boldsymbol{x}_{test}$ ***in*** $\boldsymbol{X}_{test}$ **do**
  Estimate $\hat{p}_{\boldsymbol{x}_{\text{test}}}$ via Eq. 3 with the query sample $\boldsymbol{x}_{\text{test}}$ and the key-value pair $(\boldsymbol{x}_i, y_i) \in (\boldsymbol{X}, \boldsymbol{y})$.
  if $\hat{p}_{\boldsymbol{x}} > thres$ **then** $\boldsymbol{X}'_{\text{test}} \cup \{\boldsymbol{x}_{\text{test}}\}$.
Return: $Y_{out} = F(\boldsymbol{X}'_{\text{test}})$.

of rounds in iterative refinement labeling.

We denote the trained predictor at the k-th round and the current prediction of the sample j at the k-th round as $\{f_{\theta^k}\}_{0\le k\le K}$ and $f_{\theta^k}(\boldsymbol{x}_j)$, respectively. The refurbished label y_j^{k+1} can be obtained by a convex combination of the current label y_j^k and the associated prediction $f_{\theta^k}(\boldsymbol{x}_j)$:

$$y_j^{k+1} = \alpha_j^k y_j^k + (1-\alpha_j^k) f_{\theta^k}(\boldsymbol{x}_j), \tag{6}$$

where $\alpha_j^k \in [0,1]$ is the label confidence and y_j^0 is the initial noisy label of training data. As shown in Eq. (6), we apply the exponential moving average to mitigate the damage of under-predicted labels of *noisy samples* and thus the confidence score α_j^k is a crucial parameter in this self-adaptive training scheme. We propose to update α_j^k according to the training loss at the current step to alleviate the instability issue of using instantaneous prediction under *noisy samples*. α_j^k of the sample $\boldsymbol{x}_j$ at the k-th step is calculated by:

$$\alpha_j^k = 1\left\{\mathcal{L}\left(f_{\theta^k}(\boldsymbol{x}_j), y_j^k\right) \le \epsilon^k\right\}, \tag{7}$$

where $1\{\cdot\}$ denotes the indicator function, ϵ^k is a hyper-parameter to control label confidence, and $\mathcal{L}(\cdot,\cdot)$ denotes the supervised loss function, *e.g.*, the cross-entropy loss. In order to develop a more coherent predictor that improves its ability to evaluate the consistency of *noisy samples*, we adaptively set the value of ϵ^k to roughly adjust the ratio of labels in training data, denoted by r. We also perform the hyperparameter study in Sec. 5.3 and empirically identify that RA-Labeling achieves a relatively stable performance when $r < 10\%$.

Different from the conventional noisy labeling methods [Song *et al.*, 2022], we further propose a combination method to integrate the learned predictor. Specifically, after running K iterations of iterative refinement labeling, we obtain $K+1$ iterated predictors $\{f_{\theta_0}, \cdots, f_{\theta_K}\}$. We integrate all predictors using the following two different combination

Algorithm 2: RA-Labeling

Input: Training data $\left(\boldsymbol{X}, \boldsymbol{y}^0\right) \in \mathbb{R}^{N\times d} \times \mathbb{R}^N$, the model $f : \mathbb{R}^d \to \mathbb{R}$.
Parameter: Iteration number K, combination scheme C.
Output: $F : \mathbb{R}^d \to \mathbb{R}$
Init: $\theta^0 = \arg\min_{\theta^0} \sum_{j=1}^N \mathcal{L}\left(f_{\theta^0}(\boldsymbol{x}_j), y_j^0\right)$
for $k \in \{0, \ldots, K-1\}$ **do**
  for $j \in \{1, \ldots, N\}$ **do**
    $y_j^{k+1} = \alpha_j^k y_j^k + (1-\alpha_j^k) f_{\theta^k}(\boldsymbol{x}_j)$
  $\theta^{k+1} = \arg\min_{\theta^{k+1}} \sum_{j=1}^N \mathcal{L}\left(f_{\theta^{k+1}}(\boldsymbol{x}_j), y_j^{k+1}\right)$
Return: $F = C\left(\{f_{\theta^k} | 0 \le k \le K\}\right)$.

schemes: (1) $C = C_{\text{Last}}$: making a prediction according to the last iterated predictor. (2) $C = C_{\text{Vote}}$: making a prediction that averages all iterated predictors. Then the output prediction model F is defined as follows:

$$F(\boldsymbol{x}) := \begin{cases} C_{\text{Last}}\left(\{f_{\theta^k}\}_{0\le k\le K}\right)(\boldsymbol{x}) = f_{\theta^K}(\boldsymbol{x}), & C = C_{\text{Last}} \\ C_{\text{Vote}}\left(\{f_{\theta^k}\}_{0\le k\le K}\right)(\boldsymbol{x}) = \mathbb{E}_k\left[f_{\theta^k}(\boldsymbol{x})\right], & C = C_{\text{Vote}} \end{cases}. \tag{8}$$

Note that RA-Labeling can be seamlessly incorporated into existing machine learning methods to mitigate the negative consequences of learning from noisy labels. The pseudocode of RA-Labeling is shown in Algorithm 2.

5 Experiments

In this section, we perform extensive experiments on three real-world financial markets: China's A-share stock, the cryptocurrency (OKEx), and ETFs to validate the effectiveness of LARA. We first introduce the experimental setup (Sec. 5.1), and then benchmark the LARA framework compared with deep learning methods on the Qlib quantitative investment platform [Yang *et al.*, 2020] (Sec. 5.2). Finally, we provide experimental analysis to demonstrate the effectiveness of each component in LARA (Sec. 5.3) and ablation studies to show the sensitivity of hyperparameters in LARA (in Appendix).

5.1 Experimental Setups

Dataset Collection. For stocks, we use the dataset from Qlib: the daily data of the constituents of CSI300[2]. We follow the temporal order to split datasets into training (01/01/2008–12/31/2014), validation (01/01/2015–12/31/2016), and testing (01/01/2017–08/01/2020) data. For cryptocurrency, we use the BTC/USDT trading pair. Each record is one market snapshot captured for approximately every 0.1 second. The training data comes from 7 consecutive trading days (01/03/2021–07/03/2021), whose total quantity is 10 million. The validation and testing data

[2]CSI300 consists of the 300 largest and most liquid A-share stocks, which reflects the overall performance of the market.

Methods		China's A-share stocks			Cryptocurrency			Ranking Count	
		PR(%)	WLR	AR	PR(%)	WLR	AR	1^{st}	2^{nd}
Quantitative Investment Methods	ALSTM	51.9	1.133	3.24E-3	47.9	0.933	5.9E-5	0	0
	TabNet	51.8	1.299	5.14E-3	51.0	0.890	1.4E-5	0	0
	Transformer	53.2	1.230	5.73E-3	38.7	0.888	-5.0E-5	0	0
	Adamct	52.7	<u>1.309</u>	5.73E-3	49.3	<u>1.177</u>	1.3E-4	0	**2**
	LightGBM	55.0	**1.331**	<u>7.26E-3</u>	51.0	0.890	1.4E-5	<u>1</u>	<u>1</u>
	DoubleEnsemble	54.0	1.225	5.75E-3	-	-	-	0	0
	TCTS	55.6	0.913	2.09E-3	-	-	-	0	0
Time-series Methods	iTransformer	53.8	1.095	4.01E-3	-	-	-	0	0
	PatchTST	53.0	1.274	5.17E-3	-	-	-	0	0
	TimesNet	55.5	1.131	6.16E-3	-	-	-	0	0
Noisy Labels Methods	CNLCU	52.6	1.233	5.03E-3	52.9	**1.220**	1.2E-4	<u>1</u>	0
	FINE	55.3	1.070	4.35E-3	<u>56.3</u>	0.863	9.7E-5	0	<u>1</u>
	SEAL	<u>56.6</u>	1.200	5.95E-3	53.0	0.969	7.2E-5	0	<u>1</u>
Ours	LA-Attention	<u>56.6</u>	1.142	5.27E-3	51.2	0.826	9.5E-5	0	<u>1</u>
	RA-Labeling	55.2	1.038	3.53E-3	56.2	1.034	<u>1.4E-4</u>	0	<u>1</u>
	LARA	**59.1**	1.274	**7.79E-3**	**57.8**	1.059	**1.5E-4**	**4**	0

Table 1: Quantitative comparisons among different methods on the China's A-share stocks and the cryptocurrency (BTC/USDT). - means that the corresponding method is either not implemented or unsuitable for corresponding setting. We retrieve the top 1000 signals with the highest probability for each experiment. The best performance is highlighted in **bold**. The second best is highlighted with <u>underline</u>.

come from the following 7 trading days (08/03/2021–10/03/2021 and 11/03/2021–14/03/2021). We use technical factors (*e.g.*, moving average of price) and limit-order book factors (*e.g.*, buy/sell pressure indicators) to calculate input features [Achelis, 2001]. For ETFs, we conduct experiments over four Chinese exchange-traded funds (ETFs): 159915.SZ, 512480.SH, 512880.SH, and 515050.SH. All of them are liquid and traded in large volumes. We chronologically split datasets into three time periods for training (01/01/2020–17/04/2020), validation (20/04/2020–29/05/2020), and testing (01/06/2020–06/07/2020). Due to the space limit, we only show the experimental results of 512480.SH. The results for other ETFs, the complete factor list, and the statistics of all datasets can be found in Appendix.

Baselines. (1) *Quantitative investment methods in Qlib:* ALSTM [Qin *et al.*, 2017], TabNet [Arık and Pfister, 2021], Transformer [Vaswani *et al.*, 2017], Adamct [Jiang *et al.*, 2022], LightGBM [Ke *et al.*, 2017], DoubleEnsemble [Zhang *et al.*, 2020], and TCTS [Wu *et al.*, 2021]. (2) *Time-series analysis methods:* Due to the temporal nature of daily stock data, we also implemented several SOTA time-series methods: iTransformer [Liu *et al.*, 2023], PatchTST [Nie *et al.*, 2022], TimesNet[Wu *et al.*, 2022]. (3) *Noisy labels methods:* CNLCU [Xia *et al.*, 2021], FINE [Kim *et al.*, 2021], SEAL [Chen *et al.*, 2021]. For these methods, we utilize LightGBM as the base model.

Performance Metrics. We use four commonly-used financial metrics [Yang *et al.*, 2020] to evaluate the performance of LARA, *i.e.*, *Precision* (PR), *Win-Loss Ratio* (WLR), *#Transactions*, and *Average Return* (AR). For the descriptions of each metric, please check the appendix for more details.

Evaluation Protocols. Following [Zhang *et al.*, 2020], we adopt a representative trading strategy to evaluate the performance regarding the actual trading scenarios of the market. *Note that we build individual models for long and short positions.* (1) *Trading signals:* #Transactions is an important criterion to measure the transaction frequency of trading strategies in high-frequency quantitative trading. According to [Abner, 2010], it is a common practice to set thresholds to trigger tradings signals, but appropriate thresholds remains a challenging research issue. We retrieve the top 100 to 1000 samples with the highest predicted probability as the corresponding trading signals (control the same trading frequency for different algorithms). (2) *Trading Strategies:* we do not set a trading position limit. At a certain time step, if the model gives a positive signal, we open a new long position and close this position after the prediction horizon $\triangle$ (1 day/1 min/10 secs for stocks/ETFs/crypto). Similarly, we open short positions for the negative signals and close them after $\triangle$.

For LARA, we use the default hyperparameter in *LightGBM* and the *metric learning module*. Other methods use their suggested parameters on Qlib or we conduct a gird search for them, the details can be found in Appendix. All experiments are repeated 5 times, and we report the mean results with standard deviations. In what follows, we always record *the average results of long and short positions.*

5.2 Experimental Results

Main Results. In Table 1, we show the quantitative results on China's A-share stocks and the cryptocurrency (OKEx). LARA achieves 59.1% and 57.8% precision on stocks and crypto respectively, and outperforms the others by up to 2.5% and 1.5%. As for the average return, LARA performs better than other competitors on stocks and improves a great

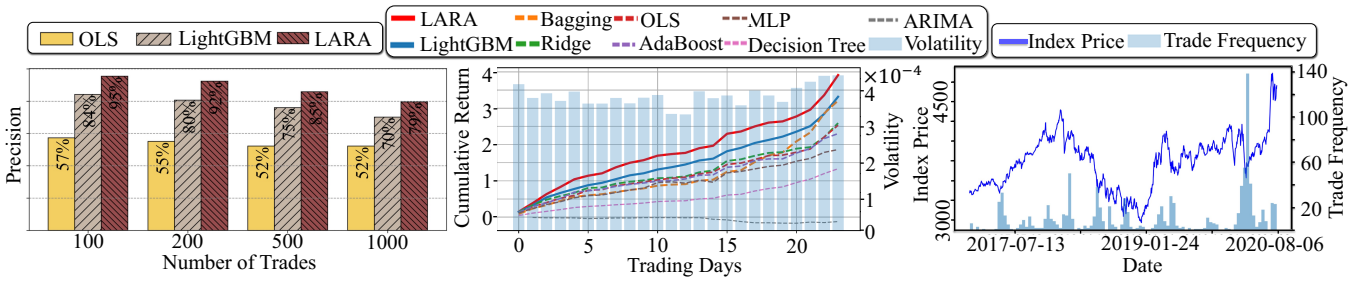

Figure 4: **Left:** Precision with the different number of trades (#Transactions) among three methods on 512480.SH. **Middle:** Quantitative comparisons over cumulative return between LARA and a set of baselines on 512480.SH. The right y-axis illustrates the volatility [Campbell *et al.*, 1997] on each trading day. **Right:** Plot of the trade frequency in the China's A-share market and the corresponding stock index price. The left y-axis denotes the stock index price while the right y-axis denotes the trade frequency.

Training	Testing	PR (%)	WLR	AR (%)
-	-	$70.16_{\pm 0.89}$	$2.872_{\pm 0.106}$	$0.166_{\pm 0.002}$
-	✓	$70.39_{\pm 1.07}$	$2.757_{\pm 0.184}$	$0.166_{\pm 0.002}$
✓	-	$37.82_{\pm 11.09}$	$1.712_{\pm 0.188}$	$0.062_{\pm 0.037}$
✓	✓	$\mathbf{78.58_{\pm 0.67}}$	$\mathbf{2.957_{\pm 0.259}}$	$\mathbf{0.196_{\pm 0.003}}$

Table 2: Ablation studies with LA-Attention applied in the training and testing phase, respectively. ✓(-) means that we evaluate w (w/o) LA-Attention in the corresponding phase.

margin on the crypto. The outstanding performance of these two indicators demonstrates that LARA indeed has the ability to capture potential profitable signals. For win-loss ratio, LARA can still achieve a decent effect comparable to other methods (WLR is larger than 1), indicating a relatively low trading risk in real-world markets. In addition, noisy labels methods usually perform better than other methods, indicating the necessity of handling noisy labels on financial data. Our LARA not only extracts potential profitable samples but also dynamically modifies their labels for training, culminating in superior performance compared to other methods of noisy labels methods.

Effectiveness of LA-Attention. We then explore whether LA-Attention can boost performance when concentrating on the *samples with high* $p_{\boldsymbol{x}}$. We conduct experiments on ETFs to consider whether to use LA-Attention in the training and testing phases. In Table 2, we find that LA-Attention applied in both the training and testing phases outperforms other ablation competitors on all performance metrics. This result is in line with the fact that LA-Attention can capture much more reliable trading opportunities with a significantly higher average return. If we merely use LA-Attention in the [training/testing] phase, the performance decreases [8%/40%] on precision. We posit that the significant gap stems from substantial deviations in the distributions of datasets between these two phases. Thus, we should pay attention to the *samples with high* $p_{\boldsymbol{x}}$ in both two phases to capture more reliable trading opportunities.

5.3 Experimental Analysis

Transactions. First, in order to investigate the effect of #Transactions, we choose two representative methods: OLS, LightGBM (widely used for investment). In Fig. 4 left, LARA consistently outperforms the other two models in terms of precision under different #Transactions. These results highlight the effectiveness of LARA across different trading frequencies.

Trading Performance. Second, we illustrate the performance of the back-testing strategy in terms of cumulative return (Fig. 4 middle). We choose the top 1000 signals with the highest predicted probability for fair comparison and do not set position limits. For the sake of simplicity, transaction costs are ignored since #Transaction is the same over all methods, which does not change conclusions. It is obvious that LARA consistently outperforms other baseline models throughout the testing period, which demonstrates that LARA indeed improves the prediction ability of the model on the asset price trend.

Trade Distribution. Third, in Fig. 4 right, we plot the trade distribution of LARA in China's A-share market associated with the market index price. We can clearly see that when the index price fluctuates sharply, LARA trades stocks in a higher frequency. It meets our expectation that there are more opportunities when the market is highly volatile. These results further demonstrate that LARA can be used to guide real-world quantitative investments to pick up reliable trading opportunities.

6 Conclusion

We study the problem of price movement forecasting and propose the LARA (Locality-Aware Attention and Iterative Refinement Labeling) framework. In LARA, we introduce LA-Attention to extract potentially profitable samples and RA-Labeling to adaptively refine the noisy labels of potentially profitable samples. Extensive experiments on three real-world financial markets showcase its superior performance over time-series analysis methods, machine learning based models and noisy labels methods. Besides, we also illustrate how each component works by comprehensive ablation studies, which indicates that LARA indeed captures more reliable trading opportunities. We expect our work to pave the way for price movement forecasting in more realistic quantitative trading scenarios.

Acknowledgements

Jian Li, Liang Zeng, Lei Wang and Hui Niu were supported in part by the National Natural Science Foundation of China Grant 62161146004.

Contribution Statement

Liang Zeng and Lei Wang have made equal and significant contributions to this work, including the methods, implementation, and paper writing. Hui Niu conducted the literature review and implemented the evaluation framework. Ruchen Zhang and Ling Wang provided the source data to support this work. Jian Li, as the corresponding author, contributed to the idea and paper writing, as well as providing computing resources.

References

[Abner, 2010] David J Abner. *The ETF handbook: how to value and trade exchange traded funds*. John Wiley & Sons, 2010.

[Achelis, 2001] Steven B Achelis. *Technical Analysis from A to Z*. McGraw Hill New York, 2001.

[Arık and Pfister, 2021] Sercan O Arık and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In *Proc. of AAAI*, 2021.

[Bloomfield *et al.*, 2009] Robert Bloomfield, Maureen O'hara, and Gideon Saar. How noise trading affects markets: An experimental analysis. *The Review of Financial Studies*, 2009.

[Campbell *et al.*, 1997] John Y Campbell, Andrew W Lo, A Craig MacKinlay, and Robert F Whitelaw. *The econometrics of financial markets*. Princeton University Press, 1997.

[Chen *et al.*, 2021] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 11442–11450, 2021.

[De Prado, 2018] Marcos Lopez De Prado. *Advances in financial machine learning*. John Wiley & Sons, 2018.

[De Vazelhes *et al.*, 2020] William De Vazelhes, CJ Carey, Yuan Tang, Nathalie Vauquier, and Aurélien Bellet. metric-learn: Metric learning algorithms in python. *Journal of Machine Learning Research*, 2020.

[Fama, 1960] Eugene F Fama. Efficient market hypothesis. *Diss. PhD Thesis, Ph. D. dissertation*, 1960.

[Grossman and Stiglitz, 1980] Sanford J Grossman and Joseph E Stiglitz. On the impossibility of informationally efficient markets. *The American economic review*, 1980.

[Gu *et al.*, 2018] Shihao Gu, Bryan Kelly, and Dacheng Xiu. Empirical asset pricing via machine learning. Technical report, 2018.

[Harvey *et al.*, 2016] Campbell R Harvey, Yan Liu, and Heqing Zhu. ... and the cross-section of expected returns. *The Review of Financial Studies*, 2016.

[Jiang *et al.*, 2022] Juyong Jiang, Jae Boum Kim, Yingtao Luo, Kai Zhang, and Sunghun Kim. Adamct: Adaptive mixture of cnn-transformer for sequential recommendation. *arXiv:2205.08776*, 2022.

[Kaya and Bilge, 2019] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *NeurIPS*, 2017.

[Kim *et al.*, 2021] Taehyeon Kim, Jongwoo Ko, JinHwan Choi, Se-Young Yun, et al. Fine samples for learning with noisy labels. *Advances in Neural Information Processing Systems*, 34:24137–24149, 2021.

[Li *et al.*, 2018] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Proc. of NeurIPS*, 2018.

[Li *et al.*, 2019] Zhige Li, Derek Yang, Li Zhao, Jiang Bian, Tao Qin, and Tie-Yan Liu. Individualized indicator for all: Stock-wise technical indicator optimization with stock embedding. In *Proc. of KDD*, 2019.

[Liu *et al.*, 2023] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

[Lo and MacKinlay, 2011] Andrew W Lo and A Craig MacKinlay. *A non-random walk down Wall Street*. Princeton University Press, 2011.

[Lo, 2019] Andrew W Lo. *Adaptive markets: Financial evolution at the speed of thought*. Princeton University Press, 2019.

[Nie *et al.*, 2022] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

[Qin *et al.*, 2017] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *Proc. of IJCAI*, 2017.

[Sawhney *et al.*, 2021] Ramit Sawhney, Shivam Agarwal, Arnav Wadhwa, and Rajiv Shah. Exploring the scale-free nature of stock markets: Hyperbolic graph learning for algorithmic trading. In *Proc. of WWW*, 2021.

[Song *et al.*, 2022] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE TNNLS*, 2022.

[Van der Maaten and Hinton, 2008] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 2008.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of NeurIPS*, 2017.

[Wang *et al.*, 2019a] Jia Wang, Tong Sun, Benyuan Liu, Yu Cao, and Hongwei Zhu. Clvsa: A convolutional lstm based variational sequence-to-sequence model with attention for predicting trends of financial markets. In *Proc. of IJCAI*, 2019.

[Wang *et al.*, 2019b] Jingyuan Wang, Yang Zhang, Ke Tang, Junjie Wu, and Zhang Xiong. Alphastock: A buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In *Proc. of KDD*, 2019.

[Wu *et al.*, 2021] Xueqing Wu, Lewen Wang, Yingce Xia, Weiqing Liu, Lijun Wu, Shufang Xie, Tao Qin, and Tie-Yan Liu. Temporally correlated task scheduling for sequence learning. In *Proc. of ICML*, 2021.

[Wu *et al.*, 2022] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2022.

[Xia *et al.*, 2021] Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. Sample selection with uncertainty of losses for learning with noisy labels. In *International Conference on Learning Representations*, 2021.

[Xing *et al.*, 2002] Eric Xing, Michael Jordan, Stuart J Russell, and Andrew Ng. Distance metric learning with application to clustering with side-information. In *Proc. of NeurIPS*, 2002.

[Xu *et al.*, 2020] Jin Xu, Jingbo Zhou, Yongpo Jia, Jian Li, and Xiong Hui. An adaptive master-slave regularized model for unexpected revenue prediction enhanced with alternative data. In *Proc. of ICDE*, 2020.

[Yang *et al.*, 2020] Xiao Yang, Weiqing Liu, Dong Zhou, Jiang Bian, and Tie-Yan Liu. Qlib: An ai-oriented quantitative investment platform. *arXiv:2009.11189*, 2020.

[Zhang *et al.*, 2017] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. Stock price prediction via discovering multi-frequency trading patterns. In *Proc. of KDD*, 2017.

[Zhang *et al.*, 2020] Chuheng Zhang, Yuanqi Li, Xi Chen, Yifei Jin, Pingzhong Tang, and Jian Li. Doubleensemble: A new ensemble method based on sample reweighting and feature selection for financial data analysis. In *Proc. of ICDM*, 2020.