Multi-Granularity Residual Learning with Confidence Estimation for Time Series Prediction

Min Hou[1], Chang Xu[2], Zhi Li[1], Yang Liu[1], Weiqing Liu[2], Enhong Chen[1], Jiang Bian[2]
[1]School of Data Science, University of Science and Technology of China, Hefei, China
[2]Microsoft Research, Beijing, China
{minho, zhili03, liuyang0}@mail.ustc.edu.cn, cheneh@ustc.edu.cn,
{chanx, weiqing.liu, jiang.bian}@microsoft.com

ABSTRACT

Time-series prediction is of high practical value in a wide range of applications such as econometrics and meteorology, where the data are commonly formed by temporal patterns. Most prior works ignore the diversity of dynamic pattern frequency, *i.e.,* different granularities, suffering from insufficient information exploitation. Thus, multi-granularity learning is still under-explored for time-series prediction. In this paper, we propose a **M**ulti-granularity **R**esidual **L**earning **F**ramework (MRLF) for more effective time series prediction. For a given time series, intuitively, there are more or less semantic overlaps and validity differences among its representations of different granularities. Due to the information redundancy, straightforward methods that leverage multi-granularity data, such as concatenation or ensemble, can easily lead to the model being dominated by the redundant coarse-grained trend information. Therefore, we design a novel residual learning net to model the prior knowledge of the fine-grained data's distribution through the coarse-grained one. Then, by calculating the residual between multi-granularity data, the redundant information be removed. Furthermore, to alleviate the side effect of validity differences, we introduce a self-supervised objective for confidence estimation, which delivers more effective optimization without the requirement of additional annotation efforts. Extensive experiments on the real-world datasets indicate that multi-granular information significantly improves the time series prediction performance, and our model is superior in capturing such information.

CCS CONCEPTS

• **Applied computing** → **Forecasting**.

KEYWORDS

time-series prediction, multi-granularity learning

ACM Reference Format:
Min Hou[1], Chang Xu[2], Zhi Li[1], Yang Liu[1], Weiqing Liu[2], Enhong Chen[1], Jiang Bian[2]. 2022. Multi-Granularity Residual Learning with Confidence Estimation for Time Series Prediction. In *Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512056

WWW '22, April 25–29, 2022, Virtual Event, Lyon, France

ACM ISBN 978-1-4503-9096-5/22/04…$15.00
https://doi.org/10.1145/3485447.3512056

1 INTRODUCTION

Time series prediction, which contributes to accurately forecasting future values of time series given the past, has attracted great attention from both academia and industry. A good prediction of time series trends has a broad impact on many aspects of modern society such as financial market prediction [12, 49], climate forecasting [34], electricity demand estimation [43], and health monitoring [44].

Along this line, most existing works [14, 39, 51] are single-granularity oriented, which are developed on a specific granularity of time series data, with the goal of predicting the labels at the same level. For example, previous works utilize users' daily frequency electricity consumption data for daily consumption forecasting and daily price-volume data for daily stock trend prediction. However, multi-granularity data [40], which usually provides complementary detailed information that is not covered in the original granularity data, is vital for making accurate predictions. For instance, when making investment decisions, mature analysts usually study the state of stocks extensively in different granularities, such as weekly, daily, and minute levels. Since long-term (*i.e.*, coarse-grained) features reflect overall trends, while short-term (*i.e.*, fine-grained) features indicate subtle changes in small time windows, both are crucial to prediction quality in a specific task. This motivates us to investigate how to leverage multi-granularity data to enhance the time series prediction. Nevertheless, there are still some unique challenges in designing an effective solution to integrate multi-granularity data into time-series modeling.

On one hand, there is severe information redundancy between different granularity data, ignoring unique information in a certain granularity. Fine-grained data usually cover the information of coarse-grained data, since coarse-grained data are often obtained by the aggregation of the corresponding fine-grained data. Taking stock trading as an example, the daily *highest prices* are calculated from the highest prices of minute frequency data within the same day. Due to the information redundancy, straightforward methods that leverage multi-granularity data, such as concatenation or ensemble, can easily lead to the model being dominated by the redundant coarse-grained trend information. However, removing redundant coarse-grained information from the fine-grained one is not trivial due to the heterogeneity of multi-granularity data. Therefore, how to utilize granularity-specific information while avoiding the semantic overlap between coarse-grained and fine-grained data is still a great challenge.

figure 9.pdf

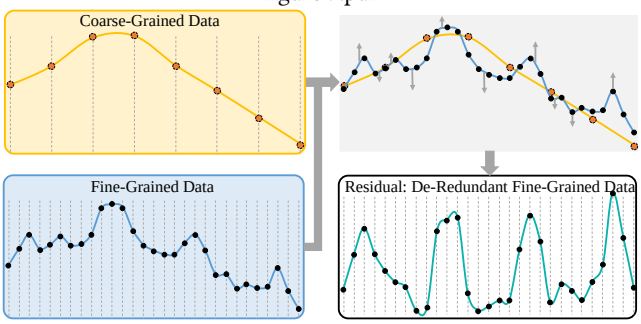


Figure 1: The change patterns in fine-grained data are dominated by the trend information of coarse-grained data, making it becomes difficult to be captured straightforwardly. By calculating the residual between fine and coarse-grained data, the redundant information be removed, and the unique fine-grained patterns can be easily captured.

On the other hand, the validity and effectiveness of different granularity data usually change over time. Intuitively, different granularity information has different effects on the final prediction of the target time granularity. For example, for daily frequency electricity consumption forecasting, daily-frequency user electricity consumption data usually plays an important role. However, at the peak of power consumption in the summer, a user's fine-grained patterns, like whether using high-power appliances such as air conditioners, may have a significant impact on the power consumption. At this time, hourly frequency data may be much more effective than daily data when forecasting the daily electricity consumption. Therefore, due to the change of effectiveness in different granularity data, we need to judge whether specific granularity data has enough confidence for the final predictions at the time.

To handle the above issues, in this paper, we propose Multi-granularity Residual Learning Framework (MRLF) for time-series prediction towards an effective exploration of multi-granularity patterns. Specifically, 1) we first propose a cross-granularity residual learning net, which contains multiple blocks with similar structures, and each block is responsible for learning information of a specific granularity. In order to remove the redundancy and ensure that the input of each block is unique to a certain granularity, we introduce a novel residual design between each block. As shown in Figure 1, we take two granularities as an example. First, the coarse and fine-grained features are obtained from raw data. We observe that the change patterns of fine-grained data are dominated by the trend information of coarse-grained data, making it becomes difficult to be captured straightforwardly. Nevertheless, given the coarse-grained data, we can get prior knowledge of the fine-grained data's distribution, and we predict the possible fine-grained data given coarse-grained data. Then, by calculating the residual between multi-granularity data, the redundant information be removed, and the unique fine-grained change patterns can be easily captured. 2) Then, to judge whether specific granularity data has enough confidence for the final predictions, we develop a Multi-Granularity Confidence Estimator. Concretely, we train a discriminator by constructing a self-supervised objective. The discriminator measures whether the information of each granularity is effective by the trend similarity between the history and the present status.

In summary, the main contributions of this work include:

1) In this paper, we present a focused study on multi-granularity data for time series prediction. To the best of our knowledge, this is among the first few studies to investigate how to dynamically fuse multi-granularity data for time series prediction task.
2) We propose a novel cross-granularity residual learning net to remove the semantic overlap inherent in multi-granularity data for better information exploitation, and further design a self-supervised confidence estimator to judge the dynamic effectiveness of each granularity.
3) Extensive experiment results on real-world datasets clearly validate the superiority of our framework in prediction accuracy compared with the state-of-the-arts.

2 PRELIMINARIES

In this section, we first introduce the related work of time series prediction. And then, we formally formulate the single-granularity and multi-granularity learning problems.

2.1 Related Work

As a classical research topic, time series prediction has been intensively studied by researchers over the past several decades. It has historically been a key area of both academia and industry, forming an integral part of applications in topics such as climate modeling [34, 37], biological sciences [15] and medicine [44], as well as commercial decision making in retail [8, 31, 50] and finance [12, 22, 52, 53] to name a few. Existing methods for time series prediction can be roughly grouped into two categories: conventional methods and deep learning-based methods.

2.1.1 Conventional Methods. Conventional methods focus on parametric models informed by domain expertise, such as Auto Regression (AR) [5], Auto Regressive Integrated Moving Average (ARIMA) [19], exponential smoothing [24], or structural time series models [16]. Most traditional methods can only learn linear relationships among different timesteps, which has an inherent deficiency in fitting many real-world time-series data that are highly nonlinear. To model non-linear relationships, some variants of the Auto Regressive model are used, such as LRidge [21], LSVR [47] and Gaussian Process [41]. However, they assume certain distribution or function form of time series and fail to capture different forms of non-linearity [23].

2.1.2 Deep Learning-Based Methods. Leveraging the ability to flexibly model various non-linear relationships, deep learning-based methods adopt deep neural networks to capture shared information across related time series for accurate forecasting. These methods provide a means to learn temporal dynamics in a data-driven manner. For example, Recurrent Neural Networks (RNNs) and its variants such as Long Short-Term Memory Networks (LSTMs) [20, 30] and Gated Recurrent Unit (GRU) [10] have become popular due to their automatic feature extraction abilities, complex patterns,

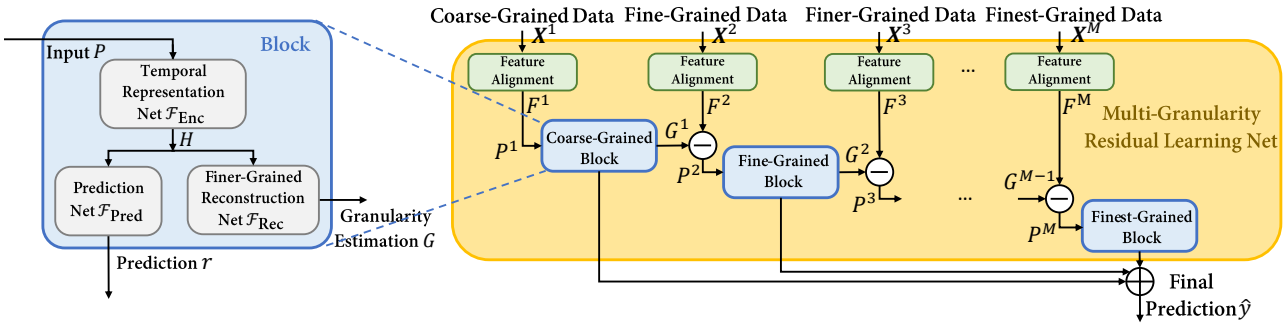


Figure 2: Illustration of the Multi-Granularity Residual Learning.

and long term dependencies modeling. To predict more accurately, complex structures such as recurrent-skip layer (LSTNet-S), temporal attention layer (LSTNet-A) [30], and a novel temporal pattern attention mechanism (TPA) [42] have been proposed. Convolutional Neural Networks (CNNs) [4, 11] are also used to enhance feature extraction ability. The authors in [28] solve the time series problem by using the deep-learning-based generative model Generative Adversarial Neural Networks (GANS). Recently, the well-known self-attention-based Transformer [27, 48, 54] have been proposed for sequence modeling and has achieved great success. Most previous studies are single granularity oriented, which results in information loss from a multi-granularity perspective. In fact, a robust time series prediction model should be able to capture temporal patterns at different time granularities. Although there exist a few works [32, 53] using "multi-scale" information for time series analysis, we clarify that this type of information is different from what we refer to by "multi-granularity". In more technical terms [40], the concept of "scale" is associated with a subdivision of the frequency spectrum in bands. "Multi-granularity" methods (also called multi-resolution methods) address the challenge of handling the coexistence of data with different granularities or levels of aggregation, i.e., with different resolutions/ different time level aggregation of statistics or features.

2.2 Problem Formulation

Most existing time series prediction methods use historical features of single granularity to predict future trends, *i.e.*, **single granularity learning**, where the input and output granularity are the same. The prediction model learns a function $\hat{y} = \mathcal{F}_{\Theta}(\boldsymbol{X})$,where $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T] \in \mathbb{R}^{D \times T}$ represents the features in the lag of past T time-steps with dimension D. The time interval between time-steps is λ.

Fine-grained data contains rich detailed information that the original-grained data don't have, but we rarely input only the most fine-grained data to the model. When the input granularity is inconsistent with the prediction granularity, it is often difficult for the model to directly capture the law of prediction granularity. Therefore, we propose not only using the data with the same granularity as labels but also multiple using finer-grained data.

This paper focuses on time series prediction using multi-granularity features, *i.e.*, **multi-granularity learning**. We assume the coarsest granularity of the feature is consistent with the label. Formally, the model learns $\hat{y} = \mathcal{F}_{\Theta}\left(\boldsymbol{X}^1, \cdots, \boldsymbol{X}^M\right)$, which maps the historical multi-granularity data to the future trend label space. For each granularity m, the data $\boldsymbol{X}^m = \left[\boldsymbol{x}_1^m, \cdots, \boldsymbol{x}_T^m\right] \in \mathbb{R}^{D \times K^m \times T}$ represents features in the lag of past T time-steps. At each time-step t, $\boldsymbol{x}_t^m \in \mathbb{R}^{D \times K^m}$ is composed of features from $K^m \in \mathbb{R}$ equally divided time periods within one time step, whose time interval is λ / K^m. Each time-slot contains D features.

3 METHODOLOGY

In this section, we first introduce the overall architecture of the Multi-granularity Residual Learning Framework (MRLF) and present the design of each component in detail. Then we elaborate the Multi-Granlarity Confidence Estimator module and discuss the learning algorithm for MRLF.

3.1 Multi-Granularity Residual Learning

Given the multi-granularity data $\{\boldsymbol{X}^1, \cdots, \boldsymbol{X}^M\}$ that $\boldsymbol{X}^1$ to $\boldsymbol{X}^M$ represent coarse to fine features, *Multi-Granularity Residual Learning* process explores informative cues of time-series future trends hidden in different granularity features. As illustrated in Figure 2, to fully exploit the information of every granularity, the process contains several blocks (the blue rectangles) with similar structures, and each block is responsible for learning information of only one granularity. Since the existence of severe redundancy between different granularity data may cause the model to be dominant by the redundant coarse-grained trend information, we propose stacking the blocks in a cascaded way from coarse to fine, and design a novel cross-granularity residual learning method to ensure the input of each block only contains unique information of a specific granularity. In this section, we first introduce the components of the net. Then we elaborate on how to stack them in by cross-granularity residual learning.

3.1.1 Feature Alignment. Since the dimensions of different raw granularity data are inconsistent, we align them to the same space to facilitate subsequent residual learning operations, as depicted in the green part of Figure 2. Specifically, we made a simple linear transformation $\mathcal{F}_{\text{Linear}}^m$ to the input $\boldsymbol{X}^m \in \mathbb{R}^{D \times K^m \times T}$ which is the raw data of granularity m in the lag of T timesteps. The aligned features are denoted as $\boldsymbol{F}^m \in \mathbb{R}^{D \times K \times T}$. We formulate the feature alignment process as:

$$\boldsymbol{F}^m = \mathcal{F}_{\text{Linear}}^m(\boldsymbol{X}^m). \tag{1}$$

3.1.2 Basic Block. The function of basic blocks is to learn granularity-specific knowledge. Its architecture is depicted in the left part of Figure 2. We describe the operation of m-th block in this subsection in detail. The m-th block accepts its respective input $\boldsymbol{P}^m$ and has two outputs, r^m and $\boldsymbol{G}^m$. Input $\boldsymbol{P}^m$ represents the de-redundant feature embedding of granularity m. We elaborate on the de-redundant process in the next subsection. The output r^m aims to perform prediction according to the current granularity m. The output $\boldsymbol{G}^m$ approximates the block's best estimate of the next granularity data $\boldsymbol{F}^{m+1}$, which indicates the information already learned by the model and with the ultimate goal of helping the downstream blocks by removing redundant components of their input that are not helpful for forecasting.

Internally, the basic block consists of several components. The first part is **Temporal Representation Net** $\mathcal{F}_{\text{Enc}}$, which further encode the sequential features $\boldsymbol{P}^m$ as:

$$\boldsymbol{H}^m = \mathcal{F}_{\text{Enc}}(\boldsymbol{P}^m). \tag{2}$$

The network architecture of $\mathcal{F}_{\text{Enc}}$ can be flexible. To capture the temporal characteristics of time series data, here we adopt a 2-layer GRU [9]. The second part is **Prediction Net** $\mathcal{F}_{\text{Pred}}$, which takes the feature embedding $\boldsymbol{H}^m$ as input, and output the prediction result r^m of the specific granularity as

$$r^m = \mathcal{F}_{\text{Pred}}(\boldsymbol{H}^m). \tag{3}$$

Another part, **Finer-Grained Reconstruction Net**, is donated as $\mathcal{F}_{\text{Rec}}$. It takes the feature embedding $\boldsymbol{H}^m$ as input, and output a finer-granularity estimation $\boldsymbol{G}^m$ of next level's coarse-grained features $\boldsymbol{F}^{m+1}$. We formulate it as follow:

$$\boldsymbol{G}^m = \mathcal{F}_{\text{Rec}}(\boldsymbol{H}^m). \tag{4}$$

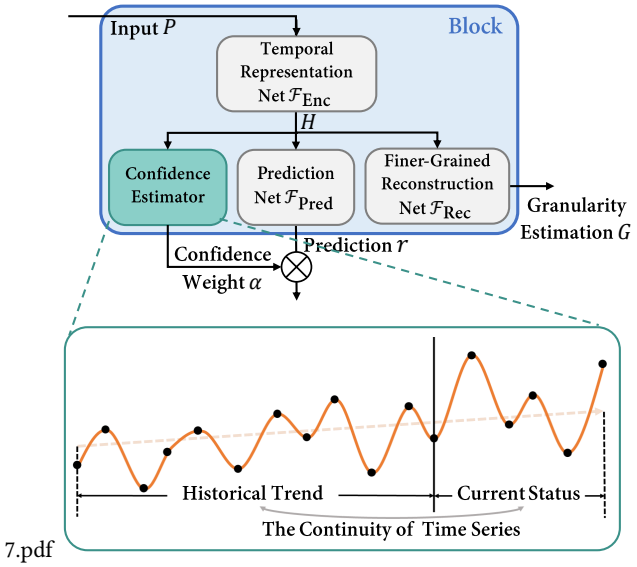
Here we use two fully-connected layer to model $\mathcal{F}_{\text{Pred}}$ and $\mathcal{F}_{\text{Rec}}$.

3.1.3 Cross-Granularity Residual Stacking. In this paper, we address the information redundancy problem by introducing a novel residual learning approach. Inspired by previous residual learning works [17, 18, 36] that explicitly let stacked layers fit residual mapping $\mathcal{F}(x) := \mathcal{H}(x) - x$, instead of the desired underlying mapping $\mathcal{H}(x)$, we propose stacking the multi-granularity blocks by residual learning fashion with the goal of removing the redundancy.

As illustrated in the yellow rectangle in Figure 2, in a special case of the very first block, its input is the coarsest-grained feature embedding, *i.e.*, $\boldsymbol{P}^1 := \boldsymbol{F}^1$. We hypothesis that the coarse-grained data itself has prior information about the distribution of fine-grained data. Therefore we simulate redundant information by using fine-grained data reconstructed from coarse-grained data. Specifically, we let the granularity estimation $\boldsymbol{G}^m$ reconstruct the feature of granularity $m+1$. Then the block $m+1$ removes the portion of the redundant prior signal that the previous block can approximate well, making the downstream blocks focus on learning the finer-granularity-specific knowledge. We formulate cross-granularity residual learning process as:

$$\boldsymbol{P}^m = \begin{cases} \boldsymbol{F}^m & \text{if } m = 1, \\ \boldsymbol{F}^m - \boldsymbol{G}^{m-1} & \text{otherwise,} \end{cases} \tag{5}$$

where $\boldsymbol{P}^m$ represents the de-redundant input of block m. In order to learn trend information from coarser-grained data, while also allowing the finer-grained information to retain uniqueness after



7.pdf

Figure 3: The underlying temporal patterns in time series often change with time, the effectiveness of different granularity data for future trend prediction is also dynamic. Therefore, we propose estimating the confidence of different granularity data as a learnable process through the continuity of different granularity data itself over time.

the residual structure, it is necessary to make the coarser-grained information characterize of the original finer-grained information as much as possible:

$$\mathcal{L}_{\text{Rec}} = \sum_{m=1,2,\ldots,M} \left\| \boldsymbol{F}^m - \boldsymbol{G}^{m-1} \right\|_F^2, \tag{6}$$

$\|\cdot\|_F$ denotes the Frobenius norm. In order not to affect the finer-grained information extraction process, when optimizing Equation (6), we fix the process of extracting fine-grained information $\boldsymbol{F}^m$ and only optimize the $\boldsymbol{G}^{m-1}$.

The output r^m of each block represents the specific time-series prediction of the target granularity. In order to comprehensively consider the information of each granularity to produce the final prediction result, we can simply average all the results to obtain the final prediction $\hat{y}$:

$$\hat{y} = \frac{1}{m} \sum_m r^m. \tag{7}$$

3.1.4 Basic Model Optimization. Combining the MSE loss of prediction task, the reconstruction loss in the cross-granularity residual learning process, and the regularization term, we reach the following loss function:

$$\mathcal{L} = \sum_{s=1}^{S} \left\| y^s - \hat{y}^s \right\|^2 + \lambda_1 \sum_{s=1}^{S} \mathcal{L}_{\text{Rec}} + \frac{\lambda_\theta}{2} \|\Theta\|_F^2, \tag{8}$$

where λ_1 and λ_θ are the hyper-parameters to balance different losses. $\|\Theta\|_F^2$ is the L2-regularizer, S is the total number of time series records. We use Adam algorithm [25] in mini-batches to update our model parameters with the backpropagation.

3.2 Self-Supervised Confidence Estimation

Since the underlying temporal patterns in time series often change with time, the effectiveness of different granularity data for future trend prediction is also dynamic. Therefore, we propose estimating the confidence of different granularity data as a learnable process to enhance the basic framework. Intuitively, the more continuous the data trend on a certain granularity in the near future, the higher the confidence of future prediction using this granularity. Thus, we can estimate this effectiveness through the continuity of different granularity data itself over time. Since there is no direct ground-truth label, we construct additional self-supervised signals relying on the inherent character of time series data. In this subsection, we elaborate the confidence estimation module and the enhanced model optimization process.

3.2.1 Confidence Estimator. As shown in figure 3, the confidence estimator is placed at the basic block introduced in the previous section. It outputs a score estimating the confidence of the prediction for a specific granularity. The confidence weight α is set to 1 for all the blocks when we choose to disable the confidence estimator. In this work, we take advantage of mutual information [6, 29, 33, 38, 46] to measure the consistence of each granular data. If the historical trend of a certain granular data has a relatively large mutual information with the current state, it indicates that the historical information of the granular data may be more helpful to future prediction results. Mutual information is a fundamental quantity for measuring the relationship between random variables. In contrast to correlation, mutual information captures non-linear statistical dependencies between variables. and thus can act as a measure of true dependence [3, 26]. Given two random variables X and Y, it can be understood as how much knowing X reduces the uncertainty in Y or vice verse.

At time-step t, we denote the representation of a certain granular data m as $\boldsymbol{h}_m^t$. To extract the historical trend information buried in previous $t-1$ time-steps, we apply an autoregressive model $AR(\cdot)$ to summarize all $\boldsymbol{h}_m^{<t} = [\boldsymbol{h}_m^1, \boldsymbol{h}_m^2, \cdots, \boldsymbol{h}_m^{t-1}]$ in the latent space and produce a trend latent representation $\boldsymbol{c}_m^t = AR(\boldsymbol{h}_m^{<t})$. Since the precise value of MI is difficult to compute, we utilize neural estimators to maximize the lower-bound of MI instead [35]:

$$\mathrm{MI}\left(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t\right) \geq \log(N) - \mathcal{L}_{\mathrm{N}}^C, \tag{9}$$

where $\mathcal{L}_{\mathrm{N}}^C$ is the contrastive loss function, that is defined as:

$$\mathcal{L}_{\mathrm{N}}^C = -\mathop{\mathrm{E}}_{\mathcal{P}_l}\left[\log \frac{D_\omega\left(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t\right)}{D_\omega\left(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t\right) + \sum\limits_{\tilde{\boldsymbol{h}}_m^l \in \hat{\mathbf{H}}_m^t} D_\omega\left(\tilde{\boldsymbol{h}}_m^t, \boldsymbol{c}_m^t\right)}\right], \tag{10}$$

where $\mathcal{P}_l$ represents the joint distribution of the historical trend and the current state, i.e., $\left(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t\right) \sim \mathcal{P}_l$. $\tilde{\boldsymbol{h}}_m^l$ denotes the negative sample randomly sampled from the marginal distribution of current state from other samples within one mini-batch, forming a set of $N-1$ elements, denoted by $\hat{\mathbf{H}}_m^t$. $D_\omega(\cdot,\cdot)$ is the discriminator parameterized by ω, we define it in the form of log-bilinear:

$$D_\omega\left(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t\right) = \exp\left(\boldsymbol{h}_m^{t\top} \cdot \boldsymbol{W} \cdot \boldsymbol{c}_m^t\right), \tag{11}$$

where $\boldsymbol{W}$ is a learnable linear transformation matrix. The loss in Equation (10) is the categorical cross-entropy of classifying the positive sample correctly.

After optimizing the Equation (10), we can prove that the optimal value for $D_\omega\left(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t\right)$ in Equation (11) is proportional to the density ratio and this is independent of the choice of the number of negative samples $N-1$. We prove that as follow, remind that Equation (9) is the categorical cross-entropy loss of classifying the positive sample correctly, with $\frac{D_\omega}{\sum_{\mathrm{M}} D_\omega}$ being the prediction model where set $\mathrm{M} = \{\boldsymbol{h}_m^t\} \cup \hat{\mathbf{H}}_m^t$ contains 1 positive sample and $N-1$ negative samples. Let us rewrite the optimal probability for $\mathcal{L}_{\mathrm{N}}^C$ as $p\left(d = i \mid \mathrm{M}, \boldsymbol{c}_m^t\right)$ with $[d = i]$ representing sample i is the 'positive' sample $\boldsymbol{h}_m^{t,(i)}$. While others $d \neq i$ represent negative samples $\boldsymbol{h}_m^{t,(d)} \in \hat{\mathbf{H}}_m^t$. The probability that sample d is positive can be derived as follows:

$$\begin{aligned} p(d = i \mid \mathrm{M}, \boldsymbol{c}_m^t) &= \frac{p(\boldsymbol{h}_m^{t,(i)}, \boldsymbol{c}_m^t) \prod_{k \neq i} p(\boldsymbol{h}_m^{t,(k)})}{\sum_{j=1}^N p(\boldsymbol{h}_m^{t,(j)}, \boldsymbol{c}_m^t) \prod_{k \neq j} p(\boldsymbol{h}_m^{t,(k)})} \\ &= \frac{\frac{p(\boldsymbol{h}_m^{t,(i)}, \boldsymbol{c}_m^t)}{p(\boldsymbol{h}_m^{t,(i)}) p(\boldsymbol{c}_m^t)}}{\sum_{j=1}^N \frac{p(\boldsymbol{h}_m^{t,(j)}, \boldsymbol{c}_m^t)}{p(\boldsymbol{h}_m^{t,(j)}) p(\boldsymbol{c}_m^t)}}. \end{aligned} \tag{12}$$

Hence, the optimal value $D_\omega^{P*}(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t)$ in Equation (10) is proportional to the density ratio $\frac{p(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t)}{p(\boldsymbol{h}_m^t) p(\boldsymbol{c}_m^t)}$, and this is independent of the choice of the number of negative samples $N-1$. Therefore, we conclude that:

$$D_\omega\left(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t\right) \propto \frac{p\left(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t\right)}{p\left(\boldsymbol{h}_m^t\right) p\left(\boldsymbol{c}_m^t\right)}. \tag{13}$$

Since the *Pointwise Mutual Information* (PMI) of a pair of samples is defined as: $\mathrm{PMI}(x, y) = \log \frac{p(x,y)}{p(x)p(y)}$, where $x \in X$ and $y \in Y$. Consequently, the discriminative scores computed by the optimal discriminators D_ω^* are:

$$\alpha_m^t = D_\omega^*\left(\boldsymbol{h}_m^t, \boldsymbol{c}_m^t\right), \tag{14}$$

Then we utilize a *softmax* layer to normalize the PMI value of each granular information:

$$\alpha_m^t = \frac{\alpha_m^t}{\sum_i \alpha_i^t}, \tag{15}$$

where α_m^t is the final confidence score of granular m. We rewrite the final prediction (Equation (7)) as confidence weighted sum of each extractor:

$$\hat{y} = \frac{1}{m} \sum_m \alpha_m^t \cdot r_m. \tag{16}$$

3.2.2 Model Optimization. Combining the MSE loss of prediction task, the reconstruction loss in the cross-granularity residual learning process, the contrastive loss to train confidence estimator, and the regularization term, we rewrite the Equation (8) and reach the following complete loss function:

$$\mathcal{L} = \sum_{s=1}^S \|y^s - \hat{y}^s\|^2 + \lambda_1 \sum_{s=1}^S \mathcal{L}_{\mathrm{Rec}} + \lambda_2 \sum_{s=1}^S \sum_{m=1}^M \sum_{t=1}^T \mathcal{L}_{\mathrm{N}}^C + \frac{\lambda_\theta}{2} \|\Theta\|_F^2, \tag{17}$$

Table 1: Performance of MRLF and Comparison Methods on Electricity Data and Stock Data.

Method	Electricity			Stock		
	CORR↑	RMSE↓	MAE↓	CORR↑	RMSE↓	MAE↓
GRU	0.9763	0.1281	0.0377	0.0759	2.3429	2.1634
LSTM	0.9582	0.1761	0.0837	0.0747	2.4507	2.2274
Transformer	0.9725	0.1410	0.0551	0.0639	3.6961	3.6304
DeepAR	0.9634	0.1655	0.0801	0.0751	2.4075	2.1923
Informer	0.9788	0.1276	0.0371	0.0868	2.2753	2.1206
SFM	0.9669	0.2571	0.1579	0.0857	2.8251	2.6348
ALSTM	0.9661	0.2606	0.1569	0.0736	2.8251	2.4941
ADV-ALSTM	0.9750	0.2237	0.1331	0.0790	3.1770	2.8754
Coarse-Grained RNN	0.9728	0.1376	0.0544	0.0759	2.3429	2.1634
Fine-Grained RNN	0.9751	0.1327	0.0498	0.0873	2.2680	2.1067
Multi-Grained RNN	0.9782	0.1297	0.0473	0.0885	2.0784	1.8209
Ensemble	0.9753	0.1307	0.0494	0.0894	3.6913	3.5862
MRLF (attention)	0.9792	0.1266	0.0358	0.0891	2.2898	1.6511
MRLF (w/o CE)	0.9772	0.1278	0.0365	0.0853	2.6085	1.9439
MRLF	**0.9852**	**0.1118**	**0.0308**	**0.0953**	**0.7262**	**0.6919**

where λ_1, λ_2, λ_θ are the hyper-parameters to balance different losses. $\|\Theta\|_F^2$ is the L2-regularizer, S is the total number of time series records. We use Adam algorithm [25] in mini-batches to update our model parameters with the backpropagation.

4 EXPERIMENTS

In this section, we conduct experiments on real-world datasets to verify the feasibility of our proposed model. We then analyze the experiment results and demonstrate the precision promotion by comparing it with various baselines.

4.1 Experiment Settings

4.1.1 Datasets. We extensively perform experiments on two real-world datasets.

- **Electricity Data.** The UCI electricity dataset[1] collects the electricity consumption (kWh) every 15 minutes of 321 clients from 2012 to 2014. The train/val/test is 24/6/6 months. We aim to predict the daily consumption of each client. The granularity of input features is 1 day, 12 hours, 4 hours, 1 hour, and 15 minutes.
- **Stock Data.** We use the quantitative investment platform Qlib[2] to collect stock sequences consisting of 1-min high-frequency statistics over the constituent stocks from the major stock indice CSI300. The datasets range from Feb. 16, 2007 to Jan. 1, 2020 with 908,606 records of 749 stocks. We split the sequences by time, the train/val/test is 94/12/12 months. 6 commonly used statistics are extracted as features, including the highest price, the opening price, the lowest price, the closing price, volume-weighted average price, and trading volume. The data are adjusted for dividends and splits, and normalized by the Z-Score method. Following [7, 13, 52], we aim to predict the daily return ratio of a stock which is formalized as $y = p_{T+2}/p_{T+1} - 1$, where p_t represents the volume-weighted average price of the stock at day t. The granularity of input features is 1 day, 1 hour, 15 mins, 5 mins, and 1 min.

[1] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams 20112014
[2] https://github.com/microsoft/qlib

4.1.2 Comparison Methods. The competitive baselines we compared can be categorized into four groups.

- The first group consists of general time series forecasting models, including **GRU**, **LSTM**, **Transformer** [48], **DeepAR** [1] and **Informer** [54]. **DeepAR** produces accurate probabilistic forecasts by constructing a powerful auto-regressive method. **Informer** is an efficient transformer-based model and is the SOTA on the Electricity Data.
- The second group consists of current top systems for stock trend prediction. **SFM** [53] aims to capture trading patterns from investors with different trading modes inspired by Fourier Transform. **ALSTM** [39] contains a temporal attentive aggregation layer based on normal LSTM. **Adv-ALSTM** [14] is a variant of ALSTM with adversarial training method, which is claimed to be the state-of-the-art method for daily trend prediction.
- The third group contains variants of our model using different granularities of data. **Coarse-grained RNN** and **Fine-grained RNN** use only coarsest-grained data or finest-grained data, respectively. The input of **Multi-grained RNN** is the concatenation of two granularity data. **Ensemble** stands for the ensemble result for five independent training models with different granularity data. The basic architectures of these methods are consistent with MRLF.
- The fourth group contains two ablation counterparts of MRLF. **MRLF (w/o CE)** and **MRLF (attention)** represents MRLF without the confidence estimation mechanism and MRLF replace Confidence Estimator with a canonical soft attention mechanism [2], respectively. For **MRLF (attention)**,

Table 2: Influence of granularity choice on Electricity Data and Stock Data.

Granularity Choices	Electricity			Stock		
	CORR↑	RMSE↓	MAE↓	CORR↑	RMSE↓	MAE↓
g1	0.9728	0.2332	0.1376	0.0729	2.8312	2.5545
g2	0.9738	0.2289	0.1355	0.0857	1.8422	1.6696
g3	0.9749	0.2241	0.1330	0.0859	2.3426	2.2971
g4	0.9749	0.2243	0.1332	0.0884	4.7757	4.6461
g5	0.9751	0.2232	0.1327	0.0841	4.4289	4.1074
g1+g2 (w/o CE)	0.9759	0.2195	0.1302	0.0753	7.9910	6.0631
g1+g2	0.9789	0.2149	0.1287	0.0721	6.1520	4.5026
g1+g3 (w/o CE)	0.9761	0.2186	0.1299	0.0757	3.4731	2.5433
g1+g3	0.9793	0.2057	0.1282	0.0744	3.1533	2.3123
g1+g4 (w/o CE)	0.9774	0.2184	0.1281	0.0824	5.8018	4.4477
g1+g4	0.9801	0.2012	0.1216	0.0806	4.9504	3.6768
g1+g5 (w/o CE)	0.9782	0.2175	0.1297	0.0852	5.0155	3.8347
g1+g5	0.9820	0.1794	0.1135	0.0864	1.3564	1.1276
g1+g3+g5 (w/o CE)	0.9788	0.1889	0.1263	0.0822	2.1274	1.3492
g1+g3+g5	0.9844	0.1786	0.1124	0.0834	0.8846	0.7403
g1+g2+g3+g4+g5 (w/o CE)	0.9772	0.1910	0.1278	0.0853	2.6085	1.9439
g1+g2+g3+g4+g5	**0.9852**	**0.1755**	**0.1118**	**0.0953**	**0.7262**	**0.6919**

query is the hidden state of $\mathcal{F}_{\text{Enc}}(\boldsymbol{P}^1)$ at time-step $T-1$, keys are $\boldsymbol{P}^1$ to $\boldsymbol{P}^M$.

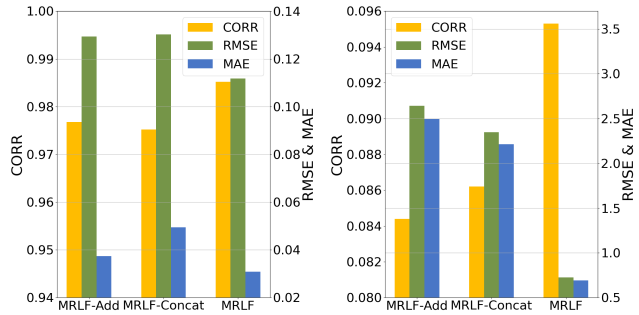
4.1.3 Implementation Details. **1) Hyperparameters:** For the autoregressive model, we use a 2-layer GRU. All the hidden size are set to 64. For all the methods, we optimize them by mini-batch Adam until convergence and tune the hyper-parameters via grid search on the validation set with the learning rate selected from [10^{-4}, 10^{-3}, 10^{-2}]. We set the λ_1 and λ_2 in Equation (17) to 1. The coefficient of L_2 regularization λ_3 is tuned amongst [10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]. For a fair comparison, RNN backbones in ALSTM and Adv-ALSTM are searched from the traditional RNN and its variants LSTM and GRU, and the number of layers is chosen from [1, 2]. The Transformer we compared has 2 encoder layers, and the number of heads is searched from [2, 5]. We tune the hyper-parameters of the baseline methods both from the values listed in their source code and similar range as used for the proposed method, and we report the best performance. **2) Platform:** All the models were trained/tested on a single Nvidia RTX 2080Ti 11GB GPU. **3) Evaluation Metrics:** Following some previous works [45, 51, 54], we adopt three commonly used evaluation metrics, including Pearson correlation coefficients (CORR), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE). For CORR, higher values are better. For RMSE and MSE, lower values are better.

4.2 Experiment Results

4.2.1 Comparison with Baselines. To demonstrate the effectiveness of our proposed model, we compare MRLF with other state-of-the-art methods of both general time series prediction and stock trend prediction, as well as model variants using different granularity data. The comparison results are shown in Table 1. Overall, our proposed MRLF substantially achieves the best results on two datasets and three evaluation metrics. We also make the following observations: Multi-Granined RNN achieves better performance than all single-granularity models, which verifies that introducing multi-granularity features can boost the performance. Furthermore. MRLF significantly outperforms the Multi-Granularity RNN, demonstrating that our model integrates the features better than baselines. We also observe that both MRLF (attention) and MRLF (w/o CE) surpass Multi-Grained RNN, and MRLF shows superior performance over the two variants, verifying the validity of different granularity is dynamic and further justify our self-supervised confidence estimator can better model the dynamic.

4.2.2 Effectiveness of Residual Learning. To study the effectiveness of the cross-granularity residual learning net, we implement two MRLF variants which replace the cross-granularity residual learning with the commonly used feature fusion process: MRLF-Add and MRLF-Concat. The two variants take the feature addition $\{\boldsymbol{F}^1, \boldsymbol{F}^1 + \boldsymbol{F}^2, \cdots, \boldsymbol{F}^1 + \cdots + \boldsymbol{F}^M\}$ and the concatenation $\{\boldsymbol{F}^1, [\boldsymbol{F}^1\boldsymbol{F}^2], \cdots, [\boldsymbol{F}^1 \cdots \boldsymbol{F}^M]\}$ as each block's input, respectively. The result are presented in Figure 4. We find that the residual learning net is consistently better than MRLF-Add and MRLF-Concat. This demonstrates that the proposed cross-granularity residual learning net indeed helps each block learn granularity-specific information and therefore boosting the performance of time series prediction.

4.2.3 Influence of Granularity Choice. We study how granularity choices affect model performance. The results are presented in Table 2. $g1, \cdots, g5$ denote five granularities from coarse to fine. We first examine single granular models' performance, and then increased to two granularities and three granularities. We add the intermediate granularity $g3$ on the basis of the coarsest granularity $g1$ and finest granularity $g5$, and the model effect improves. We find that the addition of the confidence estimation module further



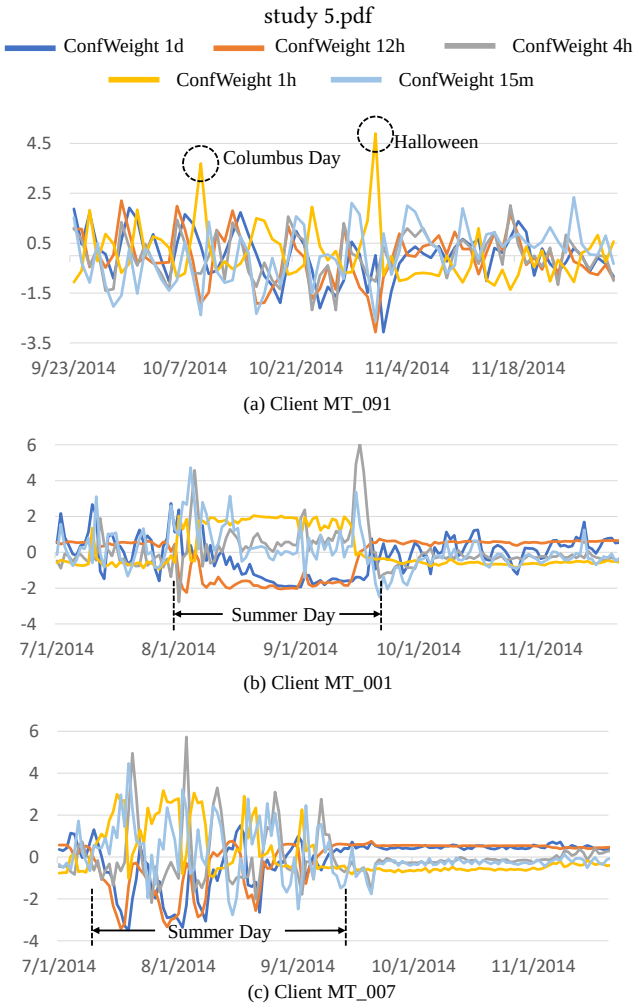
(a) Electricity Data. (b) Stock Data.

Figure 4: Effectiveness of Cross-Granularity Residual Learning Process on Electricity Data and Stock Data.

improved the effectiveness of the model, suggesting that the module captures the dynamic validity at different granularities. Finally, we test the 5 granularity model. Overall, it is clear that increasing the amount of granularity significantly improves the results, which demonstrates that the introduction of multi-granularity information effectively improves the performance of the time series prediction model. However, the marginal benefit is diminishing with the increase of granularity amounts.

4.2.4 Case Study. To intuitively explore the effectiveness of the confidence estimation mechanism, we illustrate the changes in confidence weights of some clients. For each client, we calculate the Z-Score normalized confidence weight of the 5 granularities during the test period. The results are presented in Figure 5, in which the horizontal axis represents the date, and each line reflects the change of different granularity's confidence weight.

We expect to observe the change patterns of confidence weights on some special dates when user electricity consumption habits may change. From Figure 5(a), the confidence weight of the user's fine-grained data (1h) has risen sharply in two US statutory holidays: Columbus Day and Halloween. To celebrity the festivals, users' power consumption habits have changed. The model captures the original daily frequency information is no longer applicable. More fine-grained patterns should be captured. From Figure 5(b), we find that in summer day (late-August to mid-September), the confidence weight of finer-grained data (15min, 1h, 4h) increases, while the weight of coarser-grained data (12h, 1day) decreases. This phenomenon indicates that for the user MT_001, during the peak of summer electricity consumption, finer-grained electricity usage rules (for instance whether to use high-power appliances such as air conditioners at a fine-grained time) are more crucial for electricity usage prediction. In Figure 5(c), user MT_007 also has this phenomenon, but it is from mid-July to early-September. This further proves that our model has discovered the user's personalized summer electricity usage date pattern, which is more flexible and effective than the rule-based model.



(a) Client MT_091

(b) Client MT_001

(c) Client MT_007

Figure 5: The Changes in Confidence Weights of Some Clients on Electricity Data. The horizontal axis represents the data, and each line reflects the change of different granularity's confidence weight.

5 CONCLUSION AND FUTURE WORK

In this paper, we uncover the importance of exploring multi-granularity patterns for time series prediction and propose a multi-granularity learning framework. Specifically, to overcome the semantic overlap between multi-granularity data, we design a novel cross-granularity residual learning method to avoid the model being dominated by the redundant coarse-grained trend information. A confidence estimation module is designed to strengthen the effectiveness of MRLF further. The experiments on real-world data demonstrated the effectiveness of MRLF for enhancing the time series prediction capacity.

In the future, we will study how to make the generation process of multi-granularity data learnable rather than manually specified directly. Reinforcement learning, meta learning, information theory and other ideas may be integrated into the modeling process, so that multi-granularity input data can provide more information to improve the final prediction effect.

REFERENCES

[1] 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations* (2015).

[3] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *International Conference on Machine Learning*. 531–540.

[4] Mikolaj Binkowski, Gautier Marti, and Philippe Donnat. 2018. Autoregressive convolutional neural networks for asynchronous time series. In *International Conference on Machine Learning*. PMLR, 580–589.

[5] George Box. 2013. Box and Jenkins: time series analysis, forecasting and control. In *A Very British Affair*. Springer, 161–215.

[6] Atul J Butte and Isaac S Kohane. 1999. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In *Biocomputing 2000*. World Scientific, 418–429.

[7] Chi Chen, Li Zhao, Jiang Bian, Chunxiao Xing, and Tie-Yan Liu. 2019. Investment behaviors can tell what inside: Exploring stock intrinsic properties for stock trend prediction. In *Proceedings of the 25th ACM SIGKDD*. 2376–2384.

[8] Mingyue Cheng, Fajie Yuan, Qi Liu, Xin Xin, and Enhong Chen. 2021. Learning Transferable User Representations with Sequential Behaviors via Contrastive Pre-training. In *2021 IEEE International Conference on Data Mining (ICDM)*. 51–60.

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[10] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[11] Zhicheng Cui, Wenlin Chen, and Yixin Chen. 2016. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995* (2016).

[12] Shumin Deng, Ningyu Zhang, Wen Zhang, Jiaoyan Chen, Jeff Z Pan, and Huajun Chen. 2019. Knowledge-driven stock trend prediction and explanation via temporal convolutional network. In *Companion Proceedings of The 2019 World Wide Web Conference*. 678–685.

[13] Yi Ding, Weiqing Liu, Jiang Bian, Daoqiang Zhang, and Tie-Yan Liu. 2018. Investor-imitator: A framework for trading knowledge extraction. In *Proceedings of the 24th ACM SIGKDD*. 1310–1319.

[14] Fuli Feng, Huimin Chen, Xiangnan He, Ji Ding, Maosong Sun, and Tat-Seng Chua. 2019. Enhancing Stock Movement Prediction with Adversarial Training. *IJCAI* (2019).

[15] Pierre Geurts et al. 2018. dynGENIE3: dynamical GENIE3 for the inference of gene networks from time series expression data. *Scientific reports* 8, 1 (2018), 1–12.

[16] Andrew C Harvey. 1990. Forecasting, structural time series models and the Kalman filter. (1990).

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European conference on computer vision*. Springer, 630–645.

[19] Steven Craig Hillmer and George C Tiao. 1982. An ARIMA-model-based approach to seasonal adjustment. *J. Amer. Statist. Assoc.* 77, 377 (1982), 63–70.

[20] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[21] Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 1 (1970), 55–67.

[22] Min Hou, Chang Xu, Yang Liu, Weiqing Liu, Jiang Bian, Le Wu, Zhi Li, Enhong Chen, and Tie-Yan Liu. 2021. *Stock Trend Prediction with Multi-Granularity Data: A Contrastive Learning Approach with Adaptive Fusion.* Association for Computing Machinery, New York, NY, USA, 700–709.

[23] Siteng Huang, Donglin Wang, Xuehan Wu, and Ao Tang. 2019. Dsanet: Dual self-attention network for multivariate time series forecasting. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2129–2132.

[24] J Stuart Hunter. 1986. The exponentially weighted moving average. *Journal of quality technology* 18, 4 (1986), 203–210.

[25] Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

[26] Justin B Kinney and Gurinder S Atwal. 2014. Equitability, mutual information, and the maximal information coefficient. *Proceedings of the National Academy of Sciences* 111, 9 (2014), 3354–3359.

[27] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. In *International Conference on Learning Representations*.

[28] Alireza Koochali, Peter Schichtel, Andreas Dengel, and Sheraz Ahmed. 2019. Probabilistic forecasting of sensory data with generative adversarial networks–forgan. *IEEE Access* 7 (2019), 63868–63880.

[29] Nojun Kwak and Chong-Ho Choi. 2002. Input feature selection by mutual information based on Parzen window. *IEEE transactions on pattern analysis and machine intelligence* 24, 12 (2002), 1667–1671.

[30] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 95–104.

[31] Zhi Li, Hongke Zhao, Qi Liu, Zhenya Huang, Tao Mei, and Enhong Chen. 2018. Learning from History and Present: Next-Item Recommendation via Discriminatively Exploiting User Behaviors. In *Proceedings of the 24th ACM SIGKDD (KDD '18)*. New York, NY, USA, 1734–1743.

[32] Guang Liu, Yuzhao Mao, Qi Sun, Hailong Huang, Weiguo Gao, Xuan Li, Jianping Shen, Ruifan Li, and Xiaojie Wang. 2020. Multi-scale Two-way Deep Neural Network for Stock Trend Prediction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence,IJCAI-20*. 4555–4561.

[33] Frederik Maes, Andre Collignon, Dirk Vandermeulen, Guy Marchal, and Paul Suetens. 1997. Multimodality image registration by maximization of mutual information. *IEEE transactions on Medical Imaging* 16, 2 (1997), 187–198.

[34] Manfred Mudelsee. 2013. *Climate time series analysis*. Vol. 30. Springer.

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[36] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437* (2019).

[37] Clare Ostle, Richard C Thompson, Derek Broughton, Lance Gregory, Marianne Wootton, and David G Johns. 2019. The rise in ocean plastics evidenced from a 60-year time series. *Nature communications* 10, 1 (2019), 1–6.

[38] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* 27, 8 (2005), 1226–1238.

[39] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. 2017. A dual-stage attention-based recurrent neural network for time series prediction. *IJCAI* (2017), 2627–2633.

[40] Marco S. Reis. 2019. Multiscale and Multi-Granularity Process Analytics: A Review. *Processes* 7, 2 (2019). https://doi.org/10.3390/pr7020061

[41] Stephen Roberts, Michael Osborne, Mark Ebden, Steven Reece, Neale Gibson, and Suzanne Aigrain. 2013. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371, 1984 (2013), 20110550.

[42] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. 2019. Temporal pattern attention for multivariate time series forecasting. *Machine Learning* 108, 8 (2019), 1421–1441.

[43] Arunesh Kumar Singh, S Khatoon Ibraheem, Md Muazzam, and DK Chaturvedi. 2013. An overview of electricity demand forecasting techniques. *Network and Complex Systems* 3, 3 (2013), 38–48.

[44] Huan Song, Deepta Rajan, Jayaraman J Thiagarajan, and Andreas Spanias. 2018. Attend and diagnose: Clinical time series analysis using attention models. In *Thirty-second AAAI conference on artificial intelligence*.

[45] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. 2020. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 5956–5963.

[46] Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057* (2000).

[47] Vladimir Vapnik, Steven E Golowich, Alex Smola, et al. 1997. Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems* (1997), 281–287.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[49] Likang Wu, Zhi Li, Hongke Zhao, Qi Liu, and Enhong Chen. 2022. Estimating Fund-Raising Performance for Start-up Projects from a Market Graph Perspective. *Pattern Recogn.* 121, C (jan 2022), 13 pages.

[50] Likang Wu, Zhi Li, Hongke Zhao, Zhen Pan, Qi Liu, and Enhong Chen. 2020. Estimating early fundraising performance of innovations via graph-based market environment model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6396–6403.

[51] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD*. 753–763.

[52] Wentao Xu, Weiqing Liu, Chang Xu, Jiang Bian, Jian Yin, and Tie-Yan Liu. 2021. REST: Relational Event-driven Stock Trend Forecasting. In *WWW '21: The Web Conference 2021*. ACM / IW3C2, 1–10. https://doi.org/10.1145/3442381.3450032

[53] Liheng Zhang, Charu Aggarwal, and Guo-Jun Qi. 2017. Stock price prediction via discovering multi-frequency trading patterns. In *Proceedings of the 23rd ACM SIGKDD*. 2141–2149.

[54] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*.