Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 9(90)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.874851

Exponential kernelized feature map Theil-Sen regression-based deep belief neural learning classifier for drift detection with data stream

Thangam M[1*] and A. Bhuvaneswari[2]
Assistant Professor, Cauvery College for Women (Autonomous), [Affiliated to Bharathidasan University], Trichy[1]
Associate Professor, Cauvery College for Women (Autonomous), [Affiliated to Bharathidasan University], Trichy[2]

Received: 23-September-2021; Revised: 15-May-2022; Accepted: 18-May-2022

Abstract

Data streams are potentially large and thus data stream classification tasks are not strictly stationary. In the process of data analysis, the fundamental structure may vary over time and the changes in the primary distribution of the data are known as drift. Early drift detection achieves better detection results in the evolving data stream analysis. In order to perform accurate drift detection with minimum time, a novel deep learning technique called exponential kernelized feature map Theil-Sen regression-based deep belief neural learning classifier (EKFMTR-DBNLC) was introduced. The main aim of the proposed EKFMTR-DBNLC technique is to perform multiple drift detection from the data stream using multiple layers. The proposed deep belief network comprises of various layers such as an input layer, an output layer and two hidden layers. The input layer receives the number of features and data from the dataset. The hidden layers perform the significant feature selection to reduce the drift detection time. The exponential kernelized semantic feature mapping technique is applied for identifying the significant feature for data classifications. Then, using Theil-Sen regression (TSR) function, the drifts in the data stream are detected and classified from the selected relevant features in the next hidden layer. The regression function analyzes the distribution of the data between the two-time intervals. Based on regression analysis, multiple drifts such as incremental drift, gradual drift, sudden drift and recurring drift are identified. Experimental estimation of the proposed EKFMTR-DBNLC technique and conventional methods are performed with different factors such as classification accuracy, precision, recall, F-score and drift detection time using real-world and synthetic datasets. The analyzed numerical result confirms that the proposed technique EKFMTR-DBNLC achieves 10% higher classification accuracy and also minimizes the time consumption by 13.5% than the conventional methods.

Keywords

Data stream classification, Drift detection, Feature mapping, Neural learning classifier, Regression function.

1.Introduction

Rapid expansion in technological advancement leads to exponential growth in the volume of data generated. Information shared by the people across the world through the Internet and also the data from hardware technologies such as sensors, mobile etc. leads to the growth of digital data. The data generated are dynamic in nature and is referred to as streaming data or data streams [1].

Data streams change over time, making it difficult to analyze large amounts of data in a distributed environment. The data streams are potentially large in size and thus it is not possible to process various machine learning techniques and approaches. These techniques failed to effectively process the data streams since they are highly prone to concept drift [2].

The challenge in the data stream evolves with the presence of concept drift as most of the real-world concept’s changes with time and is not constant [3]. The distribution of the data changes over time dynamically and is referred to as concept drift. Some examples are climate change, user preferences in reading web content, traffic control and management system, etc. This stream of data can’t be processed simultaneously due to the velocity of the generated data.

The concept drift challenge is becoming increasingly important in recent domains because the data is organized as a data stream rather than a static

[*] Author for correspondence

database. As a result, the concepts and data distributions remain unstable over a long term. As the data generated is dynamic, it is extremely difficult to fit data into a machine's main memory [4].

Predictive models can be trained in stages through uninterrupted updates or by using batch data to ensure model retention. Active and passive approaches to detect and handle concept drift have been proposed in the research literature, many of which have already been demonstrated in various application domains [5].

Identification of concept drift detection is easy with several existing methods, but they do not provide information about the amount of data precisely returning the types of drifts and the time involved to adapt to the new environment. The advent of deep learning (DL) models [6] is found to be useful for the classification of concept drift in data streaming applications. DL is a subset of machine learning that represents a collection of algorithms that simulates the function of different neuron levels in the brain. Over the past few years, the focus on DL has increased.

In this paper a novel drift detection technique named exponential kernelized feature map Theil-Sen regression-based deep belief neural learning classifier (EKFMTR-DBNLC) for handling multiple drifts and classifying the types of drifts have been proposed. The proposed method uses the exponential kernelized semantic feature mapping (EKSFM) technique for significant feature selection. In addition, the Theil-Sen regression (TSR) [7] method is used to analyze the distance between the features for identifying the types of drifts. A deep belief neural learning (DBNL) classifier is employed in selecting significant features and to classify the types of drifts with the assistance of various hidden layers [8].

Finally, comprehensive experiments are performed to estimate the performance of EKFMTR-DBNLC and other related works with the various performance metrics such as classification accuracy, recall, drift detection time, F-score, and precision using two synthetic datasets and two real-world datasets.

The remaining paper is organized in the sections as follows. Section 2 provides a brief explanation of the literature review. The methodology used in this work is described in Section 3. The experimental study and the related datasets on which the research is done have been discussed in Section 4. Section 5 elaborates on the discussion and the limitations. The conclusion along with the future work is included in Section 6.

2.Literature review

Researchers have proposed a variety of learning methods for concept drifts. DL is gaining popularity among researchers and multiple drift detection has to be focused on. DL algorithms provide learning strategies that outperform conventional machine learning algorithms.

Zheng et al. [9] proposed 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 stream.

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.

Yan [11] proposed a novel approach to detect concept drift in streaming data with the help of Hoeffding's dissimilarity. A Hoeffding's function measures dissimilarity levels in data streams minimizes errors in dissimilarity identification. This method achieves false alarm rate, detection delay and drift detection but failed to apply the adaptive learning algorithms to further improve the concept drift detection.

Prasad et al. [12] developed an approach to identify concept drift using ensemble strategies with multidimensional streaming data. This method achieved increase in drift detection rate, but drift detection time was not minimized. Multiple drift detection is not considered. The projection range of the field values representing the positions or Ids is used to detect recurrent drifts only.

Mahdi et al. [13] developed a novel concept drift detector, called KAPPA to identify the concept drift in a computationally efficient way by achieving accuracy, true positives and detection delay, but the designed approach concentrates on sudden drifts and failed to cover other types of drifts. Namitha and Kumar [14] have designed a new algorithm to find recurring concepts based on data stream clustering. A cluster function was employed to group similar data streams. Based on similarity function, the algorithm identified recurrent types of drifts. But the designed algorithm failed to deeply analyze the pattern for detecting the recurrence drift.

Liu et al. [15] proposed an equal intensity k-means space partitioning (EI-kMeans) technique to enhance drift identification and minimize error rate. In this technique with the assistance of k-means, an equality function was utilized for partitioning into equal spaces with which drift identification was made. However, drift detection performance results were not improved. Bi et al. [16] proposed classification over drifting and evolving stream (CODES) technique to improve accuracy and recall. The classification technique along with the evolving data streams was utilized to analyze the drift. But drift detection time was not minimized.

Chen et al. [17] introduced a fast condensed nearest neighbor (FCNN) algorithm to identify the gradual and sudden concept drifts from growing data streams. The algorithm condensed the evolving data streams using the nearest neighbor employing Euclidean distance. With Euclidean distance, the true positive and false positive rate were said to be improved. The designed algorithm increased the classification accuracy but the precision and recall were not improved. Mahdi et al. [18] developed a diversity measure-based drift detection technique to minimize the concept drift identification time and memory consumption. In this technique, detection of drift was made by employing a diversity factor based on the distance function and the overhead involved in drift detection was said to be improved. The designed technique only detects the gradual and recurrent drifts, but the other types of drifts were not detected.

Singh et al. [19] introduced an all in one stream process (AIOSP) for an efficient decision-making process of drift detection where data generation is enormous. But the system failed to find the significant features for drift detection while handling the huge amount of data. Ancy and Paulraj [20] developed a handling imbalanced data with concept drift (HIDC) mechanism to increase the precision and recall test. In this mechanism, occurrences of imbalanced data were focused on the concept drift employing mapping function but failed to develop the classification performance with minimum error.

Altendeitering and Dübler [21] proposed a support vector machine (SVM) for the classification process to identify concept drift. The classification between the concept and non-concept drift was made in a significant manner using efficient separation between the two classes. The designed SVM failed to improve accuracy and minimize execution time. Liu et al. [22] introduced a new masked distance learning (MDL) approach to detect concept drift and reduce the collective errors caused by iteratively evaluating massive data. The concept drift detection was made through masking and non-masking the relevant features. The analysis of relevant features through MDL is said to reduce concept drift detection time to a greater extent. The performance of drift detection accuracy was not achieved.

Yang et al. [23] developed an online sequential extreme learning machine (OS-ELMs) to improve concept drift detection with minimal cost. Concept drift was detected accurately and robustly with this sequential learning technique employed online. But the desired accuracy of concept drift with a lesser computational burden was not achieved. Jedrzejowicz and Jedrzejowicz [24] introduced the gene expression programming (GEP) classifier with drift detection. Gene classifier expression operators were utilized for efficient classification between drift and non-drift types in the presence of large data streams. But the classifier fails in minimizing drift detection time.

Mehmood et al. [25] introduced concept drift adaptation techniques. It ensures accuracy but failed to validate the methods with more recent estimation metrics for identifying multiple concept drifts. Yuan et al. [26] have designed an unsupervised algorithm to detect concept drift using multi-layered sliding windows. The dimensionality reduction technique was not applied to select significant features. Priya and Uthra [27] proposed imbalanced streaming of data using the DL framework. Concept drift was handled based on an adaptive sliding window, therefore, ensuring maximum accuracy. Oikarinen et al. [28] proposed virtual concept drift detection by employing supervised machine learning. With this type of learning technique, even in the presence of unknown ground-truth values, robust drift detection was ensured.

Mayaki and Riveill [29] proposed an autoregressive-based drift detection that combines a machine learning algorithm with autoregressive time series 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.

Nikpour and Asadi [30] introduced an incremental supervised clustering algorithm that clusters the data stream in a supervised manner. The data streams are analyzed by hierarchical and incremental based learning to identify the presence of concept drift. Clusters are eliminated when their values decrease over time and these clusters help in identifying class labels along with the detection of concept drifts.

A single approach may not work well under all circumstances. The specific issues with concept drift identification are listed below.

- A semi-supervised classification method detects concept drifts on time-series data but cannot be used with minimal time
- High-level feature selection using fuzzy-based methods detects concept drifts, but accuracy in classifying the types of drifts needs to be considered
- Identification of concept drifts is easy in classification-based and learner-based methods. Information about the types of drifts and drift detection time needs to be considered

Motivated by these issues and to handle four kinds of drifts, namely, sudden drift, recurring drift, gradual drift, and incremental drift, the EKFMTR-DBNLC method is proposed. An elaborate description of the EKFMTR-DBNLC method is provided in the following sections.

3.Methods

A novel technique EKFMTR-DBNLC is introduced for handling concept drift in the data stream classification. *Figure 1* presents the architecture of the proposed EKFMTR-DBNLC technique to perform classification of drifts in a data stream. Initially, data stream '$DS = d_1, d_2, \dots, d_n$' along with the features $A = a_1, a_2, \dots, a_n$ are collected from the dataset to perform actual drift detection process. With the objective of selecting the most pertinent features and to minimize the dimensionality involved, the proposed technique introduces Exponential kernelized semantic mapping technique. This technique uses semantic mapping which includes kernel operations to reduce the number of dimensions and retains the significant features for further detection processes. The changes are detected in the significant features through classification using TSR function. With the aid of the regression function, paramount classification is made that enhances drift detection. The processes are explained briefly with the help of the DBNL classifier.

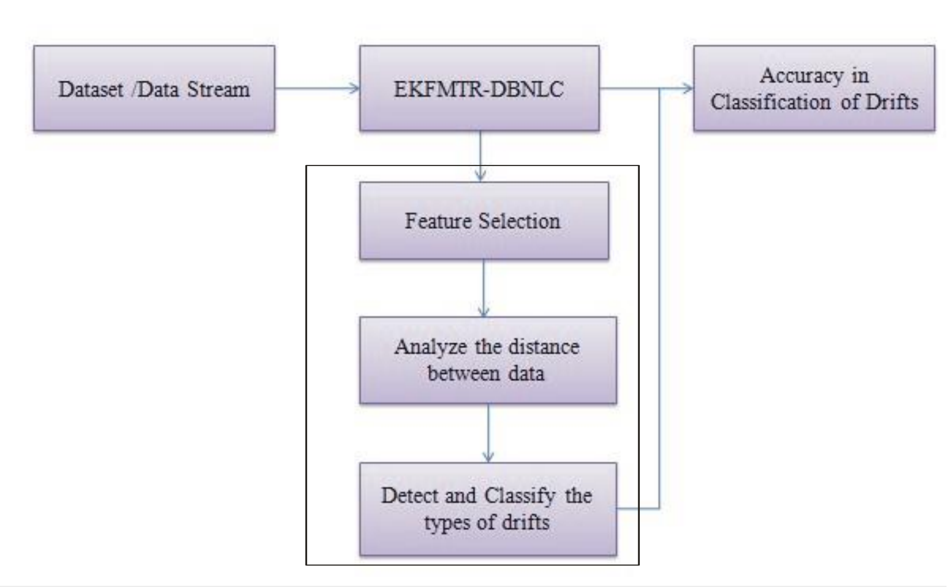
Dataset /Data Stream
EKFMTR-DBNLC
Accuracy in Classification of Drifts
Feature Selection
Analyze the distance between data
Detect and Classify the types of drifts

Figure 1 Architecture of the proposed EKFMTR-DBNLC technique

3.1Deep belief neural learning classifier

A DBNL classifier is a generative graphical network that includes different latent variables ("hidden units"). In the graphical network, connections are established between the layers but not between units within the layer. The proposed DBNL classifier comprises of an input layer, an output layer that is visible layers and another two layers that are hidden. The schematic structure is shown in *Figure 2*.

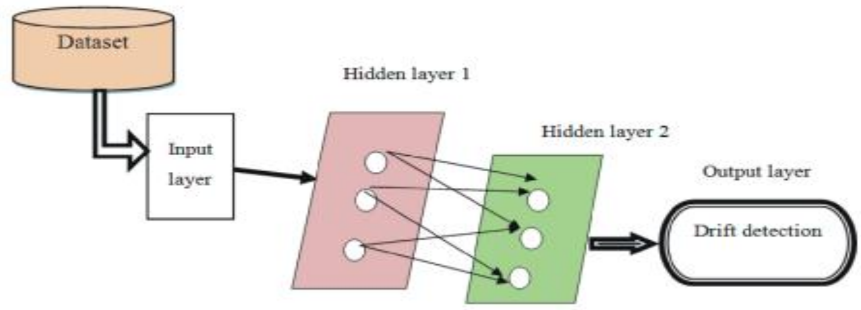

Figure 2 Schematic structure of DBNL classifier

Figure 2 demonstrates the schematic structure of the DBNL classifier. Three different layers are considered and they are the input, output and hidden layer. The input obtained from the dataset is provided in the input layer. The layers are connected from one to another in a feed-forward manner with regulating weights to form the whole network. Feature selection using exponential kernelized semantic mapping is performed at the hidden layer 1 and classification process using regression function is performed at the hidden layer 2 and finally, the classified drift type is provided as output at the particular layer.

The layer for the input receives features $A = a_1, a_2, \dots, a_n$ and the data $DS = d_1, d_2, \dots, d_n$. The action of the neuron at the input layer 'φ(t)' is given in Equation 1.

$$\varphi(t) = q + \sum_{i=1}^{m} x_i(t) r_0 \qquad (1)$$

From Equation 1,$x_i(t)$ denotes an output, r_0 indicates the regulating weights, q denotes a bias stored with a value '1'. The input is passed into the hidden layer one and significant features are selected.

3.2Exponential kernelized semantic feature mapping

In the proposed EKFMTR-DBNLC technique, the feature selection process helps to find the necessary features from the dataset provided at the input layer. The processing time and space requirement will be high if the selected features are more. Therefore, the proposed method executes significant feature selection using the EKSFM technique. It is a method that extracts a small set of features from the features of multidimensional vectors by preserving its data characteristics as such and is mainly used for dimensionality reduction.

A projection matrix is constructed using this method for the given data set. This matrix is used to map a data feature set from a high-dimensional space into a low-dimensional space. The dataset D is considered with a number of attribute or features $A = a_1, a_2, \dots, a_n$. Among them, significant feature are identified for minimizing complexity by applying the EKSFM technique.

The Exponential kernel between the features is estimated. The kernel is used to find the similarity between the features. The similarity between the features is estimated using Equation 2.

$$F\ (a_{i,} a_j) = exp\left[-0.5 \times \left(\frac{|a_i - a_j|}{\delta^2}\right)\right] \qquad (2)$$

Where, $F(a_i, a_j)$ denotes a kernel function, a_i, a_j represents the features in the input data, 'δ' denotes a deviation from the objective function, $|a_i - a_j|$ denotes a distance between the two attributes. The similarity value is provided by the kernel function between the range 0 and 1. Significant features are mapped from high dimensional space into the low dimensional set based on the similarity value.

3.3Theil-Sen regression based multiple drift detection

The selected significant features are passed to the next hidden layer for detecting the various drifts in the data streams. The TSR is a technique in machine learning applied in the second hidden layer of DL for analyzing the relationships between one or more independent variables (called 'features') and the dependent variables (i.e., outcomes). It is applied for analyzing the data stream with significant features extracted from the previous hidden layer. The proposed work identifies four drifts such as incremental drift, recurring drift, gradual drift, and sudden drift in data stream analysis. Here the outcome denotes multiple drift detection based on the data stream classification. In the proposed technique, TSR analysis is used as a generalized distance-based estimator for identifying the stream of data. Distance is used to analyze the deviation between the data.

The regression function considers the n-dimensional significant feature vector appearing at different times $'t'$. The concepts in the data are stationary if all the input stream of data samples $DS = d_1, d_2, \dots, d_n$ is generated with a similar distribution. The stream of data samples is distributed between the different time intervals t and $t + \Delta$. The probability of the data distribution is expressed in Equation 3.

$$p_t(DS, Y) \neq p_{t+\Delta}(DS, Y) \qquad (3)$$

Where, $p_t(DS, Y)$ denotes a probability of n-dimensional data sample distribution $DS = d_1, d_2, \dots, d_n$appearing at time t. $p_{t+\Delta}(DS, Y)$ indicates a probability of n-dimensional data sample distribution $DS = d_1, d_2, \dots, d_n$appearing at time $t + \Delta$, 'Y' represents the target regression outcomes. Equation 3 indicates that the probability of n-dimensional data sample distributions with successive time intervals is not the same.

Then the regression function analyzes the data between the successive time intervals t and $t + \Delta$ based on the distance measure and is calculated using Equation 4.

$$\boldsymbol{\nabla} = (d_t - d_{t+\Delta}) \qquad (4)$$

Where ∇ denotes a total variation distance $d_t - d_{t+\Delta}$. Based on the variation measure, the data stream is classified and the different drifts are identified using Equation 5.

$$Y = \begin{cases} if(\nabla \geq -1, \mathrm{ID}) \\ if(\nabla \geq +1, \mathrm{GD}) \\ if(\nabla = 0, \mathrm{RD}) \\ Otherwise, SD \end{cases} \qquad (5)$$

Where, Y denotes a regression outcome. If the value of the total variation is greater than -1, it denotes incremental drift, if the value of the total variation is greater than +1, it denotes gradual drift, if the value of the total variation is zero, it denotes recurring drift, or else a abrupt change causes sudden drift without variation.

The output of the hidden layer is expressed as given in Equation 6.

$$\beta(t) = \left(\sum_{i=1}^{m} x_i(t) * r_0\right) + (r_1 * \beta_{t-1}) \qquad (6)$$

where, $\beta(t)$ represents the hidden layer output at a time 't', $x_i(t)$ symbolizes input data, β_{t-1} denotes the last output of the hidden layer. The inputs provided '$x_i(t)$' are multiplied by the weight r_0. In Equation (6), the symbol $' * '$ denotes a convolution operator, r_1 indicates a weight between the hidden and input layer. The output layer yields the classification results. This method achieves the classification of drifts in data stream correctly and minimizes the false positive rate. The process of EKFMTR-DBNLC is introduced with the algorithm given below.

Algorithm 1 given above explains the procedure involved in the process of drift detection and data stream classification with higher accuracy and minimal time. To begin with, the streaming data along with its features involved is obtained from the dataset. Then the features are passed to the input layer using EKSFM. The kernel function is applied to find the important features. To achieve better classification results with minimal time, the insignificant features are removed. Then the next hidden layer processes the significant features selected. The TSR is applied in the same hidden layer for analyzing the data between the two consecutive time intervals. Based on regression analysis, the DL classifier categorizes the data and finds the various drifts. Finally, classification results are obtained at the output layer.

```
Input: Number of features $A = a_1, a_2, \dots, a_n$ and the data $DS = d_1, d_2, \dots, d_n$
Output: Increase the classification accuracy of Drifts
Begin
    Number of features $A = a_1, a_2, \dots, a_n$ and data $DS = d_1, d_2, \dots, d_n$
    For each feature $a_i$
        Measure Semantic Mapping using Exponential Kernel '$F\ (a_{i,} a_j)$'
        Find significant features
    Project the significant features
    Remove the irreverent features
    Select the significant features
    End for
    For each data '$d_i$' with selected features
        Apply Theil-Sen regression
    For each data at '$t$'
       For each data at '$t + \Delta$'
           Compute the distance '$\nabla$'
    if ($\nabla \geq -1$)
    Detect the incremental drift
    else if ($\nabla \geq +1$)
    Detect the gradual drift
    else if ($\nabla = 0$)
    Detect the recurring drift
    else
    Detect the sudden drift
    end if
    end for
    Obtain the classification results
    end for
    end for
End
```

4.Results

Standard datasets used in another similar research were used here. An experimental evaluation of the proposed EKFMTR-DBNLC and two other existing methods, ESCR [9] and DEFNN [10] were performed in Python using two synthetic datasets and two real datasets given in the *Table 1* below. The synthetic datasets were generated using a massive online analysis (MOA) framework.

Table 1 Real and synthetic datasets

Dataset	Name of the dataset	Size/ number of instances	Number of attributes
Synthetic	Streaming ensemble algorithm (SEA)	1,00,000	3
Synthetic	Hyperplane	1,00,000	10
Real	Electricity	45,312	8
Real	Beijing PM2.5	43,824	13

SEA dataset: The SEA is a synthetic dataset consisting of two classes and three features. In which only two features are relevant, and the third being noise. The values of all the attributes are between 0 and 10. The dataset points with various concepts are classified into four blocks. The classification is done using $f1+f2 \leq \theta$ in each block. $f1$, $f2$ indicates the first and second attribute and θ is a threshold value.

Hyperplane dataset: The synthetic dataset Hyperplane is specifically utilized in simulating the incremental drift. This hyperplane generator consists of 10 features and 100000 instances. The drift generated is said to be incremental that is achieved by varying the weight by 0.1 for each instance with an addition of 5% noise to the data.

Beijing PM2.5 dataset: The real dataset called Beijing PM2.5 Dataset is a weather dataset extracted from the UC Irvine (UCI) machine learning repository. This dataset consists of the PM2.5 meteorological data of the US Embassy in Beijing by concentrating four drifts, incremental, gradual, recurring, and sudden. The dataset consists of 43824 instances and 13 attributes. The attribute characteristics are integer and real type with an overall of 43824 instances. The attributes are row number, year, month, day, hour, PM2.5 concentration, dew point, temperature, pressure, combined wind direction, cumulated wind speed, cumulated hours of snow, and cumulated hours of rain. From the dataset, a significant feature PM2.5 concentration was considered over different hours for drift detection.

Electricity dataset: Electricity is an extensively utilized real dataset described from the Australian New South Wales Electricity Market. In this electricity dataset, prices are not fixed and are influenced only by two distinct factors, namely market demand and market supply. The values are here set every five minutes. The electricity dataset comprises of 45, 312 instances. Moreover, the class label identifies price change on the basis of the moving average of last 24 hours. Also, normalized versions of electricity real datasets include numerical values ranging between 0 and 1. The dataset includes 8 attributes and 2 classes. The attributes are date, day of the week, period, nswprice, nswdemand, vicprice, vicdemand and transfer. The two classes are referred to as UP (i.e., 1) and DOWN (i.e., 0) and the attribute period ranging between 0 and 0.50 is DOWN and between 0.50 and 1 is UP.

4.1Performance metrics

The performance analysis of EKFMTR-DBNLC and two other existing methods, ESCR [9] and DEFNN [10] are discussed with different metrics such as classification accuracy, precision, recall, drift detection time and F-Score for both synthetic and real datasets. The results obtained are represented in the form of a graph.

4.1.1Classification accuracy

The classification accuracy is measured as the accuracy involved during the process involved in drift classification. It is referred to as the ratio of incoming stream data with drifts classified into various classes of drifts correctly to the total streaming data considered for experimentation. It is mathematically expressed as given below.

$$C_{Acc} = \frac{n_{cc}}{n} \times 100 \quad (7)$$

From the above Equation (7), classification accuracy 'C_{Acc}' is calculated based on the number of correctly classified drifts 'n_{cc}' to the input provided 'n'. It is calculated in percentage (%).

Performance results for classification accuracy of the three methods are illustrated in *Figure 3*. From the observed results, the EKFMTR-DBNLC technique performs well than the other two existing methods. This is because of applying the DBNL classifier. The TSR function is applied to the DBNL classifier for analyzing the stream of the data and identifying the multiple drifts. Hence, classification accuracy gets improved.

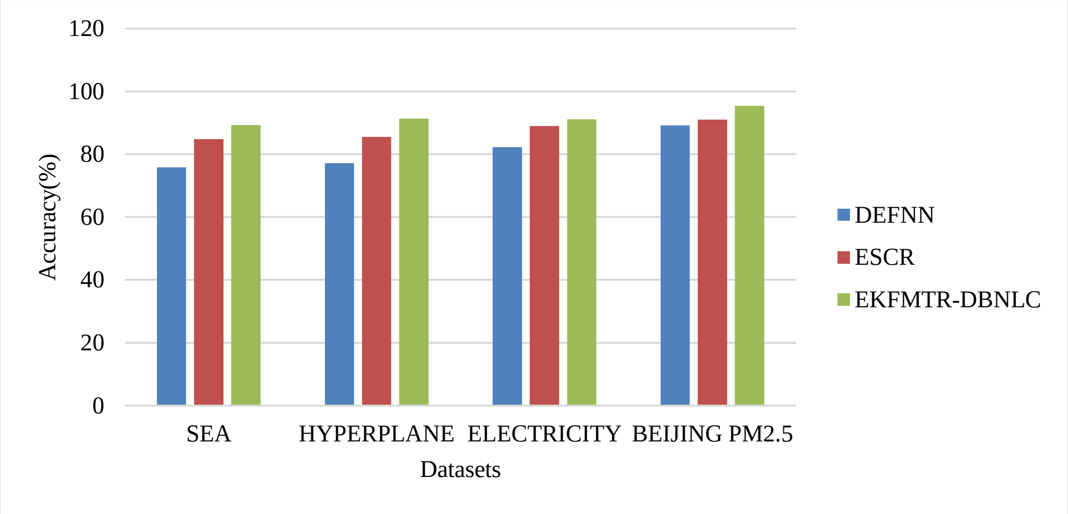

Figure 3 Performance results of classification accuracy

4.1.2Precision

Precision is defined as the ratio of relevant stream data in which the drifts identified are correctly classified to the total number of stream data. Therefore, the precision is calculated using Equation 8.

$$P = \left[\frac{T_p}{T_p+F_p}\right] \times 100 \quad (8)$$

Where P indicates precision, Tp denotes a true positive, Fp indicates a false positive. Precision is measured in terms of percentage (%). *Figure 4* demonstrates the graphical illustration of the precision using three various methods. From the graph, it is examined that our EKFMTR-DBNLC technique outperforms the other existing approaches. This significant improvement was achieved by applying the TSR function in the DBNL classifier.

The classifier accurately classifies the data stream and minimizes the false positive rate.

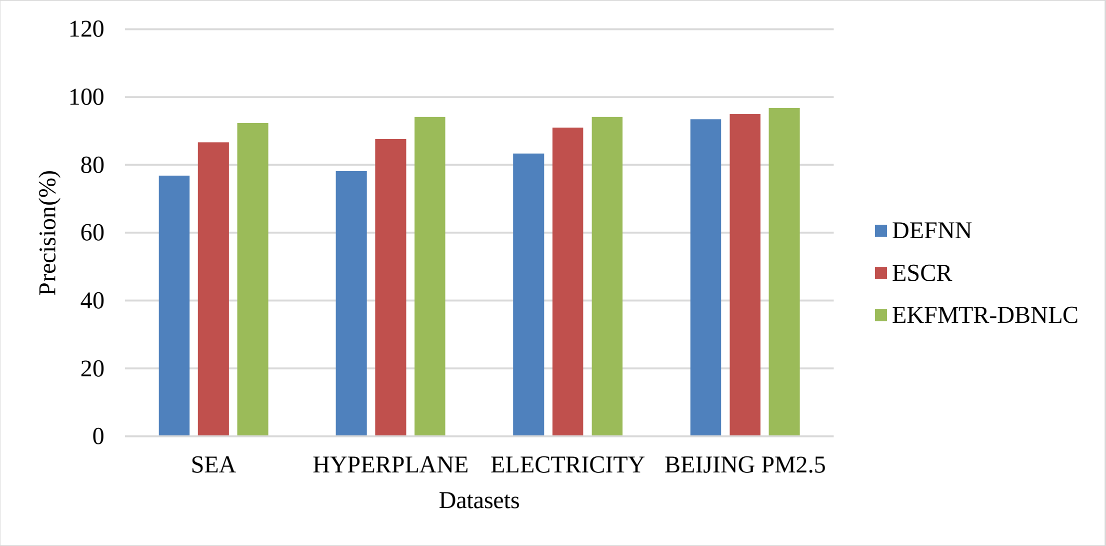

Figure 4 Performance results of precision

4.1.3Recall

Recall is measured as the ratio of true positive and the summation of true positive and false negative. Therefore, the recall rate is estimated using Equation 9.

$$R = \left[\frac{Tp}{Tp+Fn}\right] \times 100 \qquad (9)$$

Where 'R' indicates a recall, Tp denotes true positive, Fn indicates a false negative. The recall is measured in terms of percentage (%).

Figure 5 reveal the performance results of the recall. It proves that the EKFMTR-DBNLC technique offers improved performance than the existing methods. From the estimated results, it is concluded that the performance of recall is considerably increased by employing the EKFMTR-DBNLC technique. The overall assessment denotes that the recall of the proposed method is noticeably increased by 4% and 6% more than the existing methods. The reason for this improvement is due to the application of analyzing the features of various datasets using DL and hence finding the presence of multiple drifts.

4.1.4Drift detection time

The Drift Detection Time is formulated as the amount of time taken by the proposed algorithm to detect the various drifts through the data stream classification. The drift detection time is mathematically expressed in Equation 10.

$$DDT = [n] \times t[DD] \qquad (10)$$

Where, DDT denotes a Drift Detection Time, n denotes the number of data, $t[DD]$ denotes a time taken for detecting the drift. The overall drift detection time is calculated in milliseconds (ms).

Figure 6 depicts a comparison of drift detection time. The drift detection time is directly related to the size of the data. While increasing the number of data used for conducting the experiment, the drift detection time of existing methods gets increased. From these results, it is noticed that drift detection time is relatively lesser using the proposed EKFMTR-DBNLC technique. The reason behind the minimum time consumption is due to the application of the EKSFM technique. The feature mapping technique accurately finds significant features from the number of features set for data classifications. Then the regression function uses this significant feature and finds the types of drifts with minimum time.

4.1.5F-Score

F-score is an accuracy estimation of the drift detection test. It is calculated using precision and recall. Precision refers to the number of true positive results (i.e., appropriate detection of the drift types) divided by the number of all positive results (i.e., drift types), including those not identified correctly (i.e., inappropriate detection of drift types). Recall refers to the number of true positive results divided by the number of all instances (i.e., the number of sample data) that should have been identified as positive.

$$\mathrm{F}-score = 2 \times \frac{\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}} \qquad (11)$$

From the above Equation (11), the F-score value is measured based on the precision and the recall.

Moreover, the highest F-score value is 1.0, which indicates precision and recall is perfect, whereas the lowest F-score value is 0, if either the precision or the recall is zero. *Figure 7* depicts a comparison of F-Score. The F-score calculated using the EKFMTR-DBNLC method is comparatively higher than the state-of-the-art methods. This is due to the application of TSR function while classifying different types of drifts in the DBNL framework. The DBNL framework in turn accurately classifies different data streams accurately according to drift types and therefore improves the precision and recall.

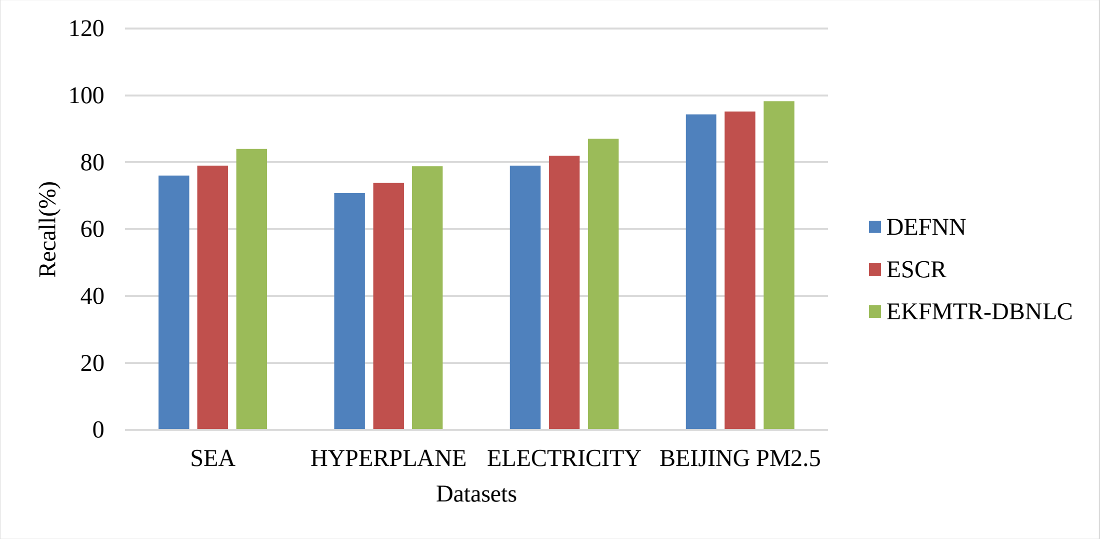

Figure 5 Performance results of recall

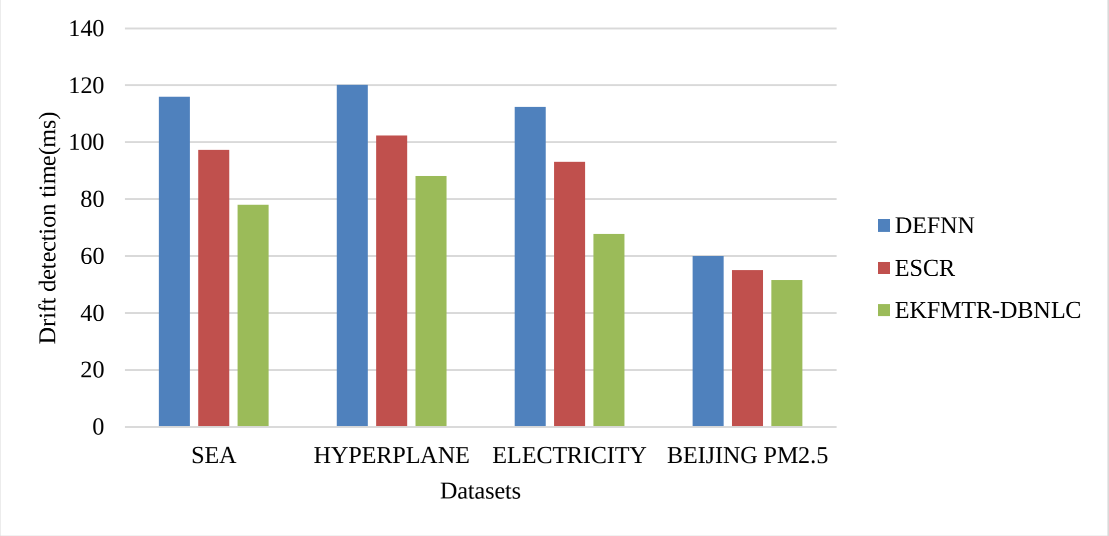

Figure 6 Performance results of drift detection time

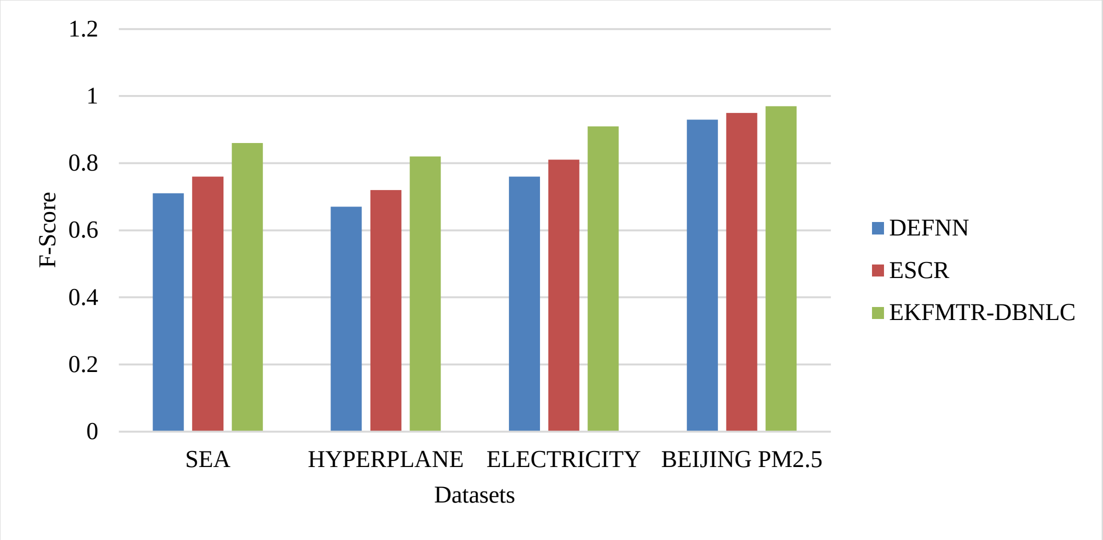

Figure 7 Performance results of F-score

5.Discussion

In this section, a detailed discussion including the key findings, interpretations, implications, limitations, and recommendations are discussed in detail.

- By designing separate feature selection and classification model via different hidden layers, the classification accuracy, precision, recall, drift detection and F-score involved in the detection of distinct types of drift found to be improved using the novel EKFMTR-DBNLC technique upon comparison with ESCR [9] and DEFNN[10].
- By employing EKSFM technique to identify significant features for further classification process, the concept drift detection time was found to be significantly reduced when compared to state-of-the-art methods.
- A paramount classification between different drift types based on the data is said to be assured by means of Theil-Sen regression function.
- The proposed method achieves increases in classification accuracy, precision and recall compared with the existing conventional methods using synthetic and real-world datasets.
- The drift detection time has been minimized using EKFMTR-DBNLC in both synthetic and real time datasets.
- The F-score of EKFMTR-DBNLC method achieves best result when compared with the existing methods using real time and synthetic datasets.

5.1Limitations

DL generates stable and steady static models from archival data. However, once stationed in production, these models degrade with time. There might be transposes and substitutes in data distribution in real-world applications, hence resulting in biased predictions or detections. Also, there may occur supplementary components in real-world interconnections that would have influenced the predictions or detections. Hence, keeping an eye on the alternates consistently in our model's practices is of paramount significance. Identifying such drifts and robotizing explicit actions for retraining the model provides robust and unbiased predictions or detections over time. Clustering can also be considered for better performance in the detection of drifts. A complete list of abbreviations is shown in *Appendix I.*

6.Conclusion and future work

Concept drift detection is a significant event in data stream analysis. A novel DL-based drift detection technique called EKFMTR-DBNLC is introduced. This technique analyzes the data stream and multiple drift detection through the feature mapping and classification in the different hidden layers. In the first hidden layer, an EKSFM technique is employed to find the important features. With the selected significant features, the data classification is performed using the Theil-Sen regression function in the second hidden layer. Based on regression analysis, various drift data classification is accurately performed by deeply analyzing the selected significant features. The in-depth experiments are accomplished with a number of data and compare the results of the proposed technique with two conventional algorithms. The observed numerical results have confirmed that the proposed EKFMTR-DBNLC technique has enhanced the performance of DL classification in terms of accuracy, precision, recall, F-Score and drift detection time.

Although our proposed EKFMTR-DBNLC technique charts outperform most metrics in the literature, there are also some potential improvements. One such extension is to examine the temporal characteristics as DL degrades with time and therefore resulting in biased predictions or detections. This could be done by either checking the trade-off between temporal aspects during drift detection and the regularity of recurring concepts or by accumulating a new norm to the learning element. Due to the significance of certain parameters in the temporal adjusting learning factor, and using the method proposed in the application data would be an interesting extension of this task.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Author's contribution statement

Thangam M: Conceptualization, methodology, analysis, investigation, validation, writing, original draft preparation. **Dr. A. Bhuvaneswari:** Validation, supervision and project administration.

References

[1] Khamassi I, Sayed-mouchaweh M, Hammami M, Ghédira K. Discussion and review on evolving data streams and concept drift adapting. Evolving Systems. 2018; 9(1):1-23.

[2] Krawczyk B, Minku LL, Gama J, Stefanowski J, Woźniak M. Ensemble learning for data stream analysis: a survey. Information Fusion. 2017; 37:132-56.

[3] Wares S, Isaacs J, Elyan E. Data stream mining: methods and challenges for handling concept drift. SN Applied Sciences. 2019; 1(11):1-19.

[4] Gama J, Medas P, Castillo G, Rodrigues P. Learning with drift detection. In Brazilian symposium on artificial intelligence 2004 (pp. 286-95). Springer, Berlin, Heidelberg.

[5] Wang X, Chen W, Xia J, Chen Z, Xu D, Wu X, et al. ConceptExplorer: visual analysis of concept drifts in multi-source time-series data. In conference on visual analytics science and technology 2020 (pp. 1-11). IEEE.

[6] Hatamikhah N, Barari M, Kangavari MR, Keyvanrad MA. Concept drift detection via improved deep belief network. In electrical engineering Iranian conference on 2018 (pp. 1703-7). IEEE.

[7] Shah SH, Rehman A, Rashid T, Karim J, Shah S. A comparative study of ordinary least squares regression and Theil-Sen regression through simulation in the presence of outliers. Journal of Science and Technology. 2016; 137-42.

[8] Hua Y, Guo J, Zhao H. Deep belief networks and deep learning. In proceedings of 2015 international conference on intelligent computing and internet of things 2015 (pp. 1-4). IEEE.

[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.

[10] Pratama M, Pedrycz W, Webb GI. An incremental construction of deep neuro fuzzy system for continual learning of nonstationary data streams. IEEE Transactions on Fuzzy Systems. 2019; 28(7):1315-28.

[11] Yan MM. Accurate detecting concept drift in evolving data streams. ICT Express. 2020; 6(4):332-8.

[12] Prasad KS, Rao AS, Ramana AV. Ensemble framework for concept-drift detection in multidimensional streaming data. International Journal of Computers and Applications. 2020:1-8.

[13] Mahdi OA, Pardede E, Ali N. KAPPA as drift detector in data stream mining. Procedia Computer Science. 2021; 184:314-21.

[14] Namitha K, Kumar GS. Learning in the presence of concept recurrence in data stream clustering. Journal of Big Data. 2020; 7(1):1-28.

[15] Liu A, Lu J, Zhang G. Concept drift detection via equal intensity k-means space partitioning. IEEE Transactions on Cybernetics. 2020; 51(6):3198-211.

[16] Bi X, Zhang C, Zhao X, Li D, Sun Y, Ma Y. CODES: efficient incremental semi-supervised classification over drifting and evolving social streams. IEEE Access. 2020; 8:14024-35.

[17] Chen D, Yang Q, Liu J, Zeng Z. Selective prototype-based learning on concept-drifting data streams. Information Sciences. 2020; 516:20-32.

[18] Mahdi OA, Pardede E, Ali N, Cao J. Diversity measure as a new drift detection method in data streaming. Knowledge-Based Systems. 2020.

[19] Singh VK, Verma S, Kumar M. Stream processing with concept drift for event identification in sensors enabled IoT environment. IEEE Sensors Journal. 2019; 19(24):12187-95.

[20] Ancy S, Paulraj D. Handling imbalanced data with concept drift by applying dynamic sampling and ensemble classification model. Computer Communications. 2020; 153:553-60.

[21] Altendeitering M, Dübler S. Scalable detection of concept drift: a learning technique based on support vector machines. Procedia Manufacturing. 2020; 51:400-7.

[22] Liu A, Lu J, Zhang G. Concept drift detection: dealing with missing values via fuzzy distance estimations. IEEE Transactions on Fuzzy Systems. 2020; 29(11):3219-33.

[23] Yang Z, Al-dahidi S, Baraldi P, Zio E, Montelatici L. A novel concept drift detection method for

incremental learning in nonstationary environments. IEEE Transactions on Neural Networks and Learning Systems. 2019; 31(1):309-20.

[24] Jedrzejowicz J, Jedrzejowicz P. GEP-based classifier with drift detection for mining imbalanced data streams. Procedia Computer Science. 2020; 176:41-9.

[25] Mehmood H, Kostakos P, Cortes M, Anagnostopoulos T, Pirttikangas S, Gilman E. Concept drift adaptation techniques in distributed environment for real-world data streams. Smart Cities. 2021; 4(1):349-71.

[26] Yuan Y, Wang Z, Wang W. Unsupervised concept drift detection based on multi-scale slide windows. Ad Hoc Networks. 2021.

[27] Priya S, Uthra RA. Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. Complex & Intelligent Systems. 2021:1-17.

[28] Oikarinen E, Tiittanen H, Henelius A, Puolamäki K. Detecting virtual concept drift of regressors without ground truth values. Data Mining and Knowledge Discovery. 2021; 35(3):726-47.

[29] Mayaki MZ, Riveill M. Autoregressive based drift detection method. arXiv preprint arXiv:2203.04769. 2022.

[30] Nikpour S, Asadi S. A dynamic hierarchical incremental learning-based supervised clustering for data stream with considering concept drift. Journal of Ambient Intelligence and Humanized Computing. 2022; 13(6):2983-300.

Ms. Thangam M pursued her Master's degree in Computer Applications from Bharathidasan University in the year 2003 and is working as an Assistant Professor at Cauvery College for Women (Autonomous), Trichy. She has 13 years of academic experience and her areas of interests include Machine Learning, Artificial Intelligence and Data Mining.
Email: thangamm.it@cauverycollege.ac.in

Dr. A. Bhuvaneswari is working as an Associate Professor in Computer Science at Cauvery College for Women (Autonomous), Trichy. She completed her Doctorate in Computer Science in the year 2015. She has around 18 years of Academic experience and 11 years of research experience. She has published several research papers in International Journals and Conferences. Her area of interest is Mobile Communication and IoT.
Email: bhuvaneswari.it@cauverycollege.ac.in

Appendix I

S. No.	Abbreviation	Description
1	AIOSP	All In One Stream Process
2	DBNL	Deep Belief Neural Learning
3	DD	Time Taken for Drift Detection
4	DEFNN	Deep Evolving Fuzzy Neural Network
5	DL	Deep Learning
6	EI-kMeans	Equal Intensity k-Means
7	EKSFM	Exponential Kernelized Semantic Feature Mapping
8	ESCR	Efficient Semi-supervised Classification with Recurring Concept Drift
9	FCNN	Fast Condensed Nearest Neighbor
10	GEP	Gene Expression Programming
11	HIDC	Handling Imbalanced Data with Concept Drift
12	MDL	Masked Distance Learning
13	MOA	Massive Online Analysis
14	OS-ELMS	Online Sequential Extreme Learning Machine
15	P	Precision
16	R	Recall
17	SEA	Streaming Ensemble Algorithm
18	SVM	Support Vector Machine
19	T_P	True Positive
20	TSR	Theil-Sen Regression