# Towards Dynamic Trend Filtering through Trend Point Detection with Reinforcement Learning

**Jihyeon Seong** [1] , **Sekwang Oh** [1] and **Jaesik Choi** [1,2]
[1]Korea Advanced Institute of Science and Technology (KAIST), South Korea
[2]INEEJI, South Korea
{jihyeon.seong, oskoskosk, jaesik.choi}@kaist.ac.kr,

## Abstract

Trend filtering simplifies complex time series data by applying smoothness to filter out noise while emphasizing proximity to the original data. However, existing trend filtering methods fail to reflect abrupt changes in the trend due to 'approximateness,' resulting in constant smoothness. This approximateness uniformly filters out the tail distribution of time series data, characterized by extreme values, including both abrupt changes and noise. In this paper, we propose Trend Point Detection formulated as a Markov Decision Process (MDP), a novel approach to identifying essential points that should be reflected in the trend, departing from approximations. We term these essential points as Dynamic Trend Points (DTPs) and extract trends by interpolating them. To identify DTPs, we utilize Reinforcement Learning (RL) within a discrete action space and a forecasting sum-of-squares loss function as a reward, referred to as the Dynamic Trend Filtering network (DTF-net). DTF-net integrates flexible noise filtering, preserving critical original subsequences while removing noise as required for other subsequences. We demonstrate that DTF-net excels at capturing abrupt changes compared to other trend filtering algorithms and enhances forecasting performance, as abrupt changes are predicted rather than smoothed out.

## 1 Introduction

Trend filtering emphasizes proximity to the original time series data while filtering out noise through smoothness [Leser, 1961]. Smoothness in trend filtering simplifies complex patterns within noisy and non-stationary time series data, making it effective for forecasting and anomaly detection [Park *et al.*, 2020]. While smoothness achieves the property of noise filtering, an 'abrupt change' denotes a point in a time series where the trend experiences a sharp transition, signaling a change in slope. Given that abrupt changes determine the direction and persistence of the slope, it is crucial to incorporate them into the trend. Traditional trend filtering employs a sum-of-squares function to reflect abrupt changes while utilizing second-order differences as a regularization term to attain smoothness [Hodrick and Prescott, 1997; Kim *et al.*, 2009]. However, we found that the constant nature of smoothness filters out abrupt changes, making it challenging to distinguish them from noise.

The issue of constant smoothness arises from the reliance on the property of 'approximateness.' Evidence presented by [Ding *et al.*, 2019] suggests that the sum-of-squares function eliminates tail distribution as outliers since it approximates a Gaussian distribution with a light-tail shape. As both abrupt changes and noise reside within the tail distribution, filtering out only noise becomes challenging. This uniform filtering results in the loss of valuable abrupt changes that should be reflected in the trend [Wen *et al.*, 2019].

In this paper, we propose Trend Point Detection formulated as a Markov Decision Process (MDP), aiming to identify essential points that should be reflected in the trend, departing from approximateness [Sutton and Barto, 2018]. These essential points are termed Dynamic Trend Points (DTPs), and trends are extracted by interpolating them. We utilize the Reinforcement Learning (RL) algorithm within a discrete action space to solve the MDP problem, referred to as a Dynamic Trend Filtering network (DTF-net) [Schulman *et al.*, 2017]. RL can directly detect essential points through an agent without being constrained by fixed window sizes or frequencies within the time series data domain. This dynamic approach enables the adjustment of noise filtering levels for each subsequence within the time series.

Building on prior research regarding reward function learning based on Gaussian Process (GP) [Biyik *et al.*, 2020], we define the reward function as the sum-of-squares loss function from Time Series Forecasting (TSF). This choice is supported by [Ding *et al.*, 2019], which suggests that the sum-of-squares function approximates a Gaussian distribution and functions similarly to a Gaussian kernel. Note that using a Gaussian kernel function as a reward leverages RL to effectively optimize the agent while learning the full distribution of time series data. Through the TSF reward, temporal dependencies around DTPs can be captured, and the level of smoothness is controlled by adjusting the forecasting window size. Additionally, to address the overfitting issue, we apply a random sampling method to both the state and the reward.

We compare DTF-net with four categorized baselines: trend filtering (TF), change point detection (CPD), anomaly detection (AD), and time series forecasting (TSF) algorithms.

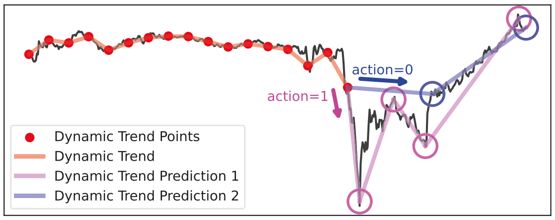



Figure 1: **Dynamic Trend Filtering.** DTF-net extracts dynamic trends from time series data. Dynamic Trend Points (DTPs) are determined based on action predictions, and the dynamic trend is extracted through interpolation. The agent's action prediction directly influences the variation in trend extraction.

First, traditional TF approaches commonly rely on approximations achieved through optimizing sum-of-squares functions or employing decomposition methods, which often neglect abrupt changes as noise. Second, CPD methods are rooted in probabilistic frameworks, prioritizing the detection of changes in distribution while often disregarding extreme values as outliers. Third, AD methods concentrate heavily on identifying abnormal points, sometimes overlooking the significance of distribution shifts in the data. Lastly, TSF models are categorized into decomposition-based and patching-based models, which also neglect abrupt changes. Contrary to all the aforementioned baselines, DTF-net focuses on point detection to reflect abrupt changes in the trend, enhancing the performance of trend filtering and forecasting. To the best of our knowledge, this is the first approach that employs MDP and RL for trend filtering, aiming to reflect both abrupt changes and smoothness simultaneously.

Therefore, our contributions are as follows:

- We identified the issue of 'approximateness,' which leads to constant smoothness in traditional trend filtering, filtering out both abrupt changes and noise.
- We introduce Trend Point Detection formulated as an MDP, aiming to identify essential trend points that should be reflected in the trend, including abrupt changes. Additionally, we propose DTF-net, an RL algorithm that predicts DTPs through agents.
- We employ the forecasting sum-of-squares cost function, inspired by reward function learning based on GP, which allows for the consideration of temporal dependencies when capturing DTPs. A sampling method is applied to prevent the overfitting issue.
- We demonstrate that DTF-net excels at capturing abrupt changes compared to other trend filtering methods and enhances performance in forecasting tasks.

## 2 Related Work

### 2.1 Trend Filtering

Traditional trend-filtering algorithms have employed various methods to capture abrupt changes. H-P [Hodrick and Prescott, 1997] and $\ell_1$ [Kim *et al.*, 2009] optimize the sum-of-squares function, a widely used cost function for trend filtering. However, they often face challenges in the delayed detection of abrupt changes due to the use of second-order difference operators for smoothness. To address this issue, the TV-denoising algorithm [Chan *et al.*, 2001] was introduced, relying on first-order differences. Nevertheless, this strategy introduces delays in detecting slow-varying trends while overly focusing on abrupt changes. These methods encounter difficulties in handling heavy-tailed distributions due to the use of the sum-of-squares function [Wen *et al.*, 2019].

Contrary to sum-of-squares function methods, alternative approaches to trend filtering exist. For example, frequency-based methods like Wavelet [Craigmile and Percival, 2002] are designed for non-stationary signals but are susceptible to overfitting. The Empirical Mode Decomposition (EMD) algorithm [Wu *et al.*, 2007] decomposes a time series into a finite set of oscillatory modes, but it generates overly smooth trends. Lastly, the Median filter [Siegel, 1982] is a non-linear filter that selects the middle value from the sorted central neighbors; therefore, outlier values that deviate significantly from the center of the data are excluded.

### 2.2 Extreme Value Theorem

Abrupt changes in a time series reside in the tail of the data distribution, making them rare events. However, their impact is significant, as they can alter the slope of the time series and affect the consistency of trends. Once an abrupt change occurs, its effects are often permanent until the next one occurs. Therefore, detecting abrupt changes is crucial to minimize false negative rates and capture important information.

Real-world time series data commonly exhibit a long-heavy tail distribution. Formally, the tail distribution is defined as follows:

$$\lim_{T\to\infty} P\{max(y_1, ..., y_T) \leq y\} = \lim_{T\to\infty} F^T(y) = 0, \quad (1)$$

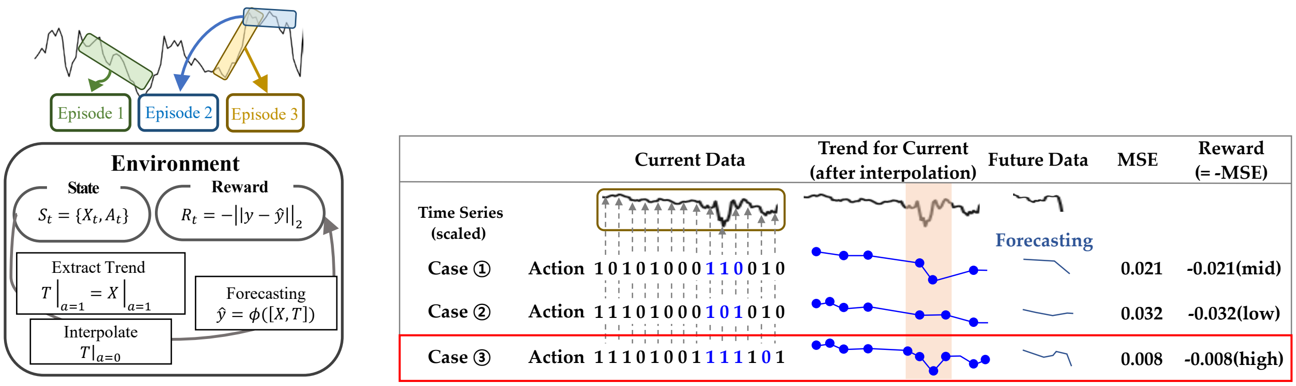
where $T$ random variables $\{y_1, ..., y_T\}$ are i.i.d. sampled from distribution $F_Y$ [von Bortkiewicz, 1921; Ding *et al.*, 2019]. Furthermore, extreme values within the tail distribution can be modeled using Extreme Value Theory.

**Theorem 1** (Extreme Value Theory [Fisher and Tippett, 1928; Ding *et al.*, 2019])**.** *If the distribution in Equation (1) is not degenerate to 0 under a linear transformation of y, the distribution of the class with the non-degenerate distribution $G(y)$ should be as follows:*

$$G(y) = \begin{cases} exp(-(1-\frac{1}{\gamma}y)^\gamma), & \gamma \neq 0, 1-\frac{1}{\gamma}y \geq 0, \\ exp(-e^{-y}), & \gamma = 0. \end{cases} \quad (2)$$

Extreme Value Theory (EVT) has demonstrated that extreme values exhibit a limited degree of freedom [Lorenz, 1963]. This implies that the occurrence patterns of extreme values are recursive and can be memorized by a model [Altmann and Kantz, 2005]. Essentially, a model with substantial capacity and temporal invariance can effectively learn abrupt changes, which are categorized as extreme values.

However, extreme values are typically either unlabeled or imbalanced, making them challenging to predict. In classification tasks, previous research [Raj *et al.*, 2016] has highlighted the susceptibility of deep networks to the data im-




(a) DTF-net Environment (b) DTF-net Reward

Figure 2: **DTF-net Architecture.** DTF-net has three processes to detect DTPs: 1) The agent predicts actions within a discrete space; 2) With the predicted actions, trends are extracted by interpolating them; 3) The agent is updated through the forecasting sum-of-squares function as a reward; with time series data $X$ and trend $\mathcal{T}$ as inputs. For the reward calculation, as demonstrated in (b)-Case 3, when DTF-net successfully identifies abrupt changes, the prediction outcomes significantly improve, resulting in the highest reward.

balance issue. In forecasting tasks, [Ding *et al.*, 2019] provided evidence that minimizing the sum-of-squares loss presupposes a Gaussian distribution, which differs significantly from long-heavy-tail distributions. Motivated by these issues, we define Trend Point Detection as a problem formulation aimed at detecting essential points that should be reflected in the trend rather than smoothed out. This formulation identifies DTPs, which encompass abrupt changes, midpoints of distribution shifts, and other critical points influencing changes in the trend slope, occurring in both short and long intervals. As illustrated in Figure 1, Trend Point Detection is formulated as an MDP and utilizes RL to detect abrupt changes directly through the agent's action prediction. Note that we train the DNNs as a policy network of an RL agent to learn the pattern of extreme value occurrence, distinct from approximating abrupt changes as output.

### 2.3 Markov Decision Process and Reinforcement Learning

MDP is a mathematical model for decision-making when an agent interacts with an environment. It relies on the first-order Markov property, indicating that the future state depends solely on the current state. MDP comprises components denoted as $\langle S, A, P, \mathcal{R}, \gamma \rangle$. Here, $S$ denotes the set of environment states, while $A$ represents the set of actions undertaken by the agent at state $S$. The transition probability, $P = \Pr(S'|S, A)$, signifies the probability of transitioning from the current state $S$ to the next state $S'$. The reward, $\mathcal{R} = \mathbb{E}[\mathcal{R}(S, A, S')|S, A]$, where $\mathcal{R}(S, A, S')$ represents the immediate reward obtained when transitioning from state $S$ to $S'$ by taking action $A$. The discount factor $\gamma \in (0, 1]$ governs the trade-off between current and future rewards [Sutton and Barto, 2018]. We can formulate any time series data with an MDP for Trend Point Detection, as detecting points always adheres to the first-order Markov property [Wu and Ortiz, 2021]. These points are determined solely by the current time step and remain unaffected by past observations, sharing properties similar to those of predicting stock trading points.

In RL, actions are predicted through a policy network denoted as $\pi(A|S) = \Pr(A|S)$ for each state, representing the probability of action $A$ at state $S$. The state-value function $v_\pi(S) = \mathbb{E}_\pi[G|S]$ estimates the expected reward value for a state $S$ under policy $\pi$, where $G = \sum_{k=0}^{\infty} \gamma^k \mathcal{R}'_k$ denotes the expected sum of future rewards starting from the next reward $R'$. In RL of discrete action spaces, methods like Advantage Actor-Critic (A2C) [Mnih *et al.*, 2016] and Proximal Policy Optimization (PPO) [Schulman *et al.*, 2017] directly train the policy $\pi$ using the estimated state-value function $v$. In contrast, Deep Q-Network (DQN) [Mnih *et al.*, 2015] finds the optimal action-value function, denoted as $q_\pi(S, A) = \mathbb{E}_\pi[G|S, A]$. This function represents the expected cumulative reward for taking action $A$ in state $S$ under policy $\pi$ and is determined through the Bellman equation (Appendix B). DTF-net utilizes RL to extract flexible trends through dynamic action prediction from a deep policy network $\pi$, learning within the time series data environment formulated as an MDP of the Trend Point Detection problem.

## 3 Dynamic Trend Filtering Network

### 3.1 Trend Point Detection

**Environment Definition**

Time series data is defined as $\mathbf{T} = \{(\mathbf{X}_1, y_1), (\mathbf{X}_2, y_2), \ldots, (\mathbf{X}_N, y_N)\}$, where $\mathbf{X} \in \mathbb{R}^D$ represents the input, $y \in \mathbb{R}^d$ represents the output, and the dataset comprises a total of $N \in \mathbb{Z}^+$ samples. Here, $D$ and $d$ denote the input and output dimensions, respectively, both of which are positive integers.

The Trend Point Detection problem formulation takes input $\mathbf{X}$ representing the environment and outputs DTPs, encompassing abrupt changes, midpoints of distribution shifts, and other critical points influencing trend slope changes occurring at both short and long intervals. The output consists of specific univariate time series $y^{(i)}$ labeled with binary values, where $i \in d$ of target.

- **State** $S = [\mathbf{X}_t, A_t]$: the positional encoded vector set of time series data $\mathbf{X}$ and action $A$ with horizon $t$.

- **Action** $A$: a discrete set with $(a = 1)$ for detecting DTP and $(a = 0)$ for smoothing.
- **Reward** $\mathcal{R}(S, A, S')$: the change in forecasting sum-of-squares function value when action $A$ is taken at state $S$ and results in the transition to the next state $S'$.
- **Policy** $\pi(A|S)$: the probability distribution of $A$ at $S$.

The RL algorithm, named DTF-net, employs a policy network $\pi$ within the defined MDP. It receives the state $S$ as input and outputs the binary labeled target $y^{(i)}$, also denoted as $A$, learned through the maximization of cumulative rewards $\mathcal{R}$. DTF-net is designed to extract dynamic trends by interpolating detected essential trend points, referred to as DTPs and represented by the set $\{y^{(i)} = 1\}$ or $\{A|_{a=1}\}$.

### Episode and State for DTF-net

Previous studies in RL for time series [Liu *et al.*, 2022a] have generally adopted a sequential approach. In contrast, DTF-net introduces dynamic segmentation with variable lengths comprising one episode through random sampling. The discrete uniform distribution is specifically chosen to ensure that all sub-sequences are considered equally:

$$\begin{aligned} s &\sim \text{unif}\{0, N\}, \\ l &\sim \text{unif}\{h + p, H\}, \end{aligned} \tag{3}$$

where $s$ represents the starting points of the sub-sequence, $l$ denotes the sub-sequence length, $h$ denotes the forecasting look-back horizon, $p$ denotes the forecasting prediction horizon, and $H$ represents the maximum length comprising one episode. With sampling, the length and starting point of the sub-sequence are defined, resulting in a non-sequential and random progression of the episode. This sampling approach mitigates the overfitting issue by allowing the model to use only a portion of the sequence.

Within a single episode, DTF-net cumulatively constructs the state $S$. To maintain a constant state length within an episode, we employ positional encoding as follows:

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}}),$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{model}}).$$

The cumulative state progression is achieved by gradually expanding the state representation $S_t$ as the step unfolds as follows,

$$S_t = PE(\{\mathbf{X}_{s:s+t}, A_{0:t}\}), \text{ where } t < l. \tag{4}$$

Through cumulative state construction, the agent can learn sequential information in the time series (Appendix D.1).

## 3.2 Reward Function of DTF-net

### GP and Reward Function Learning

Traditional trend filtering methods utilize the sum-of-squares function, also known as the Mean Squared Error (MSE), to approximate abrupt changes when extracting trends. However, [Ding *et al.*, 2019] provided evidence that minimizing the sum-of-squares function assumes that the model output distribution determined using the MSE cost function denoted as $\hat{P}(Y)$, follows a Gaussian distribution with variance $\tau$, grounded in Bregman's theory [Banerjee *et al.*, 2005].

**Algorithm 1** Reward Procedure of DTF-net

```
procedure REWARD(S_t)
    X' = X_{t-(h+p):t}, A' = A_{t-(h+p):t}
    T ← 0
    T|_{a=1} ← X'_{0:h}|_{a=1}
    while n ≤ h do
        // n for time-axis and x ∈ X'
        T_n ← x_n = x_{n-1} + (x_{n+1} − x_{n-1})/2
        n ← n + 1
    end while
    ŷ ← φ([X'_{0:h}, T])
    r ← (1/p) Σ_{i=1}^{p} (y_i − ŷ_i)^2
    return −r
end procedure
```

$$\begin{aligned} \hat{P}(Y) &= \min \sum_{t=1}^{T} ||y_t - o_t||^2, \\ &= \max_\theta \Pi_{t=1}^{T} P(y_t|x_t, \theta), \\ &= \frac{1}{N} \sum_{t=1}^{T} \mathcal{N}(y_t, \hat{\tau}^2). \end{aligned} \tag{5}$$

where $o \in Y$ represents the output from a model parameterized by $\theta$. This also suggests that model $\theta$ operates in a manner similar to a Kernel Density Estimator (KDE) employing a Gaussian kernel [Rosenblatt, 1956].

Contrary to approximations, DTF-net utilizes a policy network $\pi$ to predict DTPs, including abrupt changes. However, defining a reward function in general time series data is challenging but is the most crucial task in RL training for optimizing the policy network. To tackle this challenge, DTF-net draws inspiration from previous works, which employ the Gaussian Process (GP) for reward function learning [Kuss and Rasmussen, 2003; Biyik *et al.*, 2020].

Formally, GP [Williams and Rasmussen, 1995] assumes noisy targets $y_i = f(x_i) + \epsilon_i$ that are jointly Gaussian with a covariance function $k$:

$$P(y|x) \sim \mathcal{N}(0, \mathbf{K}), \text{ where } \mathbf{K}_{pq} = k(x_p, x_q). \tag{6}$$

With a Gaussian covariance function,

$$k(x_p, x_q|\theta) = v^2 exp(-(x_p - x_q)^\top \Lambda^{-1}(x_p - x_q)/2) + \delta_{pq}\sigma_n^2,$$

where diagonal matrix $\Lambda$, $v$, and $\sigma$ are hyperparameters in $\theta$, the predictive distribution for input $x^*$ follows Gaussian:

$$\begin{aligned} P(o_t^*|x^*, x, y, \theta) \sim & \mathcal{N}(k(x^*, x)\mathbf{K}^{-1}y, \\ & k(x^*, x^*) - k(x^*, x)\mathbf{K}^{-1}k(x, x^*)). \end{aligned} \tag{7}$$

The GP model inherently learns a full distribution of time series data, enabling RL to effectively optimize the policy network (Appendix B). Leveraging these insights, DTF-net's reward function is defined as the sum-of-squares function from

| Trend Filtering | | Linear Signal+Noise (0.2) | | | |
|---|---|---|---|---|---|
| | | 1) full-sequence | | 2) abrupt-sequence | |
| | | MSE | MAE | MSE | MAE |
| CPD | ADAGA [Caldarelli *et al.*, 2022] | 4.3434 | 1.4428 | 7.0120 | 1.8668 |
| | RED-SDS [Ansari *et al.*, 2021] | 1.0036 | 0.6782 | 1.6660 | 0.9365 |
| AD | TimesNet [Wu *et al.*, 2023] | 3.0841 | 1.4204 | 3.3304 | 1.4364 |
| | AnomalyTransformer [Xu *et al.*, 2022] | 7.7506 | 2.1817 | 10.1336 | 2.7242 |
| | DCdetector [Yang *et al.*, 2023] | *0.0094* | *0.0255* | 3.6300 | 1.3721 |
| TF | EMD [Wu *et al.*, 2007] | 5.3096 | 1.7401 | 6.4410 | 1.8431 |
| | Median [Siegel, 1982] | 4.4766 | 1.5525 | 5.6859 | 1.8204 |
| | H-P [Hodrick and Prescott, 1997] | 0.2253 | 0.3311 | 0.3238 | 0.3934 |
| | Wavelet [Craigmile and Percival, 2002] | $1e-30$ | $6e-16$ | $2e-30$ | $8e-16$ |
| | $\ell_1$ (λ=0.1) [Kim *et al.*, 2009] | 0.0461 | 0.1703 | 0.0500 | 0.1807 |
| | $\ell_1$ ($\lambda = 5e-4$) | 0.0004 | 0.0175 | 0.0004 | 0.0174 |
| | DTF-net (ours) | **0.0289** | **0.0826** | **0.0286** | **0.0855** |

Table 1: **Comparison with advanced CPD, AD, and TF methods in synthetic data.** We conduct trend filtering analysis on synthetic data, evaluating it against the ground truth of a linear signal with added noise. We consider two cases: one with the full sequence and the other with a 30-window interval sub-sequence containing abrupt changes. The evaluation metrics are Mean Squared Error (MSE) and Mean Absolute Error (MAE), where lower values indicate better performance. The best performance is **bolded**, and the second-best performance is underlined. For the special case of DCdetector, the performance is denoted in *italic*.

Time Series Forecasting (TSF). This choice leads to more efficiency in calculating rewards compared to GP while achieving reward function learning within the Gaussian distribution. To incorporate captured abrupt changes into the forecasting model, DTPs are included as an additional input. As shown in Figure 2, when the forecasting model predicts upward or downward trends instead of smoothing them out, the agent receives a higher reward. Thus, DTF-net learns temporal dependencies when capturing DTPs.

### Forecasting Reward Function of DTF-net

As shown in Algorithm 1, the reward process involves time series data $\mathbf{X}_{t-(h+p):t}$ and action $A_{t-(h+p):t}$ in state $S_t$ at time step $t$, both having a sequence length denoted as $(h+p)$, where $h$ denotes the past horizon and $p$ denotes the forecasting horizon (under the condition $t-(h+p) > 0$). The trend $\mathcal{T}$ initiates with $\mathbf{X}$ values assigned only under the condition of action $A|_{a=1}$, and linear interpolation is applied for the remaining values. Subsequently, forecasting is conducted with a prediction length $p$ defined by a hyperparameter. The reward is computed as the negative sum-of-squares loss between the predicted $\hat{y}$ and the ground truth $y$.

DTF-net uses a penalty reward as a negative value from the Mean Squared Error (MSE) function, and there is a possibility of an overfitting issue. Therefore, DTF-net utilizes random sampling from a discrete uniform distribution, providing better control over model updates.

$$k \sim \text{unif}\{s, s+l\},$$
$$R = \begin{cases} \text{REWARD}(E_t) & \text{if } t = k, \\ 0 & \text{if } t \neq k. \end{cases} \tag{8}$$

We empirically demonstrate that irregularly applying penalties through sampling can prevent overfitting rather than penalizing at every step in Section 4.2 (Appendix D.2).

In summary, DTF-net is designed to extract dynamic trends $\mathcal{T}$ by interpolating detected DTPs. A simple ML time series predictor $\phi$ is integrated into DTF-net to calculate the reward, with the input of $[\mathbf{X}, \mathcal{T}]$. As shown in Figure 2, including the detected abrupt changes as an additional input to the forecasting model ensures that the forecasting output reflects both upward and downward trends, maximizing the reward.

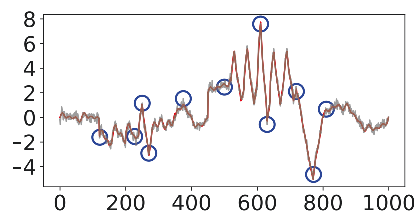


Figure 3: **Synthetic Data.**

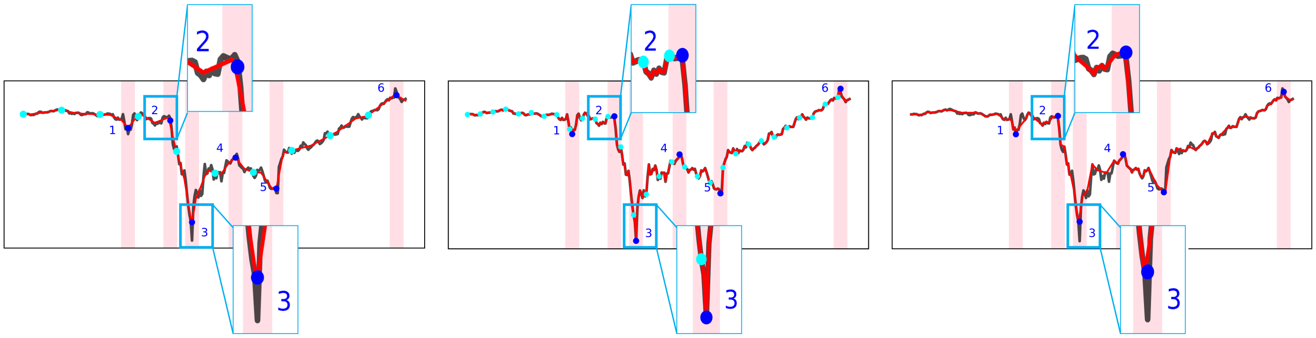
## 4 Experiment

### 4.1 Trend Filtering Analysis

#### Experimental Settings

Analyzing trend filtering methods quantitatively poses two challenges: 1) defining a ground truth for the trend is challenging, and 2) labeling abrupt changes is challenging. To address these, as shown in Figure 3, we generate a synthetic trend signal with $1,000$ time points. This synthetic dataset contains 11 abrupt changes, including 1) a sudden drop to negative values around -2 and -5 at time points 100 and 800, respectively; 2) a mean shift from 0 to 4 with high variance occurring between time points 500 and 700; and 3) a sine wave starting from time point 800 to 1000, completing one cycle. We add Gaussian noise with a standard deviation of 0.2 to simulate real-world conditions.

To evaluate the trend filtering results, we set up the experiment as follows. We employ Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics to measure the proximity to the original data, which is a key aspect of trend filtering. For the ground truth, we use a linear signal with added noise to assess DTF-net's robustness to noisy data. In cases with added noise, we assume that filtering out at least 10%



(a) $\ell_1(\lambda = 0.1)$ (b) $\ell_1(\lambda = 5e-4)$ (c) DTF-net

Figure 4: **Qualitative comparison with $\ell_1$ and DTF-net.** The figure illustrates the trends obtained from the $\ell_1$ and DTF-net using the Nasdaq intraday dataset. The red line denotes the output of each trend filtering method, with red vertical boxes indicating arbitrarily set abrupt changes. The blue dots denote the captured abrupt changes, while the sky-blue dots highlight the constant smoothness from $\ell_1$. Notably, DTF-net has the capability to apply varying levels of smoothness to individual sub-sequences.

| Methods | | DTF-Linear (ours) | | $\ell_1(\lambda = 0.1)$-Linear | | PatchTST/42 | | NLinear | | DLinear | | FEDformer-f | | FEDformer-w | | Autoformer | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| Exchange | 24 | **0.0250** | **0.1198** | 0.0266 | 0.1248 | 0.0387 | 0.1513 | 0.0275 | 0.1264 | 0.0290 | 0.1284 | 0.0381 | 0.1545 | 0.0387 | 0.1564 | 0.0687 | 0.2041 |
| | 48 | **0.0487** | **0.1658** | 0.0505 | 0.1708 | 0.0624 | 0.1873 | 0.0505 | 0.1705 | 0.0585 | 0.1907 | 0.0548 | 0.1818 | 0.1068 | 0.2528 | 0.1095 | 0.2485 |
| | 96 | **0.0980** | **0.2349** | 0.1007 | 0.2440 | 0.1833 | 0.3436 | 0.0990 | 0.2361 | 0.1063 | 0.2530 | 0.1440 | 0.2980 | 0.1386 | 0.2894 | 0.1834 | 0.3306 |
| | 192 | 0.1983 | 0.3583 | 0.2045 | 0.3518 | 0.2550 | 0.3987 | 0.2030 | 0.3400 | 0.1959 | 0.3554 | 0.2790 | 0.4163 | 0.2841 | 0.4217 | 0.3465 | 0.4510 |
| | 336 | **0.3160** | **0.4561** | 0.3337 | 0.4666 | 0.5161 | 0.5442 | 0.4174 | 0.4857 | 0.3276 | 0.4627 | 0.4466 | 0.5130 | 0.5685 | 0.5890 | 0.4488 | 0.5291 |
| | 720 | **0.7933** | **0.6874** | 0.9515 | 0.7636 | 1.1143 | 0.8063 | 1.0420 | 0.7807 | 0.9071 | 0.7415 | 1.2122 | 0.8492 | 1.2912 | 0.8876 | 1.2463 | 0.8694 |
| ETTh1 | 24 | 0.0253 | 0.1205 | *0.0234* | *0.1140* | 0.0266 | 0.1238 | 0.0266 | 0.1240 | 0.0273 | 0.1262 | 0.0358 | 0.1450 | 0.0381 | 0.1524 | 0.0694 | 0.2042 |
| | 48 | 0.0375 | 0.1479 | *0.0366* | *0.1442* | 0.0393 | 0.1506 | 0.0388 | 0.1503 | 0.0404 | 0.1523 | 0.0547 | 0.1778 | 0.0602 | 0.1921 | 0.0797 | 0.2205 |
| | 96 | **0.0519** | **0.1740** | 0.0521 | 0.1744 | 0.0550 | 0.1790 | 0.0519 | 0.1745 | 0.0551 | 0.1815 | 0.0786 | 0.2126 | 0.0919 | 0.2348 | 0.0857 | 0.2292 |
| | 192 | **0.0676** | **0.2013** | 0.0693 | 0.2034 | 0.0705 | 0.2050 | 0.0694 | 0.2046 | 0.0730 | 0.2076 | 0.0933 | 0.2344 | 0.1000 | 0.2464 | 0.0993 | 0.2428 |
| | 336 | 0.0803 | 0.2247 | *0.0796* | *0.2238* | 0.0814 | 0.2260 | 0.0826 | 0.2280 | 0.0948 | 0.2414 | 0.1117 | 0.2597 | 0.1418 | 0.2958 | 0.1287 | 0.2792 |
| | 720 | **0.0776** | **0.2224** | 0.0789 | 0.2244 | 0.0869 | 0.2329 | 0.0814 | 0.2273 | 0.1800 | 0.3494 | 0.1310 | 0.2858 | 0.1224 | 0.2766 | 0.1378 | 0.2939 |
| Illness | 24 | **0.5881** | **0.5358** | 0.6119 | *0.5299* | 0.6228 | 0.5305 | 0.6325 | 0.5639 | 0.7831 | 0.7462 | 0.6969 | 0.6256 | 0.7100 | 0.6352 | 0.7432 | 0.6704 |
| | 48 | **0.6858** | **0.6359** | 0.6925 | 0.6322 | 0.7109 | 0.6642 | 0.6892 | 0.6453 | 0.8217 | 0.7750 | 0.7099 | 0.6935 | 0.6961 | 0.6972 | 0.7855 | 0.7370 |
| | 60 | 0.6640 | 0.6423 | 0.6666 | *0.6324* | 0.6465 | 0.6381 | 0.6730 | 0.6347 | 0.9195 | 0.8361 | 0.8309 | 0.7653 | 0.8192 | 0.7641 | 0.8945 | 0.8055 |

Table 2: **Evaluating DTF-net in TSF task.** We conduct TSF experiments using three non-stationary datasets: Exchange Rate, ETTh1, and Illness. We evaluate performance using MSE and MAE, where lower values indicate better performance. In the following results, the best-performing models using DTF-net are highlighted in **bold**, and models using $\ell_1$ trend filtering are highlighted in *italic*. Additionally, for comparison, the best-performing models using only original data are underlined.

of noise is necessary to confirm smoothness. We divide the proximity evaluation into two categories: full-sequence and sub-sequence. In the sub-sequence evaluation, a 30-window interval is set around labeled abrupt changes to assess temporal dependencies are well captured.

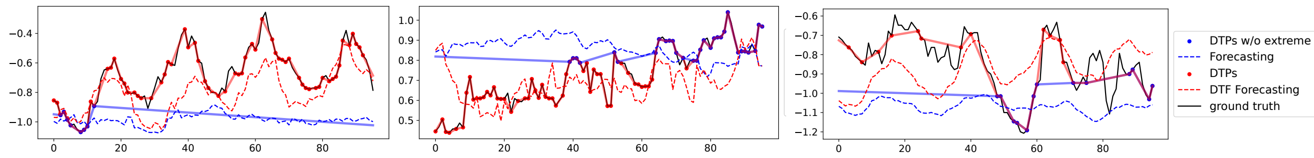
### Performance Analysis

Table 1 emphasizes DTF-net's superior performance compared to CPD and AD algorithms. CPD algorithms are designed to identify shifts in data distribution and tend to capture the midpoint of these changes, treating extreme values as outliers. On the other hand, AD algorithms focus exclusively on pinpointing anomalous data points, often overlooking the midpoint of changes. For instance, DCdetector is particularly adept at identifying abnormal values, demonstrating superior performance across entire data sequences. However, its effectiveness diminishes when dealing with abrupt sub-sequences. This shortfall stems from its focus solely on detecting abnormal points and short intervals surrounding abrupt changes. Consequently, while it maintains commendable performance on full sequences, it falls short in accurately filtering trends within abrupt sub-sequences.In essence, the differing objectives of CPD and AD algorithms make them less suitable for trend-filtering (Figure 7 in Appendix A).

In comparison with other trend filtering methods, DTF-net outperforms all methods except those prone to overfitting. Decomposition-based methods like EMD and Median generate excessively smoothed trends. The frequency-based method, Wavelet, tends to overfit to noise. The $\ell_1$ method shows sensitivity to hyperparameter $\lambda$, as evident by $\lambda = 5e-4$ causing overfitting to noise. Therefore, DTF-net excels in capturing abrupt changes while reflecting temporal dependencies within noisy and complex time series data.

### Nasdaq Dataset

To demonstrate the proficiency of DTF-net on complex real-world datasets, we perform additional analysis on the Nasdaq intraday dataset from July 30th to August 1st, 2019, characterized by rapid changes. Here, we arbitrarily set 6 abrupt changes and qualitatively analyze the results. As shown in Figure 4, it is evident that the $\ell_1$ trend filtering algorithm extracts trends that either underfit or overfit depending on the




(a) ETTh1 (24 pred) (b) EXC (24 pred) (c) ETTh1 (336 pred)

Figure 5: **Qualitative analysis of the impact of abrupt changes on TSF.** We conduct forecasting experiments to evaluate the influence of trends incorporating extreme values with long-heavy tails on two datasets, ETTh1 and Exchange rate (EXC). The figure illustrates that including abrupt changes (red) in forecasting plays a crucial role without undergoing smoothing (blue). It is evident that the results appear smoother when extreme values are excluded (depicted by the blue line) in both short-term (24 pred) and long-term (336 pred) forecasting.

parameter $\lambda$ due to constant smoothness. For point 3, in detail, $\ell_1$ with ($\lambda = 0.1$) filtered out noise, while ($\lambda = 5e-4$) captured it as abrupt changes. In contrast, DTF-net accurately captures five abrupt changes and concurrently performs noise filtering for point 3. This accomplishment is attributed to the dynamic nature of trend extraction from DTF-net.

## 4.2 Trend Filtering in Time Series Forecasting

### Experimental Settings

We analyze how DTF-net effectively captures abrupt changes and extend our evaluation to include a real-world dataset for a common time series task. Time Series Forecasting (TSF) models are expected to predict potential incidents associated with extreme values, providing valuable insights for critical decision-making [Van den Berg *et al.*, 2008]. To assess the practicality of DTF-net in real-world scenarios, we apply it to TSF, incorporating the extracted trend as an additional input feature. Formally, the forecasting model receives input as $\mathbf{X}' = [\mathbf{X}, \mathbf{P}] \in \mathbb{R}^{D+1}$, where $\mathbf{P}$ represents the trend from DTF-net. Under the same conditions, we compare this model to those using $\ell_1$ as additional inputs and only the original sequence $\mathbf{X}$ as inputs. We employ DTF-net with the TSF models NLinear and DLinear [Zeng *et al.*, 2023], which are considered state-of-the-art yet simplest in the field of TSF. The experiment focuses on the univariate forecasting case to assess trend filtering effectiveness (Appendix C). Note that DTF-net is not directly linked with TSF models; instead, the extracted trend from DTF-net is provided as additional input.

### Performance Analysis

We choose three non-stationary datasets from the TSF benchmark dataset: Exchange Rate, ETTh1, and Illness (Appendix C.1). Table 2 indicates that DTF-net outperforms in most cases. Among the three datasets, the exchange rate dataset is the most intricate, exhibiting the least seasonality and the highest level of noise. Given the absence of periodicity in financial data, $\ell_1$ trend filtering encounters difficulties in extracting clear trends. However, DTF-net demonstrates robustness when dealing with non-stationary time series data.

However, models employing $\ell_1$ trend filtering have advantages when dealing with more stationary data that exhibits a recursive pattern. The piece-wise linearity assumption of $\ell_1$ is particularly pronounced in short-term predictions within ETTh1, as it is the least noisy and most stationary dataset among the three. As shown in Table 2, $\ell_1$ achieves the best results for 24- and 48-hour forecasting windows in ETTh1. While DTF-net also outperforms the single forecasting model, the linearity characteristic of $\ell_1$ is better suited for short-term predictions within ETTh1. In contrast, for long-term predictions, we demonstrate that DTF-net performs the best. In the case of Illness with a small dataset size, DTF-net also performs well without overfitting.

### Ablation Study

**How to prevent RL overfitting?** To mitigate the risk of overfitting in RL-based trend filtering, we introduce a reward sampling method. As shown in Figure 9, we observe that reward sampling prevents overfitting, achieving optimal performance with a reward sampling (Appendix B.4).

**Empirical analysis on extreme value** DNNs often generate smooth and averaged predictions as they typically optimize forecasting performance through empirical risk minimization. However, by incorporating accurately captured abrupt changes as additional information into the model, predictions are enhanced while representing both upward and downward signals instead of providing solely smooth estimates. To qualitatively assess the impact of abrupt changes on forecasting tasks, we compare two different trends: the original trends from DTF-net (red) and a version where 10% of extreme values are excluded (blue). As shown in Figure 5, this comparison demonstrates how incorporating abrupt changes can enrich forecasting by providing more detailed and accurate predictions.

## 5 Conclusion

We propose DTF-net, a novel RL-based trend filtering method directly identifying trend points. Traditional trend filtering methods struggle to capture abrupt changes due to their inherent approximations. To address this, we formalize the Trend Point Detection problem as an MDP and utilize RL within a discrete action space. The reward function is defined as the sum-of-squares loss from forecasting tasks inspired by reward function learning within the Gaussian distribution, allowing for the capturing of temporal dependencies around DTPs. DTF-net also tackles overfitting issues through random sampling. Compared to other trend filtering methods, DTF-net excels in identifying abrupt changes. In forecasting tasks, DTF-net enhances predictive performance without compromising the prediction output to be smooth.

## Acknowledgments

This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation; No. 2022-0-00184, Development and Study of AI Technologies to Inexpensively Conform to Evolving Policy on Ethics; No. 2021-0-02068, Artificial Intelligence Graduate School Program (KAIST)).

## References

[Adams and MacKay, 2007] Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

[Altmann and Kantz, 2005] Eduardo G Altmann and Holger Kantz. Recurrence time analysis, long-term correlations, and extreme events. *Physical Review E*, 71(5):056106, 2005.

[Ansari *et al.*, 2021] Abdul Fatir Ansari, Konstantinos Benidis, Richard Kurle, Ali Caner Turkmen, Harold Soh, Alexander J Smola, Bernie Wang, and Tim Januschowski. Deep explicit duration switching models for time series. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS'21)*, 2021.

[Banerjee and Gelfand, 2003] Sudipto Banerjee and AE Gelfand. On smoothness properties of spatial processes. *Journal of Multivariate Analysis*, 84(1):85–100, 2003.

[Banerjee *et al.*, 2005] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, Joydeep Ghosh, and John Lafferty. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6(10), 2005.

[Biyik *et al.*, 2020] Erdem Biyik, Nicolas Huynh, Mykel J. Kochenderfer, and Dorsa Sadigh. Active preference-based gaussian process regression for reward learning. In *Proceedings of Robotics: Science and Systems (RSS'20)*, 2020.

[Caldarelli *et al.*, 2022] Edoardo Caldarelli, Philippe Wenk, Stefan Bauer, and Andreas Krause. Adaptive Gaussian process change point detection. In *Proceedings of International Conference on Machine Learning (ICML'22)*, 2022.

[Chan *et al.*, 2001] Tony F Chan, Stanley Osher, and Jianhong Shen. The digital tv filter and nonlinear denoising. *IEEE Transactions on Image processing*, 10(2):231–241, 2001.

[Chow, 1960] Gregory C Chow. Tests of equality between sets of coefficients in two linear regressions. *Econometrica: Journal of the Econometric Society*, pages 591–605, 1960.

[Craigmile and Percival, 2002] Peter F Craigmile and Donald B Percival. Wavelet-based trend detection and estimation. *Entry in the Encyclopedia of Environmetrics. Chichester, UK: John Wiley & Sons*, pages 2334–2338, 2002.

[Dickey and Fuller, 1979] David A Dickey and Wayne A Fuller. Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366a):427–431, 1979.

[Ding *et al.*, 2019] Daizong Ding, Mi Zhang, Xudong Pan, Min Yang, and Xiangnan He. Modeling extreme events in time series prediction. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'19)*, 2019.

[Fisher and Tippett, 1928] Ronald Aylmer Fisher and Leonard Henry Caleb Tippett. Limiting forms of the frequency distribution of the largest or smallest member of a sample. In *Mathematical Proceedings of the Cambridge Philosophical Society*, 1928.

[Geyer, 1992] Charles J. Geyer. Practical Markov Chain Monte Carlo. *Statistical Science*, 7(4):473 – 483, 1992.

[Gong and Huang, 2012] Rongsheng Gong and Samuel H. Huang. A kolmogorov-smirnov statistic based segmentation approach to learning from imbalanced datasets: With application in property refinance prediction. *Expert Syst. Appl.*, 39(6):6192–6200, may 2012.

[Hallac *et al.*, 2017] David Hallac, Sagar Vare, Stephen Boyd, and Jure Leskovec. Toeplitz inverse covariance-based clustering of multivariate time series data. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*, 2017.

[Han *et al.*, 2019] Jiyeon Han, Kyowoon Lee, Anh Tong, and Jaesik Choi. Confirmatory bayesian online change point detection in the covariance structure of gaussian processes. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'19)*, 2019.

[Hodrick and Prescott, 1997] Robert J Hodrick and Edward C Prescott. Postwar us business cycles: an empirical investigation. *Journal of Money, credit, and Banking*, pages 1–16, 1997.

[Kim *et al.*, 2009] Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. \ell_1 trend filtering. *SIAM review*, 51(2):339–360, 2009.

[Kuss and Rasmussen, 2003] Malte Kuss and Carl Rasmussen. Gaussian processes in reinforcement learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS'03)*, 2003.

[Leser, 1961] Conrad Emanuel Victor Leser. A simple method of trend construction. *Journal of the Royal Statistical Society: Series B (Methodological)*, 23(1):91–107, 1961.

[Liu *et al.*, 2022a] Xiao-Yang Liu, Ziyi Xia, Jingyang Rui, Jiechao Gao, Hongyang Yang, Ming Zhu, Christina Dan Wang, Zhaoran Wang, and Jian Guo. Finrl-meta: Market environments and benchmarks for data-driven financial reinforcement learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS'22)*, 2022.

[Liu *et al.*, 2022b] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Proceed-*

*ings of Advances in Neural Information Processing Systems (NeurIPS'22)*, 2022.

[Lorenz, 1963] Edward N Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2):130–141, 1963.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Mnih *et al.*, 2016] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of International Conference on Machine Learning (ICML'16)*, 2016.

[Nie *et al.*, 2023] Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *Proceedings of International Conference on Learning Representations (ICLR'23)*, 2023.

[Park *et al.*, 2020] Youngjin Park, Deokjun Eom, Byoungki Seo, and Jaesik Choi. Improved predictive deep temporal neural networks with trend filtering. In *Proceedings of ACM International Conference on AI in Finance (ICAIF '20)*, 2020.

[Ploberger and Krämer, 1992] Werner Ploberger and Walter Krämer. The cusum test with ols residuals. *Econometrica: Journal of the Econometric Society*, pages 271–285, 1992.

[Raj *et al.*, 2016] Vidwath Raj, Sven Magg, and Stefan Wermter. Towards effective classification of imbalanced data with convolutional neural networks. *Artificial Neural Networks in Pattern Recognition*, pages 150–162, 2016.

[Rosenblatt, 1956] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, pages 832–837, 1956.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[Siegel, 1982] Andrew F Siegel. Robust regression using repeated medians. *Biometrika*, 69(1):242–244, 1982.

[Sutton and Barto, 2018] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[Van den Berg *et al.*, 2008] Jeroen Van den Berg, Bertrand Candelon, and Jean-Pierre Urbain. A cautious note on the use of panel models to predict financial crises. *Economics Letters*, 101(1):80–83, 2008.

[von Bortkiewicz, 1921] Ladislaus von Bortkiewicz. *Variationsbreite und mittlerer Fehler*. Berliner Mathematische Gesellschaft, 1921.

[Wen *et al.*, 2019] Qingsong Wen, Jingkun Gao, Xiaomin Song, Liang Sun, and Jian Tan. Robusttrend: A huber loss with a combined first and second order difference regularization for time series trend filtering. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'19)*, 2019.

[Williams and Rasmussen, 1995] Christopher Williams and Carl Rasmussen. Gaussian processes for regression. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS'95)*, 1995.

[Wu and Ortiz, 2021] Tong Wu and Jorge Ortiz. Rlad: Time series anomaly detection through reinforcement learning and active learning. *ArXiv*, abs/2104.00543, 2021.

[Wu *et al.*, 2007] Zhaohua Wu, Norden E Huang, Steven R Long, and Chung-Kang Peng. On the trend, detrending, and variability of nonlinear and nonstationary time series. *National Academy of Sciences*, 104(38):14889–14894, 2007.

[Wu *et al.*, 2021] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS'21)*, 2021.

[Wu *et al.*, 2023] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *Proceedings of International Conference on Learning Representations (ICLR'23)*, 2023.

[Xu *et al.*, 2022] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. In *Proceedings of International Conference on Learning Representations (ICLR'22)*, 2022.

[Yang *et al.*, 2023] Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. Dcdetector: Dual attention contrastive representation learning for time series anomaly detection. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'23)*, 2023.

[Zeng *et al.*, 2023] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'23)*, 2023.

[Zhou *et al.*, 2021] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'20)*, 2021.

[Zhou *et al.*, 2022] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of International Conference on Machine Learning (ICML'22)*, 2022.