electronics

MDPI

Article

Temporal Attention for Few-Shot Concept Drift Detection in Streaming Data

Ximing Lin [1], Longtao Chang [1], Xiushan Nie [1] and Fei Dong [2,*]

[1] School of Computer Science and Technology, Shandong Jianzhu University, Jinan 250101, China; 2021115010@stu.sdjzu.edu.cn (X.L.); 20191101026@stu.sdjzu.edu.cn (L.C.); niexiushan19@sdjzu.edu.cn (X.N.)
[2] School of Journalism and Communication, Shandong Normal University, Jinan 250358, China
* Correspondence: 115046@sdnu.edu.cn

Abstract: Concept drift describes unforeseeable changes in the underlying distribution of streaming data over time. Concept drift is a phenomenon in which the statistical properties of a target domain change over time in an arbitrary way. These changes might be caused by changes in hidden variables that cannot be measured directly. With the onset of the big data era, domains such as social networks, meteorology, and finance are generating copious amounts of streaming data. Embedded within these data, the issue of concept drift can affect the attributes of streaming data in various ways, leading to a decline in the accuracy and performance of models. There is a pressing need for new models to re-adapt to the changes in streaming data. Traditional concept drift detection algorithms struggle to effectively capture and utilize the key feature points of concept drift within complex time series, thereby failing to maintain the accuracy and efficiency of the models. In light of these challenges, this study introduces a novel concept drift detection method that incorporates a temporal attention mechanism within a prototypical network. By integrating a temporal attention mechanism during the feature extraction process, our approach enhances the capability to process complex time series data, preserves temporal locality, strengthens the learning of key features, and reduces the amount of labeled data required. This method significantly improves the detection accuracy and efficiency of small sample streaming data by better capturing the local features of the data. Experiments conducted across multiple datasets demonstrate that this method exhibits comprehensive leading performance in terms of accuracy and F1-score, with excellent recall and precision, thereby validating its effectiveness in enhancing concept drift detection in streaming data.

Keywords: concept drift; prototypical network; temporal attention mechanism; meta metric learning

check for updates

Citation: Lin, X.; Chang, L.; Nie, X.; Dong, F. Temporal Attention for Few-Shot Concept Drift Detection in Streaming Data. *Electronics* **2024**, *13*, 2183. https://doi.org/10.3390/electronics13112183

Academic Editor: Ping-Feng Pai

Received: 2 May 2024
Revised: 21 May 2024
Accepted: 25 May 2024
Published: 3 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Concept drift presents significant challenges to supervised learning tasks, as it leads to changes in the statistical properties of the data generation process, thereby affecting the accuracy and robustness of predictive models. Consequently, the task of detecting concept drift has emerged as an important research topic in the fields of machine learning and data mining. However, concept drift detection faces multiple challenges, including but not limited to how to accurately and timely identify the occurrence of concept drift, how to distinguish genuine concept drift from noise or random fluctuations, and how to effectively adapt to concept drift with limited labeled data. Furthermore, detection algorithms must minimize false alarms while maintaining high detection sensitivity to avoid frequent and unnecessary model updates. Due to the rarity and diversity of concept drift, its detection remains a challenging task. Early research primarily focused on manually extracting features to model normal and concept drift events, classifying the phenomenon of concept drift, and completing detection tasks [1].

With the advancement of synthetic intelligence technologies, significant progress has been made in the study of concept drift detection [2]. Mainstream detection methods

typically rely on labeled data to identify changes in data distribution, assuming that each sample in datasets collected at different time points has a corresponding label. By comparing the model's performance on the most recent data with its historical performance, concept drift can be detected. However, in actual production scenarios, the distribution of the data is not static but changes over time [3]. This leads to the inadequacy of the original offline-trained models to adapt to new test data distributions after a certain period, necessitating timely detection of concept drift and model updates to ensure model accuracy and robustness [4]. Although these methods can identify changes in data distribution to some extent, they usually cannot simultaneously detect concept drift and identify the types of concept drift.

To address the limitations of traditional supervised learning methods, this work proposes a Temporal Attention mechanism in a Prototypical Network for Concept Drift Detection (TAPN-CDD). By incorporating a temporal attention mechanism during feature extraction, this method enhances the ability to process complex time series data, preserves temporal locality, and strengthens the learning of key features. It requires a smaller amount of labeled data, better captures the local features of the data and improves the detection accuracy and efficiency of small sample streaming data. Experiments across multiple datasets have demonstrated the effectiveness of this method in enhancing the performance of concept drift detection in streaming data.

The main contributions of this paper are threefold:

- We utilized a prototypical network for concept drift detection and identification and employed artificially derived classification prototypes as comparison points. The pending time series streaming data were inputted, and the prototypical network conducted the pending time series classification to obtain the predicted type based on which concept drift is detected.
- We designed a temporal attention module to learn the relative importance of different instances in various streaming data for concept drift detection, thereby dynamically allocating attention weights. It enhanced the capability to process complex time series data, preserves temporal locality, and strengthens the learning of key features.
- We employed meta-learning methods to reduce the demand for labeled data streams, automatically optimized the parameters of the prototypical network and attention module to achieve optimal model classification performance, and effectively improved the efficiency of concept drift detection on small sample streaming data.

The rest of this paper is organized as follows: Section 2 introduces related work. Section 3 provides details of our proposed TAPN-CDD framework. Section 4 discusses the experimental results on several synthetic datasets as well as real-world data. Finally, Section 5 concludes the paper and presents future work.

2. Related Work

2.1. Overview of Concept Drift

As machine learning models become an increasingly popular solution for automation and predictive tasks, many tech companies and data scientists have adopted the following work paradigm: data scientists are responsible for solving a specific problem, they are given a snapshot of the available relevant data, and they work on training the model to solve it. Once the model is tested, it goes into production. Eventually, the performance of the model starts to degrade, which is usually due to concept drift.

Concept drift is a situation where the statistical properties of the target variable (what the model is trying to predict) change over time in unforeseen ways. It refers to the phenomenon that the statistical properties of the output target change randomly over time in an arbitrary manner [5], which refers to an online supervised learning scenario in which the relationship between the input data and the output target varies over time. After Schlimmer et al. [6] proposed concept drift for the first time in 1986, data mining researchers at home and abroad have carried out in-depth studies on concept drift, respectively. Nowadays, concept drift has become a research hotspot in the field of data mining,

and when the predictive model encounters concept drift, the predictive model should be dynamically adjusted in order to respond appropriately to concept drift [7].

Concept drift is categorized into the following two types, which are shown in Figure 1:

- Virtual drift: $p(X)$ changes but $p(Y|X)$ does not. This means that the underlying distribution of features has changed but the performance of the model has not.
- Real drift: $p(Y|X)$ has changed, which means the performance of the model has changed.

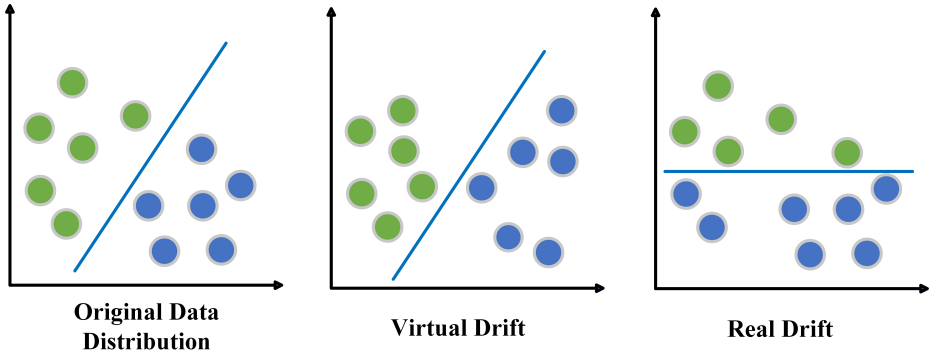

Figure 1. An example of concept drift categories (the green circles represent data labeled as y_0 at time t, the blue circles represent data labeled as y_1 at time t, and the blue line represents the decision boundary).

Besides, concept drift can occur in different ways in data distribution speed, which are shown in Figure 2.

- Sudden drift: a new concept appears in a short period of time.
- Gradual drift: a new concept gradually replaces an old concept over a long period of time.
- Incremental drift: an old concept gradually becomes a new concept over a period of time.
- Recurring concepts: an old concept may reappear after a period of time.

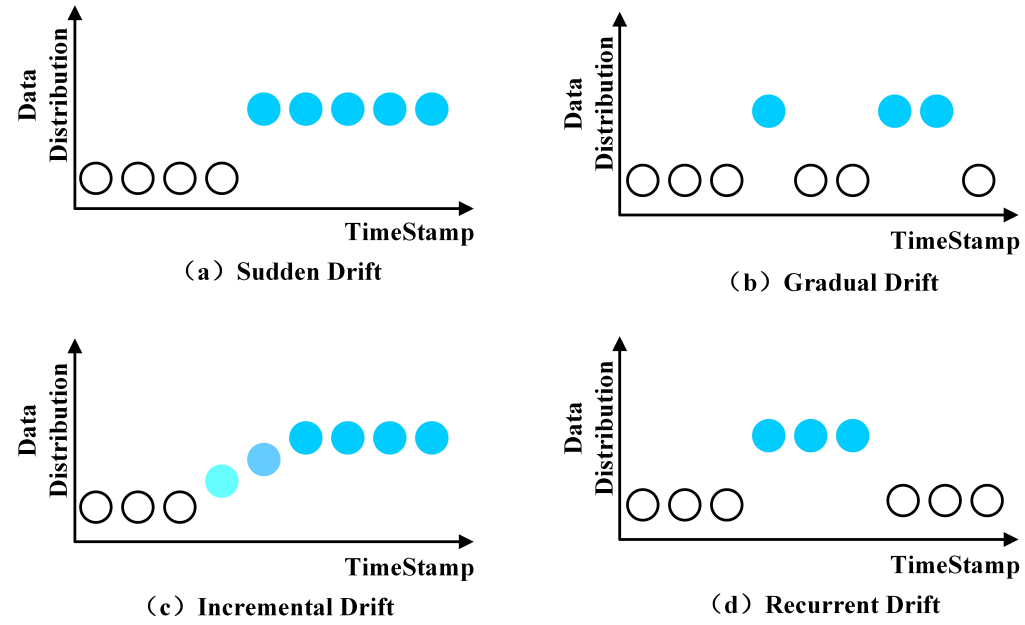

Figure 2. An example of concept drift types (the white circles represent the original data distribution, and the blue circles represent the new data distribution).

2.2. Model Performance-Based Drift Detection

As time passes, concept drift can be detected by monitoring the accuracy of the model. If there is a time period where the model performed better than after, it is evidence that the nature of the data stream has changed. In such cases, the model should be retrained with the data after the change. This method or its variations are the most commonly used strategy for detecting concept drift [7]. Gama et al. [1] defined the warning and drift threshold, categorizing them as warning and drift levels, and proposed the drift detection method (DDM). Using a window that increases over time to collect instances of data, DDM calculates the error rate of the current window when it is full; if the change of the error rate reaches the confidence level of the warning threshold, DDM retrains a new model with the data in the window while continuing the prediction with the old model. If the error rate change reaches the drift threshold, the old model is replaced with the new model. Based on DDM, Baena-Garcıa et al. [8] proposed the Early Drift Detection Method (EDDM), which used the same time window strategy and counted the distance between two consecutive prediction errors, i.e., the number of instances between two consecutive errors, to detect concept drift when the distance between two consecutive errors increases due to stable concept. Bifet et al. [9] proposed Adaptive Windowing (ADWIN), which defines two sub-windows for new and old data with an adaptive change in size within the total fixed-size data window. If there is no concept drift, the size of the new data sub-window increases, and vice versa. When the difference in error rates between the two sub-windows exceeds the given drift threshold, it indicates a concept drift phenomenon.

2.3. Data Distribution-Based Drift Detection

Concept drift detection methods based on data distribution use distance functions or distance measures to quantify the difference between the historical data and the new data distribution. If the difference is proven to be statistically significant, the prediction or decision-making model will be retrained [2]. Kifer et al. [10] first proposed this strategy.

2.4. Multiple Hypothesis Test-Based Drift Detection

Multiple hypothesis testing drift detection algorithms also apply similar techniques as mentioned in the previous two categories. However, different from others, multiple hypothesis testing drift detection algorithms use multiple hypothesis testing in different ways to detect concept drift during the hypothesis testing stage. For example, Yu et al. [11] proposed the Hierarchical Linear Four Rates (HLFR) framework, which detects concept drift from different data stream distributions, including imbalanced data, by using a set of hierarchical hypothesis tests in an online environment.

2.5. Meta Learning-Based Drift Detection

In recent years, with the increasing application of meta-learning in various fields, Yu et al. [12] proposed Meta-ADD, a meta-learning-based pre-trained model for concept drift active detection. Meta-ADD uses a prototypical network of metric learning similarity networks for concept drift detection. During pre-training, prototype centers are extracted for normal data streams and sudden, gradual, and incremental concept drifts. During detection, input data are measured for similarity with the prototype centers for both concept drift detection and type discrimination. Lin et al. [13] proposed the Meta-ADPTF method, which utilizes Transformers as the prototypical network for embedding space neural networks and adaptive window meta metric learning for detecting concept drift. It significantly enhances accuracy and robustness.

3. Proposed Method

In this section, to address the issue that traditional mainstream concept drift detection methods cannot effectively capture and utilize the key feature points of concept drift within complex time series, thereby maintaining the accuracy and efficiency of the model, a new network named TAPN-CDD is designed. It consists of a prototype network integrated

with a temporal attention module aimed at better capturing the local features of data and enhancing the capability to process time series stream data [14], as shown in Figure 3. Initially, a temporal attention module is devised to learn the relative importance of different instances within diverse streaming data for concept drift detection, thereby dynamically allocating attention weights. This attention module is incorporated into the feature extraction function, adjusting the original feature representation to obtain a weighted feature representation for computing the distance to the prototype. During the training phase, artificially generated normal streaming data along with streaming data containing various types of concept drift are used as the training dataset. These are input into the prototypical network to generate prototype representations. Through meta-learning, the parameters of the prototypical network and attention module are automatically optimized to achieve optimal classification performance.

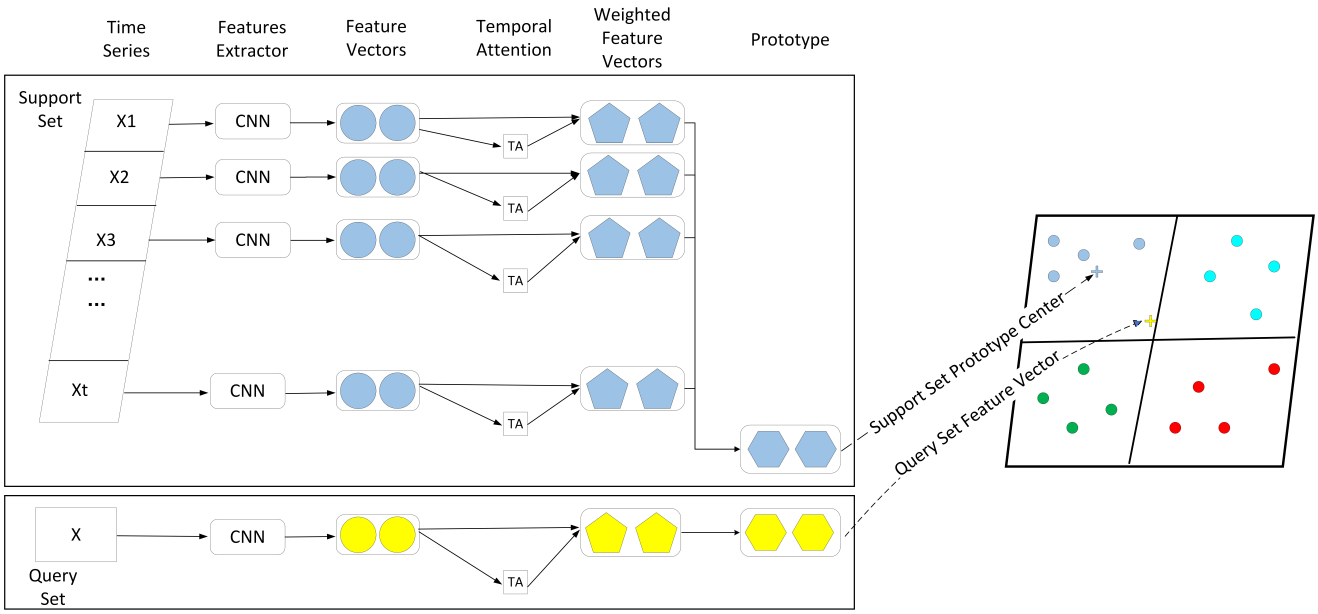

Figure 3. The framework of TAPN-CDD (at the left side, blue represents the support set's vectors, while yellow represents the query set's vectors; at the right side, different colors represent different prototypes, which are normal data without drift, sudden drift data, gradual drift data and incremental drift data).

In the detection phase, real test datasets are fed into the trained prototypical network to obtain the weighted feature representations at various time points. These representations are then used to calculate distances to the prototype centers, and the output categories are classified accordingly to detect concept drift. This method not only preserved temporal locality but also enhanced the learning of key features through the attention mechanism with a reduced requirement for data quantity, effectively improving the efficiency of concept drift detection in small sample streaming data.

3.1. Data Pre-Processing

In this section, we proposed a method for concept drift feature extraction that can overcome the variances in data streams from different sources. Given the periodic nature of recurrent drifts, the focus is on identifying whether old concepts re-emerge, rather than emphasizing the immediate detection of concept drift to adapt models to new concepts in a production environment, without concern for whether the current concept is an old one. Thus, our work necessitated the extraction of features from normal data without drift, and from data exhibiting sudden, gradual, and incremental concept drifts. Changes in data distribution can be represented by changes in error rates. Therefore, intuitively, the alteration in error rates is crucial for determining the type of concept drift that has occurred.

During the pre-training phase, this work utilized the MOA model to generate data streams that individually exhibit sudden, gradual, incremental, and normal (no drift) characteristics through data generators such as SEA, HYP, AGR, etc. Methods such as Naive Bayes classifiers, decision trees, SVM, or SGD gradient descent classification are employed to classify and predict N generated data streams. The prediction classification error rate at each moment, e_t, is obtained, and a fixed-size sliding time window W of size n is used to accommodate these error rates, resulting in an error rate sequence X_i. The error rate of each data stream can be accommodated by t time windows, thus yielding t error rate sequences. The set $\{X, y\}$, where $X = \{X_1, \ldots, X_l\}$ represented the error rate sequences and $y \in \{1, 2, 3, 4\}$ denoted the category labels of the data streams corresponding to the error rate sequences, is taken as the feature to be extracted for this work.

3.2. Features Extraction

In this section, a CNN architecture incorporating one-dimensional convolutional layers was designed for processing time series data. This architecture aimed to capture local dependencies and patterns within the time series. Following the convolutional layers, pooling layers were added to reduce the dimensionality of the features and extract key features. This approach contributed to a reduction in computational load and prevented overfitting. Furthermore, the ReLU activation function was employed after the convolutional layers to introduce non-linearity, thereby enhancing the model's expressive capacity.

Time series data were input into the CNN, where features are automatically extracted through convolutional layers. Each convolutional layer learns different temporal patterns and features. The convolution operations produce feature maps, which, after being processed through activation functions, highlight important features within the data.

Consider a time series $X = \{X_1, X_2, \ldots, X_T\}$, where X_t is the observation at time point t and T is the total length of the series. Feature extraction was performed on the observation X_t at each time point, yielding the feature vector h_t as follows:

$$h_t = f_\theta(X_t) \tag{1}$$

where f represents the CNN network and θ denotes the prototypical parameters of the CNN, such as the size of the convolution kernels. From this, the feature sequence $\{h_1, h_2, \ldots, h_T\}$ was extracted.

3.3. Temporal Attention Weights

For each time point t, an attention weight α_t is calculated to reflect the importance of that time point. Based on the extracted feature sequence $\{h_1, h_2, \ldots, h_T\}$, the time attention weight for the feature vector h_t at the current time point is derived from the features of previous time points. The calculation of time attention weights α_t is as follows:

$$\alpha_t = \frac{exp(g(h_t, h_T))}{\sum_{i=1}^{T-1} exp(g(h_i, h_T))} \tag{2}$$

where $g(h_t, h_T)$ represented a scoring function used to evaluate the importance of the feature vector h_t at time point t relative to the feature vector h_T at the latest time point T.

The calculation of α_t is based on the softmax function, which normalizes the exponent of $g(h_t, h_T)$ against the sum of the exponents for all time points, resulting in a weight that ranges between 0 and 1. This weight reflects the relative importance of the feature vector at time point t within the entire time series, where a higher α_t value indicates that the time point t is more important. Therefore, by assigning different weights α_t to each time point in the time series, the model can focus on those time points that are most critical for the current task, ignoring those that are less important or more noisy. Additionally, α_t can be dynamically adjusted to reflect the importance of the latest data, helping the model to adapt in real-time to changes in the data, thereby enhancing the model's flexibility and adaptability.

3.4. Drift Detector via Temporal Attention Mechanism

The training of the prototypical network with temporal attention comprises two parts. First, the embedding function CNN network with learnable prototype parameters θ maps the data stream X_i into the embedding space $f_\theta(X_i)$, yielding the embedding vector $h_i = f_\theta(X_i)$. Next, time attention is calculated for the embedding vector, and the resulting attention weight α_i is multiplied by the feature vector h_i to obtain a weighted feature representation. The weighted feature representations of all the time points are summed to produce the final weighted vector z, which encapsulates the information of the entire streaming data and places greater emphasis on those time points assigned higher weights, as follows:

$$z = \sum_{i=1}^{T} \alpha_i h_i \tag{3}$$

The prototype center c_k for each streaming data is the mean of their respective weighted vectors, as follows:

$$c_k = \frac{1}{T} z \tag{4}$$

where T is the length of the support set data stream. Subsequently, the Euclidean distance is employed as the distance function d, mapping the query set into the embedding space and applying temporal attention weighting to obtain the query point q. The nearest prototype center c_k to the query point is identified, and classification is performed using Softmax as follows:

$$p_\theta(y = k|q) = \frac{exp(-d(z_\theta(q), c_k))}{\sum_{k'} exp(-d(z_\theta(q), c_{k'}))} \tag{5}$$

Hence, we use the negative log-likelihood function:

$$J_\theta = -\log p_\theta(y = k|q) \tag{6}$$

as the objective function for the drift detector, optimizing the drift detector by minimizing the negative log-likelihood function. Through iterative optimization of c_k and $J_\$theta$, a drift detector containing globally optimal prototype parameters θ is obtained.

During the concept drift detection phase, the streaming data to be analyzed are fed into the time window W_t for capture. Data stream features corresponding to the window are obtained and divided into a support set and a query set. The support set is input into the prototypical network for feature extraction and weighting. After a few iterations, a drift detector equipped with the current streaming data's prototype center c_k and the optimized prototype parameters θ' is obtained. The query set is then passed into the drift detector, which outputs the distance comparison between the query points and the prototype center c_k, resulting in the classification outcome and completing the concept drift detection task.

4. Experiments

In this chapter, a comparison was made with nine different approaches. In this section, an evaluation of the proposed TAPN-CDD is conducted. This work employed the SGD gradient descent classification method and traditional concept drift detection algorithms such as DDM [1], EDDM [8], ADWIN [9], HDDM_A [15], HDDM_W [15], KSWIN [16], and Page–Hinkley [17], in addition to meta-metric learning concept drift detection algorithms like Meta-ADD [12] and Meta-ADPTF [13], as comparative methods.

4.1. Datasets

Since the data instances are generated by predefined rules and specific parameters, synthetic datasets are a good choice for evaluating the performance of learning algorithms in different concept drift scenarios. The dataset used in this work consists of five synthetic datasets along with seven publicly available real-world datasets. Table 1. provides detailed information corresponding to synthetic and real-world datasets.

Table 1. Synthetic and real-world datasets.

Datasets	Number of Instances	Number of Features	Number of Classes
SEA	Random	3	2
Hyperplane	Random	10	2
AGR	Random	6	3
RTG	Random	Random	Random
RandomRBF	Random	Random	Random
Airline	539,395	8	2
CovType	581,012	54	7
PokerHand	1,025,010	10	10
Spam	9324	499	2
Electricity	45,312	8	2
USEP	87,648	6	2
Renewables	52,584	6	2

SEA [18] dataset's each sample has three feature dimensions and two classes. Its source is Real Drift, which detects bursty Concept drift.

Hyperplane [19] dataset's each sample has 10 feature dimensions and two classes. Its source is real drift and can detect gradual vs. incremental concept drift.

The AGR [20] dataset is a common source of data in early work on decision tree learning. Each sample contains six feature dimensions and three classes.

The RTG [21] dataset is constructed by randomly selecting attributes for segmentation and assigning a random class label to each leaf. Once the tree is constructed, new examples are generated by assigning uniformly distributed random values to the attributes and then the class labels are determined through the tree. Once the tree is constructed, new examples are generated by assigning uniformly distributed random values to the attributes and then the class labels are determined through the tree.

The RandomRBF [22] dataset can randomly generate the total number of samples, feature dimensions, and number of classes. It is sourced from a mixture of drifts and can be used to detect bursty, progressive, and incremental Concept drifts.

The Airline [23] dataset contains 539,395 samples, each with eight feature dimensions. The dataset contains a large number of records that include detailed information on the arrival and departure of all commercial flights within the United States from October 1987 to April 2008.

The CovType [24] dataset, which has a total of 581,012 samples, has 54 features per sample, with seven classes, where all but the first 10 of the 54 feature dimensions are floating-point numbers, and the rest are One-hot variables.

The PokerHand [24] dataset, which can be used to detect concept drift in category imbalance, contains 1,025,010 samples, each containing 10 feature dimensions and 10 classes.

The spam [25] dataset, which is mainly used for progressive drift detection, has 9324 samples containing 499 feature dimensions and two classes.

The electricity [1] dataset is a widely used dataset collected from the South Wales electricity market of Australia. Due to market supply-demand dynamics, the electricity price in this dataset is not fixed and is collected every five minutes. The electricity dataset consists of 45,312 instances from May 1996 to December 1998. Each instance comprises eight fields - date, day of the week, timestamp, electricity prices and demands for New South Wales and Victoria, planned power transmission between states, and a categorical label. The label indicates the change in the current electricity price with respect to the 24-h moving average of past prices and comprises two categories: up and down.

The USEP [26] dataset is the uniform energy price for settlement purposes in Singapore, which applies to all energy injections or extractions that are deemed to have occurred at the Singapore hub. It is the weighted average of nodal prices of all purchasing nodes within every half hour. The USEP dataset consists of 87,648 instances from January 2018 to December 2022. The label indicates the change in the current price with respect to the 24-h moving average of past prices and comprises two categories, up and down.

The renewables [27,28] dataset is a dataset provided by Renewables.ninja that provides hourly simulation data for wind and photovoltaic power generation for all European countries using historical weather conditions. The renewables dataset consists of 52,584 instances from January 2014 to December 2019. The label indicates the change in wind and photovoltaic power generation with respect to the 24-h moving average of past values and comprises two categories, up and down.

4.2. Evaluation Indicators

In this paper, we evaluate the performance of the defect prediction models using commonly used evaluation metrics: accuracy(Acc) and F1-score (F1). Accuracy refers to the ratio of correctly classified predictions to the total number of predictions made. The higher the accuracy, the better the performance of the classifier. F1-score is a measure of classification performance, computed as the harmonic mean of precision (P) and recall (R), and reflects the robustness of the model. The F1-score ranges from 0 to 1, where a value closer to 1 indicates a better predictive performance of the model.

4.3. Experiment Setting

In this experimental section, 4000 data streams were artificially generated each containing 2000 samples to simulate the prediction error of the learning model, with each data stream simulating a single type of change. Subsequently, 800 data streams were selected from each type of change, totaling 3200, to serve as the initial training dataset for feature extraction. The remaining data streams were mixed to form the test set. For the prototypical network architecture, the embedding function utilized was a CNN containing two convolutional layers. The time window for collecting time series data was set to 50 time points. During the training process, for each type of concept change, five data streams were randomly selected as the support set, and 15 data streams were used to evaluate model efficacy. In the change detection phase, the trained support data streams were used as the final prototypes, with the real dataset serving as the query set. The prototypical network was trained using the Adam optimization algorithm, with the learning rate set to 10^{-4}.

4.4. Experimental Results and Analysis

We compared the performance of the TAPN-CDD model with that of traditional concept drift detection methods and meta-learning drift detection methods.

It can be observed that in the synthetic datasets, the experimental results, as shown in Table 2, indicate that the accuracy of the prototypical network TAPN-CDD, which incorporates a temporal attention module, increased by 3% to 5% compared to Meta-ADD and Meta-ADPTF, with a precision improvement of 0.02 to 0.04. This represents a significant improvement over traditional concept drift detection algorithms.

In real-world datasets, as illustrated in Table 3, the TAPN-CDD demonstrated an improvement in both accuracy and precision compared to Meta-ADD and Meta-ADPTF, marking a significant enhancement over traditional concept drift detection algorithms. The experiments have shown that our proposed TAPN-CDD has significantly improved accuracy and robustness compared with previous detection methods.

Table 2. Results of performance under different methods in synthetic datasets.

Methods	SEA		Hyperplane		AGR		RTG		RandomRBF	
	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1
*	70.87	0.814	75.93	0.764	42.67	0.372	55.10	0.624	57.13	0.530
*+DDM	73.10	0.802	79.20	0.787	44.42	0.445	55.95	0.650	61.95	0.590
*+EDDM	72.55	0.797	78.03	0.793	43.01	0.429	55.46	0.655	61.41	0.584
*+ADWIN	74.48	0.812	78.47	0.798	45.19	0.452	56.24	0.652	62.40	0.594

Table 2. *Cont.*

Methods	SEA		Hyperplane		AGR		RTG		RandomRBF	
	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1
*+HDDM_A	74.28	0.811	78.41	0.798	45.68	**0.457**	55.44	0.656	62.03	0.593
*+HDDM_W	74.13	0.809	78.65	0.800	45.17	0.439	56.20	0.650	61.52	0.585
*+KSWIN	73.63	0.814	77.79	0.792	**46.22**	0.437	55.70	0.657	61.76	0.589
*+Page-Hinkley	73.92	**0.816**	77.83	0.795	44.39	0.443	55.83	0.659	62.37	0.593
*+Meta-ADD	74.99	0.813	78.65	0.796	44.50	0.446	57.44	0.660	62.51	0.595
*+Meta-ADPTF	76.55	0.814	79.21	0.802	45.12	0.450	57.50	0.662	63.70	0.601
*+TAPN-CDD	**76.99**	0.813	**83.72**	**0.841**	44.30	0.442	**60.76**	**0.697**	**66.32**	**0.631**

* represents SGD classification method, the bolded data represents the best performance.

Table 3. Results of performance under different methods in real-world datasets.

Methods	Airline		CovType		PokerHand		Spam		Electricity		USEP		Renewables	
	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1	Acc (%)	F1
*	55.61	0.487	90.40	0.441	45.36	0.316	82.31	0.894	69.52	0.714	81.48	0.863	67.77	0.619
*+DDM	57.48	**0.524**	91.09	0.909	50.25	0.453	91.21	0.931	74.88	0.786	84.57	0.868	76.91	0.708
*+EDDM	**57.51**	0.523	91.40	0.913	50.27	0.453	92.31	0.932	75.22	0.791	87.74	0.903	74.18	0.699
*+ADWIN	57.50	0.523	91.41	0.914	50.23	0.453	92.55	0.934	75.53	0.797	81.94	0.840	71.38	0.702
*+HDDM_A	57.44	0.522	91.41	0.913	50.30	0.454	91.39	0.932	75.40	0.794	86.51	0.896	76.88	0.703
*+HDDM_W	56.71	0.514	91.25	0.907	50.29	0.458	92.60	0.934	74.91	0.787	86.27	0.894	77.34	0.648
*+KSWIN	56.92	0.515	91.32	0.909	50.20	0.456	92.48	0.933	75.83	0.801	87.38	0.901	75.17	0.588
*+Page-Hinkley	57.50	0.523	91.39	0.913	50.26	0.457	92.64	0.934	76.34	0.811	87.26	0.900	71.93	0.704
*+Meta-ADD	57.38	0.519	91.40	0.912	50.29	**0.459**	93.48	0.934	82.55	0.849	90.06	0.915	84.81	0.849
*+Meta-ADPTF	57.43	0.522	91.46	0.922	50.34	0.456	93.53	**0.934**	83.88	0.859	91.65	0.927	88.34	**0.883**
*+TAPN-CDD	57.45	0.522	**92.28**	**0.927**	**50.37**	0.455	**93.60**	0.931	**84.00**	**0.861**	**92.21**	**0.934**	**88.76**	0.858

* represents SGD classification method, the bolded data represents the best performance.

5. Conclusions and Future Work

5.1. Conclusions

Addressing the issue that traditional mainstream concept drift detection methods cannot effectively capture and utilize key feature points of concept drift within complex time series, thereby maintaining the model's accuracy and efficiency, this work proposes a prototypical network concept drift detection method based on a temporal attention mechanism. Compared to previous concept drift detection algorithms, this method, grounded in a prototypical network, learns the optimal prototype parameters for data from different source domains through a few iterations, quickly resulting in a dedicated drift detector for each. This significantly enhances the generalizability of the concept drift detection model. By incorporating a temporal attention mechanism during feature extraction and employing attention-weighted summation when determining the prototype center, the model adaptively focuses on the most important time points within the stream data. This improves the capability to process complex time series stream data, effectively addressing the issue of not being able to utilize the most relevant time points or periods for the current stream data. It better captures the local features of the data, allowing for more timely and accurate detection of concept drift and model updates, thereby significantly enhancing the model's detection accuracy and efficiency.

5.2. Future Work

While this method has demonstrated excellent performance across multiple datasets, it still exhibits some limitations. For instance, if input data contain imbalances or similar issues, the attention mechanism might exacerbate the impact of these problems. Additionally, the introduction of a temporal attention layer increases the number of parameters and the computational complexity of the model, which could lead to higher training costs and durations. Moreover, although the attention weights provide a form of interpretability by indicating which time points are more critical, the overall decision-making process of the model remains challenging to interpret, especially within complex network architectures. Future work should, therefore, focus on more in-depth studies of interpretability. Research could explore ways to enhance the transparency of the decision-making process within attention mechanisms, making it easier for users to understand the model's behavior. Developing more efficient training algorithms to reduce the computational burden caused by increased model complexity is also essential. Additionally, when designing models with temporal attention, consideration should be given to optimizing energy efficiency and computational resources, particularly for applications on mobile devices and edge computing scenarios.

Author Contributions: X.L.: Conceptualization, methodology, writing—original draft, L.C.: Validation, writing—review & editing, X.N.: Conceptualization, Supervision, project administration, F.D.: Conceptualization, resource, project administration. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gama, J.; Medas, P.; Castillo, G.; Rodrigues, P. Learning with drift detection. In Proceedings of the SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence—Advances in Artificial Intelligence, Sao Luis, Brazil, 29 September–1 Ocotber 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 286–295.
2. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under concept drift: A review. *IEEE Trans. Knowl. Data Eng.* **2018**, *31*, 2346–2363. [CrossRef]
3. Korycki, Ł.; Krawczyk, B. Concept drift detection from multi-class imbalanced data streams. In Proceedings of the 2021 IEEE 37th International Conference on Data Engineering (ICDE), Chania, Greece, 19–22 April 2021; pp. 1068–1079.
4. Sato, D.M.V.; De Freitas, S.C.; Barddal, J.P.; Scalabrin, E.E. A Survey on Concept Drift in Process Mining. *ACM Comput. Surv.* **2021**, *54*, 1–38. [CrossRef]
5. Farsi, B.; Amayri, M.; Bouguila, N.; Eicker, U. On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM-CNN approach. *IEEE Access* **2021**, *9*, 31191–31212. [CrossRef]
6. Schlimmer, J.C.; Granger, R.H. Incremental learning from noisy data. *Mach. Learn.* **1986**, *1*, 317–354. [CrossRef]
7. Huggard, H.; Koh, Y.S.; Dobbie, G.; Zhang, E. Detecting concept drift in medical triage. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 1733–1736.
8. Baena-Garcıa, M.; del Campo-Ávila, J.; Fidalgo, R.; Bifet, A.; Gavalda, R.; Morales-Bueno, R. Early drift detection method. In Proceedings of the Fourth International Workshop on Knowledge Discovery from Data Streams, Berlin, Germany, 18–22 September 2006; Volume 6, pp. 77–86.
9. Bifet, A.; Gavalda, R. Learning from time-changing data with adaptive windowing. In Proceedings of the 2007 SIAM International Conference on Data Mining, Minneapolis, MN, USA, 26–28 April 2007; pp. 443–448.
10. Kifer, D.; Ben-David, S.; Gehrke, J. Detecting change in data streams. In Proceedings of the VLDB 2004, Toronto, ON, Canada, 31 August 2004–3 September 2004; Volume 4, pp. 180–191.
11. Yu, S.; Abraham, Z. Concept drift detection with hierarchical hypothesis testing. In Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, TX, USA, 27–29 April 2017; pp. 768–776.
12. Yu, H.; Zhang, Q.; Liu, T.; Lu, J.; Wen, Y.; Zhang, G. Meta-ADD: A meta-learning based pre-trained model for concept drift active detection. *Inf. Sci.* **2022**, *608*, 996–1009. [CrossRef]
13. Lin, X.; Nie, X.; Dong, F.; Guo, J. A Concept Drift Detection Method for Electricity Forecasting Based on Adaptive Window and Transformer. In Proceedings of the 2023 IEEE Smart World Congress (SWC), Portsmouth, UK, 28–31 August 2023; pp. 1–7.
14. Fan, J.; Zhang, K.; Huang, Y.; Zhu, Y.; Chen, B. Parallel spatio-temporal attention-based TCN for multivariate time series prediction. *Neural Comput. Appl.* **2023**, *35*, 13109–13118. [CrossRef]

15. Frías-Blanco, I.; del Campo-Ávila, J.; Ramos-Jimenez, G.; Morales-Bueno, R.; Ortiz-Diaz, A.; Caballero-Mota, Y. Online and non-parametric drift detection methods based on Hoeffding's bounds. *IEEE Trans. Knowl. Data Eng.* **2014**, *27*, 810–823. [CrossRef]
16. Raab, C.; Heusinger, M.; Schleif, F.M. Reactive soft prototype computing for concept drift streams. *Neurocomputing* **2020**, *416*, 340–351. [CrossRef]
17. Page, E.S. Continuous inspection schemes. *Biometrika* **1954**, *41*, 100–115. [CrossRef]
18. Street, W.N.; Kim, Y. A streaming ensemble algorithm (SEA) for large-scale classification. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001; pp. 377–382.
19. Hulten, G.; Spencer, L.; Domingos, P. Mining time-changing data streams. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001; pp. 97–106.
20. Agrawal, R.; Imielinski, T.; Swami, A. Database mining: A performance perspective. *IEEE Trans. Knowl. Data Eng.* **1993**, *5*, 914–925. [CrossRef]
21. Domingos, P.; Hulten, G. Mining high-speed data streams. In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, 20–23 August 2000; pp. 71–80.
22. Bifet, A.; Holmes, G.; Pfahringer, B.; Kranen, P.; Kremer, H.; Jansen, T.; Seidl, T. Moa: Massive online analysis, a framework for stream classification and clustering. In Proceedings of the First Workshop on Applications of Pattern Analysis, Windsor, UK, 1–3 September 2010; pp. 44–50.
23. Wickham, H. ASA 2009 data expo. *J. Comput. Graph. Stat.* **2011**, *20*, 281–283. [CrossRef]
24. Asuncion, A.; Newman, D. *UCI Machine Learning Repository*; School of Information and Computer Sciences, University of California: Irvine, CA, USA, 2007.
25. Katakis, I.; Tsoumakas, G.; Vlahavas, I.J.K.; Systems, I. Tracking recurring contexts using ensemble classifiers: An application to email filtering. *Knowl. Inf. Syst.* **2010**, *22*, 371–391. [CrossRef]
26. NEMS Prices. Available online: https://www.nems.emcsg.com/nems-prices#/ (accessed on 1 May 2024).
27. Pfenninger, S.; Staffell, I.J.E. Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data. *Energy* **2016**, *114*, 1251–1265. [CrossRef]
28. Staffell, I.; Pfenninger, S.J.E. Using bias-corrected reanalysis to simulate current and future wind power output. *Energy* **2016**, *114*, 1224–1239. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.