INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

A Framework to Detect and Classify Time-based Concept Drift

M Thangam[1]*, A Bhuvaneswari[1], J Sangeetha[1]

1 Associate Professor, Cauvery College for Women (Autonomous), [Affiliated to Bharathidasan University], Trichy, Tamil Nadu, India

OPEN ACCESS

Received: 14-03-2023

Accepted: 24-11-2023

Published: 28-12-2023

Citation: Thangam M, Bhuvaneswari A, Sangeetha J (2023) A Framework to Detect and Classify Time-based Concept Drift. Indian Journal of Science and Technology 16(48): 4631-4637. https://doi.org/10.17485/IJST/v16i48.583

* **Corresponding author**.

thangamm.it@cauverycollege.ac.in

Funding: Seed Money for research projects from Cauvery College for Women (Autonomous), Tiruchirappalli – 620018, India

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To design a framework that performs time series decomposition to detect and classify the types of concept drift in a data stream. The aim of this research is to increase the classification accuracy in the detection and classification of drifts. **Methods:** The proposed method is validated using the Beijing PM2.5 dataset available in the UCI Machine Learning Repository. This dataset has 13 attributes and experiments were performed with the existing drift detection framework algorithms such as EFCDD, Meta-ADD, CIDD, and comparisons were performed with the proposed TBD framework. The outcome of this research is aggregated with Classification accuracy. An effective algorithm selection framework is presented that detects and classifies time-based concept drift existing in the data. The temporal aspects of the data are decomposed to determine the algorithm to be applied to detect and classify the types of drifts. Depending on the decomposed levels, three varied algorithms have been applied and used for the effective detection and classification of time-based drifts. **Findings:** The performance of the proposed method is validated using the classification accuracy and compared with the existing drift detection framework algorithms. The proposed framework achieves maximum classification accuracy of 95.24% than all the other existing methods. **Novelty:** A novel framework has been proposed with better classification accuracy for the detection and classification of time-based concept drift.

Keywords: Feature Selection; Concept Drift; Multiple Drift Detection; Timeseries decomposition; Classification Accuracy

1 Introduction

Over the past decade, there has been a growing focus on the utilization of real-time data. With advancements in technology, a wide range of applications now generate large quantities of data rapidly. Any series of data accompanied by a timestamp is referred to as a data stream. These data streams consist of a substantial volume of information with varying velocities [(1)].

Over time, the process of data generation undergoes changes, necessitating adaptation by the model to accommodate the incoming data. This phenomenon, referred to as concept drift, results in a decline in the performance of the predictive model [(2)]. Concept drift can manifest in different forms, including gradual, abrupt,

recurring, or incremental changes, which are contingent upon the speed and nature of the underlying changes. The frequency and severity of concept drift can vary depending on the particular problem domain and the dynamics of the data[3]. The significance of acquiring knowledge from data streams cannot be overstated for any machine learning model. The learning model needs to continuously adapt and learn in order to account for concept drift[4].

Research works have been conducted to detect concept drift by monitoring data characteristics change, which achieves high detection capabilities in a few types of concept drift, but not all. However, given the complexity of real-world streaming data applications, multiple drift types are likely to occur within a single data stream. There are several frameworks with only one drift detection without considering other types of drift which deteriorates the final performance.

Prasad et al.[5] proposed a framework to detect the presence of concept drift by using the "Ensemble Framework for Concept Drift Detection (EFCDD)" method. This method evaluates the distribution similarity of data by a scale of measurement known as data variants weight pattern. This framework considers and enhances classification accuracy. This method fails to detect recurring and incremental drifts.

Shan et al.[6] developed an Online Active Learning Ensemble framework (OALEnsemble) for classifying the data streams. This method is proposed for drifting data streams based on a hybrid labeling strategy and includes an ensemble classifier that consists of a long-term stable classifier and multiple dynamic classifiers. Active learning that holds an uncertainty strategy adjusts the decision threshold to detect the drift occurrence. This framework increases the accuracy, but does not minimize the complexity involved in data stream classification as the dimensionality reduction was not performed.

Khammassi et al.[7] proposed the ensembleEDIST2 algorithm to handle changes over time (i.e., recurring drifts) for block-based data, weighting data, and filtering data. Complex drift involving changes in the combination of different characteristics like, speed, severity, and influence zones were handled. The ensemble's execution and changes were also detected, therefore observing improvement in terms of accuracy rate and dispensing steady actions for different concept drifts.

Li et al.[8] introduced a method to learn imbalanced data streams with concept drift using an incremental ensemble algorithm called Dynamic Updated Ensemble (DUE). Both concept drift and class imbalance were addressed with the aid of a bagging-based framework, therefore resulting in various comparatively steady data chunks. Without accessing previous block samples, a performance-based pruning mechanism was applied to eliminate inadequately behaved classifiers, therefore limiting the memory usage even though with several changes at the intermediate level.

Zheng et al.[9] implemented an efficient semi-supervised approach for classification over streaming data with recurring drift and concept evolution called efficient semi-supervised classification with recurring drift (ESCR). The designed approach uses an ensemble model with clustering-based classifiers along with change detection modules to minimize the time complexity, but it failed to optimize the efficiency of handling the data stream with multiple drifts. This approach is compared to many well-known semi-supervised classification approaches for data streams.

Pratama et al.[10] proposed a deep-evolving fuzzy neural network (DEFNN) to improve classification accuracy and concept drift detection. A deep-layer neural network is used in addition to fuzzy techniques. It is used to process dynamically generated data and is built with deep-layered network architecture. The nonlinear mapping of high-level feature selection was not performed for minimizing the drift detection time.

Priya et al.[11] implemented an effective class imbalance with concept drift detection (CIDD) using Adadelta optimizer-based deep neural networks (ADODNN), named CIDD-ADODNN model for the classification of highly imbalanced streaming data. A drift detection technique called an adaptive sliding window (ADWIN) is employed to detect the existence of the concept drift. This work does not perform feature selection and clustering techniques to identify the type of drift that occurs in the dataset.

Yu et al.[12] introduced a novel framework Meta-ADD (Active Drift Detection) to classify the type of Concept drift. This method achieves accuracy in the detection and classification of drifts by applying pre-training and fine-tuning the model offline on data streams. Meta-ADD automatically classifies the drift types and does not require a hypothesis test.

Mayaki et al.[13] proposed an autoregressive-based drift detection that integrates an ML algorithm with ARIMA models for the detection of drift in a data stream. It considers the error rate of a machine learning model and achieves higher accuracy with a low false alarm rate. Along with drift detection, concept drift adaptation is also considered.

Hierarchical Linear Fourier Rates (HLFR) is a unique concept drift detector presented by Yu et al.[14] and is included in the Hierarchical Hypothesis Testing (HHT) architecture. By substituting an adaptive training technique for a well-known remedial scheme, the concept drift adaption capabilities of HLFR can be greatly enhanced. These techniques perform better in terms of concept drift adaptability, detection accuracy, and detection duration.

To detect and classify the types of drift, a Time- Based Drift (TBD) framework is proposed which combines various concept drift handling methods to ensure that multiple kinds of drifts can be detected within one single framework. A completely unique drift detection framework is introduced to detect every type of drift occurring within a real-world data stream.

The rest of the paper is organized as follows. Section 2 introduces the proposed framework with a detailed description of its various algorithms. Section 3 presents the evaluation and comparison of the proposed framework with other existing methods. The conclusion and avenues for further research in this area are discussed in Section 4.

2 Methodology

A Time-Based Drift (TBD) framework consisting of three algorithms is designed that combine different detection methods to identify the presence of drifts from the given data stream. Each model is specialized in detecting one or two types of concept drifts. The algorithms combined in this framework are Linear Stochastic Feature Embedding based Regressive MIL Boost Data Classification[15] (LSFE-RMILBC), Adaptive Statistical Stochastic Deep Gradient Learning[16] (AS-SDGL) and Exponential Kernel Feature Map Theil-Sen Regression-Based Deep Belief Neural Learning Classifier[17] (EKFMTR-DBNLC). Figure 1 shows the flow of the proposed framework.

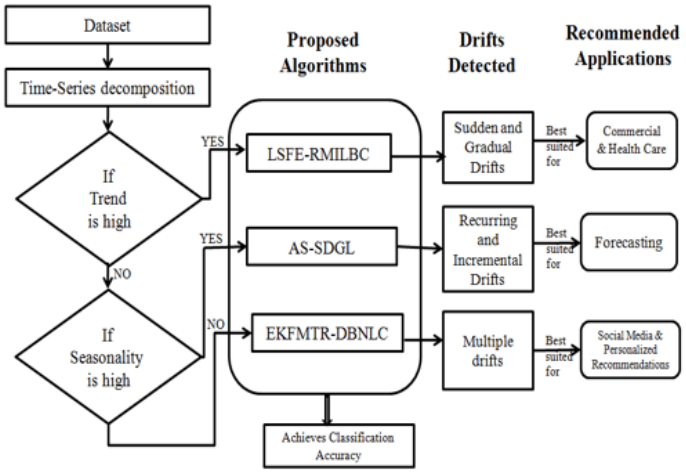

Fig 1. TBD Framework

This framework provides the users to choose a dataset and perform time-series decomposition to detect the concept drift based on their category. A collection of data points that have been indexed in time order is referred to as time series data, often known as time-stamped data[18]. These data points, which frequently include repeated measurements taken from the same source throughout time, are used to track changes over time. Data with temporal features can be broken down to reveal the main temporal trends that are present. A time series is a collection of well-defined observations of data instances that are obtained by means of repeated measurements over time. Breaking a time series into its components is time series decomposition. An observed time series data can be decomposed into three components: Trend, Seasonality, and Residual or unexpected variations.

Data patterns that show how a sequence changes over time to significantly higher or lower values are called trends. In other words, when the slope of a time series goes up or down, a trend can be noticed. A trend typically lasts for a short moment before dissipating and does not reoccur.

Seasonal variation, also known as seasonality, refers to cycles that occur on a regular basis over time. A time series' cycle structure may or may not be seasonal. It is seasonal if it consistently repeats at the same frequency; otherwise, it is not seasonal and is referred to as a cycle.

Residual or Unexpected variations are the random variations in the time series. It exhibits sudden changes in the data stream in which the patterns cannot be stated under any of the previously defined terms.

The algorithms listed in this framework use Ensemble methods, Statistical Methods and Deep Learning methods to detect the types of drifts and thereby improve accuracy in Classification. Each algorithm has its own characteristics to detect the types of drifts. There are two distinct advantages to this strategy. To begin, it can handle all contemporary kinds of temporal-based concept drift. Followed by, it provides the choice of concentrating on specific types of drifts based on the real-time data stream.

2.1 LSFE-RMILBC

This algorithm is used to detect sudden and gradual drifts by performing dimensionality reduction using the Modified Locally Linear Stochastic Embedding (MLLSE) technique to find the set of relevant features and uses the MIL Boost technique for classifying the drifts in the data stream. The Moving average regressive quadratic discriminant analysis is used to detect the drifts in the data stream by calculating the deviation of the input instances from the mean of the class. Based on moving average regressive analysis, the instances which diverge from the mean are recognized as concept drifts, in order to minimize the incorrect data stream classification.

The impact of sudden and gradual drifts plays a major difference in the commercial and retail sectors. Hence, this model can be applied to yield better accuracy results in real-time applications like Health care, Energy and Power, Factory Sensor data etc.

2.2 AS-SDGL

This algorithm deals with recurring and incremental drifts with two phases. The first phase receives the streams of data and uses Adaptive Seasonal and Trend Statistical Decomposition Drift Monitoring resulting in normalized preprocessed data.

This model caters to a second phase that takes the significant features with the aid of Change Drift Detection-based Stochastic Deep Gradient Learning and produces an output that can be used to detect concept drift.

This model can be applied in real-time applications like Forecasting models in which the impact of the presence of recurring and incremental drifts may decrease the results in the learning process.

2.3 EKFMTR-DBNLC

The proposed algorithm is used to detect the presence of multiple drifts from the data stream. This minimizes the dimension of the dataset by the Exponential Kernelized Semantic Mapping technique. After the feature selection, the change detection is performed by analyzing the feature value using Theil-Sen regression, and the total variation distance between two instances at different times is calculated to identify multiple drifts.

The neural learning classifier in the proposed model uses two visible layers such as an input layer, an output layer, and three hidden layers. This method constructs a projection matrix that is used to map a data feature set from high-dimensional data into reduced-dimensional data. The deep learning models are useful for the classification of concept drift in data streaming applications and these algorithms provide learning strategies that outperform conventional machine learning algorithms.

This method can be applied in real-time applications like Social Media Recommendations, Personalization Recommendations, and Financial Predictions.

The pseudo-code for the Time-Based Drift (TBD) framework by applying Time series decomposition is given below.

Input: Dataset D
Output: Detection and Classification of Drifts
Step 1: Begin
Step 2: Perform Time-Series Decomposition
Step 3: Evaluate Trend, Seasonality and Residual
Step 4: If Trend is high
    Select and apply LSFE-RMILBC to detect drift
    Classification of Sudden and Gradual Drifts
Else if Seasonality is high
    Select and apply AS-SDGL to detect drift
    Classification of Recurring and Incremental Drifts
Else
    Select and apply EKFMTR-DBNLC to detect drift
    Classification of Multiple drifts detected
Step 5: End if
Step 6: End if
Step 7: End

3 Results and Discussion

The experiment was conducted with Beijing PM2.5 Dataset, which was obtained from the UCI machine learning repository and implemented in Python. This dataset consists of meteorological data gathered from Beijing Capital International Airport.

Dataset characteristics are multivariate and time series and have 13 attributes and 43824 instances. When the datasets are divided into equal-sized data chunks for training, 20% of the data are maintained as the test set and the remaining data are utilized as the training data. It is expected that all labeled instances occur before unlabeled data in each chunk, where a defined portion of examples are labeled to simplify notations. Different proportions of labeled and unlabeled data are used to evaluate the proposed algorithms.

3.1 Result of Time-Series Decomposition

The dataset undergoes seasonal decomposition to determine the trend, seasonality and residual component. Figure 2 gives the time-series decomposition result of the PM2.5 dataset.

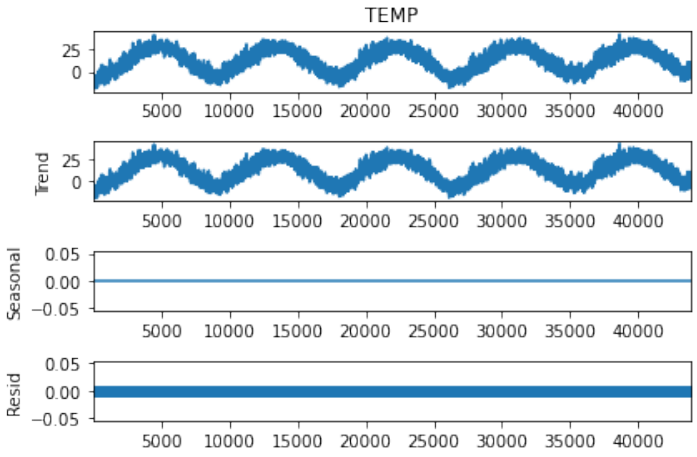

Fig 2. Results of the Time-series Decomposition - Features exhibiting Higher Trend Value

The time-series decomposition of the dataset is evaluated in Python. A decomposition is primarily an analysis tool used for time series analysis to predict forecasting models. It serves as an abstraction and offers an organized manner to consider modeling complexity, more particularly, the best way to include each of these components in a particular model. These components are to be addressed during data preparation, model selection, and model tuning. Problems in the real world are complex and noisy. There could be an upward trend followed by a downward trend. Along with the components of repeating seasonality, there could also be non-repeating cycles.

The dataset is preprocessed and the decomposition is performed. A feature is selected from the dataset and the component of the corresponding feature can be extracted. The figure shows that the TEMP and PRES features have higher trend values. So, this feature may exhibit sudden and gradual drift. The drift detection phase can be implemented by selecting LSFE_RMILBC. Thus, the time-series decomposition helps to extract the features that may be used for classification, regression, and forecasting tasks.

3.2 Performance Analysis of TBD Framework

The TBD framework is evaluated using classification accuracy as part of the performance analysis.

Classification accuracy (Acc) is measured as the proportion of all input data to incoming stream data in which drifts are correctly sorted into various classifications. The data stream classification accuracy is computed mathematically using the given formula,

$$CC = \left(\frac{n_{correctly\ classified}}{n}\right) * 100$$

Where CC denotes a classification accuracy, $n_{correctly\ classified}$ represents the number of data with drifts correctly classified, 'n' is the total number of streams of data taken as input. Therefore, CC is measured in the unit of percentage (%).

Additionally, the framework can be easily implemented based on the results of the time series decomposition of the dataset used. Experimental assessment of the proposed TBD framework and existing methods EFCDD [5], OAL Ensemble [6], CIDD [11], Meta-ADD [12] and is implemented in Python. Table 1 elucidates the comparative analysis of TBD framework with other existing models of the literature.

Table 1. Performance Analysis of TBD Framework

Framework	Classification Accuracy
EFCDD	84.14
OAL Ensemble	84.4
CIDD	90.45
Meta-ADD	93.56
Proposed TBD	95.24

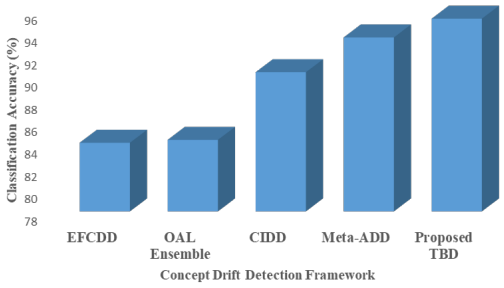

Fig 3. Comparative Analysis of TBD Framework

Figure 3 gives the graphical representation of the classification accuracy in comparison with the existing methods. The proposed TBD framework achieves higher classification accuracy in detecting the presence of drift and classifying accordingly. The time series decomposition of the dataset selects the appropriate algorithm for the detection and classification of drifts. The classification accuracy of TBD framework is increased by 11.1%, 10.84%, 4.79%, 1.56% respectively when compared with the existing methods.

4 Conclusion

An adaptive framework is needed for streaming data classification in order to recognize the presence of various concept drifts and make appropriate adaptations. Due to the complexity of real-world applications, a method for handling various types of drift is necessary. The proposed work introduced a drift detection framework capable of detecting drift and selecting the most suitable classifier for prediction. Three distinct algorithms developed were used to accommodate different levels of drift. To identify the occurrence of drift, a module for drift identification has been used, which decomposes the temporal components. Based on the temporal component, the appropriate algorithm for the training and prediction of the data is selected. All the proposed algorithms are evaluated with a real-time dataset namely Beijing PM2.5 taken from UCI Repository. The experimental results demonstrate that the proposed framework model outperforms existing models in the literature, exhibiting a high performance of classification accuracy with 95.24%. This model can be updated for handling efficient memory techniques in the future.

5 Acknowledgment

The authors are grateful to the Management and Principal for their encouragement and constant support. This research has been supported by the grant obtained under the scheme of Seed Money for research projects from Cauvery College for Women (Autonomous), Tiruchirappalli – 620018, India.

References

1) Agrahari S, Singh AK. Concept Drift Detection in Data Stream Mining : A literature review. *Journal of King Saud University - Computer and Information Sciences*. 2022;34(10, Part B):9523–9540. Available from: https://doi.org/10.1016/j.jksuci.2021.11.006.
2) Kabir MA, Begum S, Ahmed MU, Rehman AU. CODE: A Moving-Window-Based Framework for Detecting Concept Drift in Software Defect Prediction. *Symmetry*. 2022;14(12):1–20. Available from: https://doi.org/10.3390/sym14122508.
3) Manickaswamy T, Bhuvaneswari A. Concept drift in data stream classification using ensemble methods: types, methods and challenges. *INFOCOMP Journal of Computer Science*. 2020;19(2):163–174. Available from: https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/650.

4) Sanjith SL, Raj EG. Drift Detection Based Model Selection Framework For Real-Time Anomaly Detection In Iot. 2019. Available from: https://www.semanticscholar.org/paper/Drift-Detection-Based-Model-Selection-Framework-For-SanjithS-Raj/c7398d2f9f1d5232dce42e6a741e0a9422e02205.
5) Prasad KSN, Rao AS, Ramana AV. Ensemble framework for concept-drift detection in multidimensional streaming data. *International Journal of Computers and Applications*. 2022;44(12):1193–1200. Available from: https://doi.org/10.1080/1206212X.2020.1711617.
6) Shan J, Zhang H, Liu W, Liu Q. Online Active Learning Ensemble Framework for Drifted Data Streams. *IEEE Transactions on Neural Networks and Learning Systems*. 2019;30(2):486–498. Available from: https://ieeexplore.ieee.org/document/8401336.
7) Khamassi I, Sayed-Mouchaweh M, Hammami M, Ghédira K. A New Combination of Diversity Techniques in Ensemble Classifiers for Handling Complex Concept Drift. In: Learning from Data Streams in Evolving Environments;vol. 41 of Studies in Big Data. Springer, Cham. 2018;p. 39–61. Available from: https://doi.org/10.1007/978-3-319-89803-2_3.
8) Li Z, Huang W, Xiong Y, Ren S, Zhu T. Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm. *Knowledge-Based Systems*. 2020;195:105694. Available from: https://doi.org/10.1016/j.knosys.2020.105694.
9) Zheng X, Li P, Hu X, Yu K. Semi-supervised classification on data streams with recurring concept drift and concept evolution. *Knowledge-Based Systems*. 2021;215:106749. Available from: https://doi.org/10.1016/j.knosys.2021.106749.
10) Pratama M, Pedrycz W, Webb GI. An Incremental Construction of Deep Neuro Fuzzy System for Continual Learning of Non-stationary Data Streams. *IEEE Transactions on Fuzzy Systems*. 2020;28(7):1315–1328. Available from: https://doi.org/10.1109/TFUZZ.2019.2939993.
11) Priya S, Uthra RA. Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. *Complex & Intelligent Systems*. 2023;9(4):3499–3515. Available from: https://doi.org/10.1007/s40747-021-00456-0.
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. *Information Sciences*. 2022;608:996–1009. Available from: https://doi.org/10.1016/j.ins.2022.07.022.
13) Mayaki MZA, Riveill M. Autoregressive based Drift Detection Method. In: 2022 International Joint Conference on Neural Networks (IJCNN), 18-23 July 2022, Padua, Italy. IEEE. 2022;p. 1–8. Available from: https://doi.org/10.1109/IJCNN55064.2022.9892066.
14) Yu S, Abraham Z, Wang H, Shah M, Wei Y, Príncipe JC. Concept drift detection and adaptation with hierarchical hypothesis testing. *Journal of the Franklin Institute*. 2019;356(5):3187–3215. Available from: https://doi.org/10.1016/j.jfranklin.2019.01.043.
15) Thangam M, Bhuvaneswari A. Linear Stochastic Feature Embedding based Regressive MIL Boost Data Classification for Streaming Data. *Solid State Technology*. 2020;63(4). Available from: https://solidstatetechnology.us/index.php/JSST/article/view/8342.
16) Thangam M, Bhuvaneswari A. An Adaptive Statistical Stochastic Deep Gradient Learning for Handling Recurring and Incremental Drifts. In: Proceedings of the 2nd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications;vol. 237 of Lecture Notes in Networks and Systems. Springer Nature Singapore. 2022;p. 295–308. Available from: https://doi.org/10.1007/978-981-16-6407-6_27.
17) Thangam M, Bhuvaneswari A. Exponential kernelized feature map Theil-Sen regression-based deep belief neural learning classifier for drift detection with data stream. *International Journal of Advanced Technology and Engineering Exploration*. 2022;9(90):663–675. Available from: https://doi.org/10.19101/IJATEE.2021.874851.
18) Ouyang Z, Ravier P, Jabloun M. STL Decomposition of Time Series Can Benefit Forecasting Done by Statistical Methods but Not by Machine Learning Ones. In: Engineering Proceedings: The 7th International conference on Time Series and Forecasting;vol. 5 (1). MDPI. 2021;p. 1–10. Available from: https://doi.org/10.3390/engproc2021005042.