Journal of Engineering and Applied Sciences 13 (Special Issue 10): 8281-8291, 2018
ISSN: 1816-949X

ARNN for Enhancing Drift Detection of Data Stream Based on Modified Page Hinckley Model

[1]Nabeel Al-A'araji, [2]Eman Al-Shamery and [2]Alyaa Abdul-Hussein
[1]Ministry of Higher Education, Baghdad, Iraq
[2]College of Information Technology, University of Babylon, Hillah, Iraq

Abstract: Data that continuously arrive at time-varying and possibly unbound known as data streams. Connected, communicating devices that is called the Internet of Things (IoT) requires stream processing to deal with data coming from the sensors of connected devices which are the basic source of data stream now and in the future. These data streams are actually huge such as (e.g., sensor readings, call records, web page visits). This study aims to detect the drift and identify type of drift of stream. In addition to the pre-processing this consists of applying multi polynomial regression to solve the missing values problem, windowing and features extraction. An Adaptive Regression Neural Network (ARNN) for error prediction is suggested as new model to facilitate the distinction between real and virtual drift. A Modified Page Hinckley (MPH) Model is proposed to detect the drift and to determine the type of this drift. Common performance measures are used to evaluate detected drift by assuming a change point can be considered the "positive" class while no change can be considered the "negative" class. These models are formed the learning operation for concept drift. The system is applied on two real dataset (PAMAP2) for physical activity and (ELEC2) for electricity. Artificial datasets, also are used to evaluate the system performance. These dataset are SEA, SIN, moving hyper plan, circle and wave generation dataset. Accordingly, the results show that the performance of proposed system is better with time accuracy has reached to 96% when compared with previous studies and the MPH method can detect all drift types.

Key words: Data stream, concept drift, Adaptive Regression Neural Network (ARNN), Modified Page Hinckley (MPH), (ELEC2), (PAMAP2)

INTRODUCTION

The world has come to be information driven, business and government are being totally automat many features of their systems , personal, societies and nation institution have been dedicate in dole out sensors and infrastructure to collect senses data on a continuous basis. Alternatively organizations are now essential to monitor and research on information from the sensors to determinate speedy resolution, to expand manufacture processes, selections of better logistics and finally, to develop monitor and be able to work in real systems. The emergence of stream processing is made the world gets more connected and instrumented (Carney *et al*., 2002). Many examples are created in fields of data stream extending from traffic systems commercial markets industrial, healthcare, monitoring and security, large-scale structure to scientific and environmental (Andrade *et al*., 2014). Most of stream resources assume that the data is received in sequence of instance label pairs $f(a_t, b_t)$ where a_t is referred to time that b_t data arrives.

Literature review: Many papers are presented fo drift detection. By Ross *et al*. (2012) designs a new method exponentially weighted moving average to detect concept drift by using a chart which monitors the misclassification rate of classifiers. Whereas, Khamassi and Sayed-Mouchaweh (2014) error distance based approach for drift detection is used as standard early drift detection method, this method checks the error between two uninterrupted errors of classification depending two data window. It traces concept drift by maintaining two windows. One is global sliding window and another window uses to store the present example. Furthermore, by Gama *et al*. (2014) a survey listed various aspects of concept drift such as concept drift approaches, the adaptive learning processes, categorize existing for management, utmost representative, distinct and popular techniques and algorithms, also, evaluation methods of adaptive algorithms and survey describe usual applications. Ienco *et al*. (2014) proposed a change detection method, called change detection in categorical

Corresponding Author: Eman Al-Shamery, College of Information Technology, University of Babylon, Hillah, Iraq

data streams). It is well suited for categorical data streams. This method detects changes in a batch incremental scenario. It is based a summarization strategy used to compress the actual batch by extracting a descriptive summary. As well as segmentation algorithm highlights changes and issue warnings for a data. By Pinage and Santos (2015) detect changes method was proposed based on data distribution dissimilarity and update the decision model only after drift detection. A graph entropy-based method is suggested by Yao and Holder (2016) to discover concept drift in graph streams. This method uses any graph stream classier with change detector method to simplify classification on non-stationary graph streams. The researchers by Chen *et al*. (2016) applied forward a new drift prediction algorithm to predict the location of coming drift points based on historical drift. This technique is used a probabilistic system to learn drift. Moreover, the research by Zliobaite *et al*. (2016) categorizes the applications that handle the concept drift with major types of applications, a discussion was introduced emphasizing the most important application oriented aspects that address various issues such as delayed labelling, label availability, cost-benefit trade-off of the model update and other issues peculiar to a particular type of applications. The researchers by Thakre and Dongre (2016) describe concept drift and its types, it's requirement and details of changes. There are various techniques of drift detection are introduced through resampling with some windowing techniques. It is very suitable for sudden concept drift but it lack in detecting the slow change in data stream. The proposed method in this study covers all types of drift.

MATERIALS AND METHODS

In changing environments, a concept drift appears which represents change over data distribution. This change may be real or virtual depending on the way the data distribution change.

Real drift appears when changes in p(y|X) occur. This change may happen either with or without a change in p(X). Virtual drift appears if the arriving data change distribution and thus change this p(X) without touching p(y|X) (Gama *et al*., 2014).

The plot in Fig. 1 indicates that the class margins change only in the case of real drift and the earlier decision model comes to be obsolescent.

Drift patterns are categorized according to the speed of the change from one concept to another into: gradual, sudden incremental reoccurring or concepts. Moreover, some data containing a combination of multiple changes, Fig. 2 shows drift types (Zliobaite *et al*., 2016).

A drift may be occurring: Sudden (abrupt) when moving from one data mean to another (an example of this type is

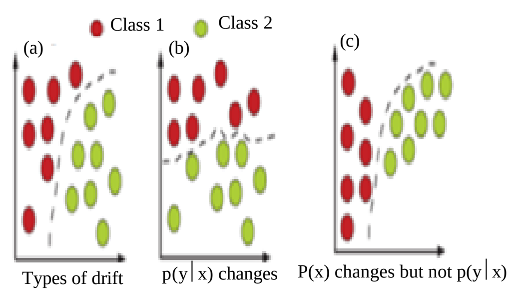

Fig. 1: Types of drift in data distribution: a) Original data; b) Real drift and c) Vitual drift

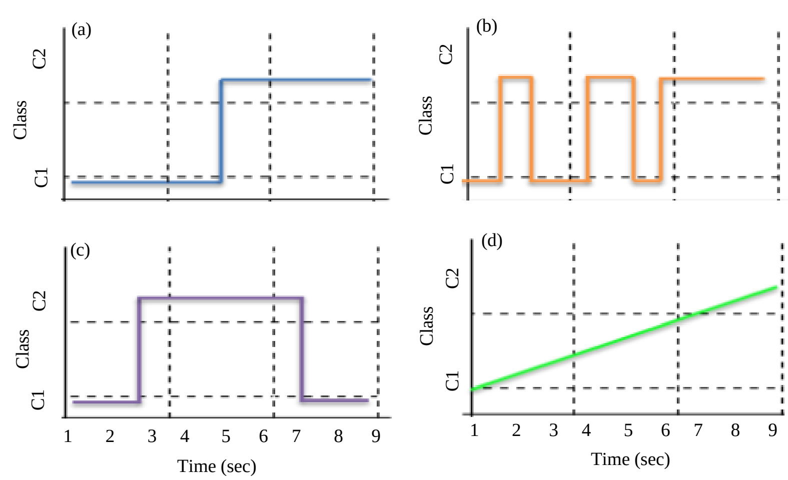

Fig. 2: a) Sudden; b) Gradual; c) Reoccurring and d) Incremental

switching from one topics of interest to another during credit analysis to get prices of public transference).

Gradually (an example related to news topics where dwelling changes to holiday homes but the individual does not switch abruptly but then goes back to the foregoing attention for some time).

Incrementally containing several in between concepts, the drift is noticed only when looking at a longer time period (for example, when a sensor come to be less accurate because of gradually it wears off). Reoccur in this type new concepts appear that are not seen previously (as happened in fashion).

Concept drift applications are categorized into: The monitoring and control applications are mostly interrelated with change detection when anomalous actions must be motioned (Gama *et al*., 2014).

The drift concept is main problem of data stream because the time represent critical factor of it in addition to the space. Methods for changing detection measure concept drift by recognizing change points for small time intervals through stream distribution. These detectors are based on the following methods:

- Sequential analysis method
- Control charts methods
- Differences between two distributions methods
- Heuristic methods (Gama *et al*., 2009)

Some examples of drift detector will be discussed briefly
Page Hinckley (PH): For sequential method which has been deduced from a Cumulative Sum (CUMSUM). It is considered to be sequential adaptation of the change detection in the mean value of a mean signal. This signal can be any classifier performance indicator or prediction error whose rise indicates a severe change in performance and thus should be alarmed (Zliobaite *et al*., 2016). The Adaptive sliding Window (ADWIN): this method re-computes the size of sliding windows which are of variable size, depending on the rate of change detected in this window which is dynamically is enlarged when there is no drift detected but shrinks when detecting drift occurs (Ienco *et al*., 2014).

Paired learners: This method maintains two learners: stable and reactive stable learner depending on all of its experience to predict, however, the reactive one based on a window from new one example to predict. It makes interaction between these two learners and their accuracy to handle drift (Bach and Maloof, 2008). The main difference among these popular methods is some of them

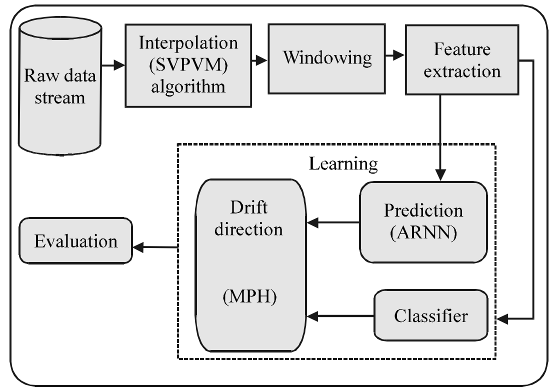

Fig. 3: General block diagram of the proposed system

depend on classification outputs (e.g., error rate) as indicators for detecting concept change and other depend on error from predictor.

Proposed system: The proposed system involves seven main stages (the pre-processing, windowing, feature extraction, error prediction, classifier, drift detection and evaluation) as shown in Fig. 3.

Preprocessing stage: The lost values are considered a common problem in much pivotal real applications. Huge stream of information contains missing values for many reasons for instance, a malfunction in a piece of equipment, a tissue section on a slide which has not been stained properly, a technician forgetting to enter a value in a spread sheet, etc. If they are neglected, they will be lost along with information related. The investigator eliminates missing points of data or fills it with reasonable estimate when analysing datasets with missing values. Removing missing values is applied simply but just sensible when size of missing is small and the application that uses the dataset not critical.

In this research, mathematical approach is introduced for solving the missing value problem based on polynomial curve fitting Segmentation of Variable Points and Variable Models (SVPVM). The main steps of the SVPVM Model are dynamic length segmentation, thresholdig, polynomial regression model and fitting missing points. The SVPVM Model steps are applied for each attribute (column) in the dataset and has been shown in algorithm 1. The missing values of each attribute are estimated depending on the relation among the known values for current attribute (column).

The first step in SVPVM is dynamic length segment, dataset with missing values are not processed as one block, they are divided into many overlapping samples for more precision as shown in algorithm 1. More details of this method has been explained in pervious study (Al-Al'araji *et al*., 2016).

Algorithm 1; SVPVM:
Input: - Matrix (D_{rc}) raw stream where (r) is (1,…, n) number of records and (c) is (1,…, m) number of attribute
-Threshold (θ)
Output:Matrix ($\hat{D}_{rc}$) of data without missing points

1- Let: $j \leftarrow 2$
While $j \leq m$ do (Repeats all steps for each attribute in the data)
Let:
$y_r \leftarrow D_{rj}$(Vector (y) of current attribute (i))
$x_r \leftarrow D_{r1}$(Vector (x) represents the arrival time of data)
$L \leftarrow 0$
$Na \leftarrow 0$ (number of missing points in current attribute)
$s \leftarrow 1$ (Number of NaN Segment)
-Compute:
For r =1 to n
If y_r = NaN then $Na \leftarrow Na+1$
End for (End loop r)
-Compute : $W \leftarrow n-Na$ (Number of known points)
2- For $i \leftarrow 1$ to n
If y_i = NaN then $L \leftarrow L+1$
Else
$B_{s1} \leftarrow x_i$(Time of known point)
$B_{s2} \leftarrow y_i$(Value of known point)
$B_{s3} \leftarrow x_{i-L}$ (NaN Segment start time)
$B_{s4} \leftarrow x_{i-1}$ (NaN Segment end time)
$B_{s5} \leftarrow i - L$ (NaN Segment start location)
$B_{s6} \leftarrow i-1$ (NaN Segment end location)
$B_{s7} \leftarrow ((i-1) - (i\_L))+1$
(NaN Segment Length)
$s \leftarrow s+1$ (Next NaN segment)
Generate matrix (B_{sk}) of all missing segments and known points where (s) is (1,…, W) represents number of known points 3 -From Bsk matrix select number of known points and k = 1 to 7 column of related information)
End else
End for (End loop i)
3-From B_{sk} matrix select number of known points according to threshold (θ) with shift one point and overlapping.
Let $O \leftarrow \theta - 1$ (polynomial order)
For $f \leftarrow 1$ to W-O
$VX \leftarrow Bt1$ (Vector of time of known points t = f, …, f+O)
$VY \leftarrow Bt2$ (Vector of values known points t = f, …, f+ O)
Using VX and VY to solving for thepolynomial coefficients by any standard method for coefficients calculation
For $g \leftarrow B_{f5}$ to B_{f6}
$\hat{D}_{gj} \leftarrow \alpha 0 \sum_{h=1}^{o} \alpha_h x_g^h$ (α_0, α_h are cofficients)
End for (End loop g)

End for (End loop f)

j = j+1

End while (End loop while)
4 -Return $\hat{D}$

Windowing stage: In time series, the data in the current time window is more important than the previous one. The sliding windows can be used appropriately. The window size is modified to detect changes at different locations. The size is chosen to be incremental as in Z, 2Z, 4Z, etc. where Z represents initial window size.

Feature extraction stage: In order to facilitate learning, it is possible to use generality steps and non-redundant information feature extraction methods. The features are calculated from a dataset among a set of statistical measurements. Therefore, twenty statistical measuring methods have been used: Mean Absolute value (MA), Zero Crossings (ZC), Slope Sign Changes (SSC), Modified Mean Absolute Value Type 1 (MMAV1), Modified Mean Absolute Value Type 2 (MMAV2), Simple Square Integral (SSI), Variance (VAR), Root Mean Square (RMS), Willison Amplitude (WA), Standard Deviation (SD), Median (MED), Teager Energy Operator (TE), Harmonic Mean (HM), Mean (M), Skewness (SW), Kurtosis (Ku) Integrated Signal (IS), Waveform Length (WL) and Average Power (P).

Depending on backward elimination feature selection method (15) of them appear to have positive effect features and other features indicated bad effect because the classifier accuracy has decreased. Feature extraction stage generates feature vector for each time points of calculated features. The result contains many rows represent number of feature vectors and columns represent number of selected features.

KNN classifier: KNN classifier is used for evaluation at two stages: feature extraction and drift detection. It is employed at the feature extraction stage to select the positive effect features. At the drift detection method, the classifier is used to get the class value and to detect drift inside that class. KNN satisfies good accuracy which may reach 96% a compared to other classifiers such as naive base.

Adaptive Regression Neural Network (ARNN): In order to reduce error and increase accuracy for the prediction method, there has been proposed the Adaptive Regression Neural Network (ARNN). In other words, it has been proposed to have many nodes for the input, the number of nodes has been determined by the number of windows of features for the selected dataset. Multiple nodes for the output represent the estimated output. Weights of the ARNN have been initialized by the coefficients of multiple regressions that have been calculated for each data attribute as in:

$$V = f^T \left(ff^T\right)^{-1} Y \quad (1)$$

In the start of the ARNN the actual Y is used. The feedback replaces Y by Y to use it in next window. The error between actual output and the estimated output is computed by:

$$E = |\hat{Y}-Y| \quad (2)$$

where, Y and Y represent estimated prediction and desired output, respectively, the feedback in ARNN is used to reduce the error in each epoch of training. The value of Y is computed by Eq. 3.

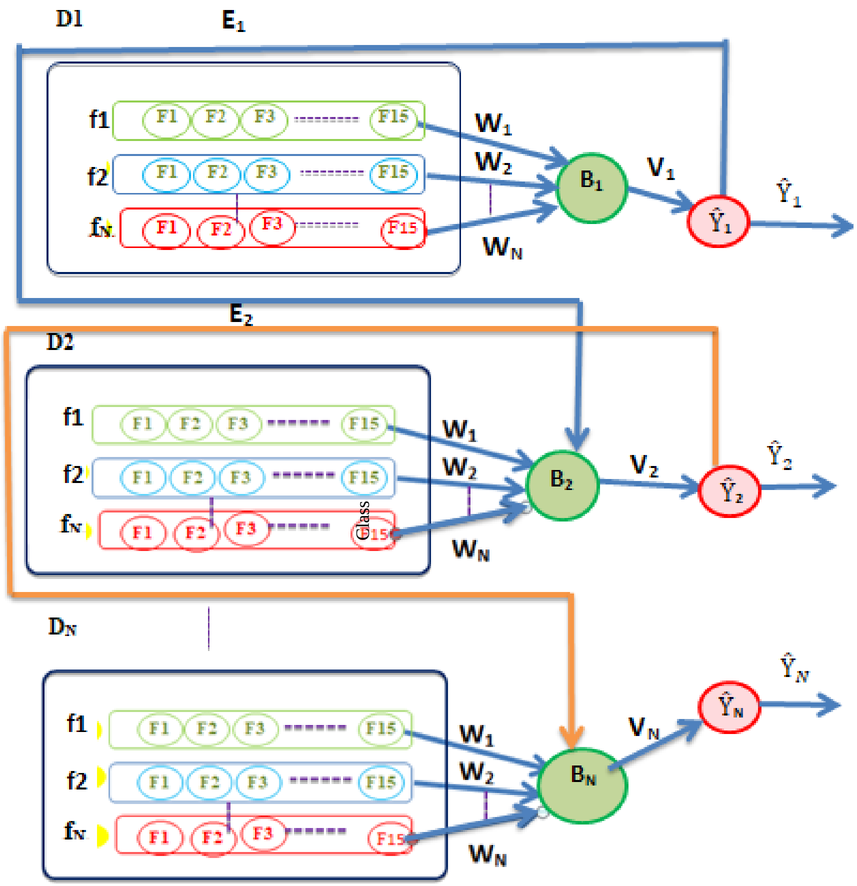

Fig. 4: The proposed ARNN topology

$$\hat{Y} = BV+E \tag{3}$$

In partial feedback of Error (E) in Eq. 3 has been used to specify the currently period time. The value of B = f W and initially W = 1 for each f.

A Contribution Update Factor (CUF) represents the adaptation process of weights update according to the time and contribution of error by addition of it to the feedback operation. The feedback of error and Y is used for adapting network. The ARNN has been trained on the selected training set of data. The updating process continues on the training data and the prediction values are adjusted result until a minimum error will have been reached. Figure 4 shows the general structure of ARNN while algorithm 2 explains all steps for ARRN.

Where, f_1 , f_2, …, f_N represent feature vectors that are founded in each window of (D_1, D_2, …, D_M) where N represents the number of feature vectors and M represents the window size. The weights of RNN are represented by W_1, W_2, …,W_N. For each feature vector, it contain (15) selected features that have positive effect on the classification accuracy and also contain output node used as feedback with weight represented by V_1, V_2, ...,V_N and feedback (E) where (E) is prediction error at that feature vector. The error feedback helps in error reduction for the next epoch in ARNN which make the drift detection operation more accurate. For further details, see previous research by Nabeel *et al*. (2017).

Algorithm 2; ARNN algorithm:

Input: a training pair (X_1 ,Y_1), (X_2 ,Y_2), …, (X_N,Y_N)of windows (D_1,D_2 ,…, D_M). F_I is (I×1) the feature vector, $Y_I$is vector of output and I = 1,2,3, …., N where N is length of window, M is number of windows

Output: E_I: reduced error vector

1- Initialize:

E_{max} is chosen

$W_I \leftarrow 1$ (W is vector of weight (I×1)

$E \leftarrow 0$ (Prediction Error)

2- Training step starts here

2.1. Input is presented and the layer1 output is computed

2.2. $B_i \leftarrow F_i W_i$ (For i = 1, 2, ..., I)

2.3 Prediction output is calculated in layer 2

2.4. $\hat{Y}_i \leftarrow B_i V_i + E_i$

3- Error vector is computed

$E_i \leftarrow |\hat{Y}_i - \hat{Y}_i|$ (For i = 1, 2, …, I)

4- weights are updated

4.1. $V_i \leftarrow X_i^T \left(XX_i^T\right)^{-1} \hat{Y}_i$ where (T) is vector transpose and (-1) is inverse of vector
4.2. $Y_i \leftarrow \hat{Y}_i$
5- if $I<N$ then $I \leftarrow I$, go to step2 otherwise go to step6
6- The training phase is finished
6.1If E_i<Emax then terminate the training epoch
and output weight B and error E
6.2. If $E_i \geq$Emax then $M \leftarrow M+1$, $I \leftarrow I$
And initiate the new training phase by going to step2
End

Drift detection by Modified Page Hinckley (MPH): The proposed MPH an extended version of the PH in its general form can detect sudden drift only whereas the MPH is adapted to detect many drift types (sudden, gradual and incremental) and produce warning alarm. Algorithm 3 displays the MPH. Three modifications are proposed in MPH:

It is commonly known that, the threshold (λ value in the standard PH method is taken as fixed for all data and data windows. This patently affects drift detection. If this value is not appropriate chosen, it would not lead to error detection or one might get a number of error detections. The threshold (β) for detecting sudden drift is proposed in MPH to change its value from window to another. The threshold is calculated as the median of the current window.

Algorithm 3; Modified Page Hinckley (MPH):
Input: E_i: Predicted error where (i = 1, ..., T) is the number of error values in the window of current class.Ω: Magnitude of change that is allowed. α: increasingstep factor. k: number of classes

Output: DS: Sudden drift, DG: Gradual/Incremental change
DW warning for change.
1- For $L \leftarrow 1$ to k
2- $x_i = E_{iL}$ (i = 1, ..., T)
Calculate:

$$\bar{x} = \frac{1}{T}\sum_{i=1}^{T} x_i$$

$M_i = \left(x_i - \bar{x} - \Omega\right)$ (Change in current window, i = 1, ..., 0T)
$s_i = \sqrt{x_i\left(1-x_i\right)/i\left(i=1,...,T\right)}$ (i=1... T)
(for previous window)
$H_j = \left(x_j - \bar{X} - \Omega\right)$ (Change in previous window, j = 1, ..., T-1)
$G_i = \alpha H_j + M_i$ (MPH test, j = 1, ..., T-1, i = 1, ..., T)
$mG = \min(G_i)$ (i = 1, ..., T)
$ms = \min(s_i)$ (i = 1, ..., T)
$mx = \min(x_i)$ (i = 1, ..., T)
SG = $SG = \sum_{i=1}^{T} G_i$ (i = 1, ..., T)
α is calculated according to equation as MPH thresold
3- Check for drift type:
If (SG-mG)>β
$DS \leftarrow 1$ (Set sudden drift alarm)
Else If $(x_i+s_i) \geq (mx+3ms)$ (I = 1, ..., T)
$DG \leftarrow 1$ (Set gradual drift)
Else If $(x_i+s_i) \geq (mx+2ms)$ (i = 1, ..., T)
$DW \leftarrow 1$(Set drift warning)
End if
4- Next Class , Go to step 1
End

Table 1: Datasets description

Dataset	Data type	No. inst	No. attr	No. class	Missing points	Drift type
PAMAP2	R	3850505	52	18	yes	-
ELEC2	R	27552	4	7	-	-
SEA						
SEA1	A	5000	3	2	-	S
SEA2	A	20000	3	2	-	S
SEA3	A	20000	3	2	-	S
SEA4	A	20000	3	2	-	S
SIN						
SIN1	A	1000	1	2	-	S
SIN2	A	1000	1	2	-	S
SIN3	A	1000	1	2	-	S
SIN4	A	2000	1	2	-	S
Wave						
WAV1	A	154755	52	18	yes	-
WAV2	A	800	1	-	yes	-
WAV3	A	13	1	-	yes	-
Hyper plane	A	360	2	2	-	G/I
Circle	A	500	4	2	-	G

Proposed increasing step factor (α) balance the importance of older samples in the drift indicator. The test variable is computed for the past error values (T-1).

Proposed conditions for checking if gradual/ incremental drift occurs. For each point (i) in the window assuming that amount of error is the probability of wrong prediction (x_i) with standard deviation (s_i). The error (pi) will decrease in the learning algorithm whereas the number of points increases when the distribution is static. Change occurs when significant increase in the error. The accuracy for time of the proposed system is calculated by the proposed Eq. 4:

$$\text{Time Accuracy}\left(\text{TA}\right) = \frac{\text{Real drift location}}{\text{Detected drift location}} \times 100\% \tag{4}$$

Dataset: The proposed system is implemented and tested on Real (R) and Artificial datasets (A) to determine the behaviour of the proposed models and specify drift types (S: Sudden, G: Gradual, I: Incremental). Some descriptions about these dataset are listed in Table 1 (Kadwe and Suryawanshi, 2015; Minku *et al*., 2010).

RESULTS AND DISCUSSION

The results for the proposed methods are evaluated by using artificial dataset and also many evaluation measurers have been used such as (MSE, MAE, precision, recall, F1-measure, G-measure, TA and others). Also, comparison with other traditional methods is performed. For pre-processing results of three artificial dataset are compared with Mean (MF) and Linear (LF) interpolation methods to fine missing points as shown in Table 2.

Table 2: Interpolation results

Data/Methods	Evaluation measures MAX	MSE	MAE
WAV1			
SVPVM	0.0063	0.2185×10^{-5}	0.64229×10^{-3}
LF	0.0135	0.1756×10^{-4}	0.0024
MF	0.0109	0.51576×10^{-4}	0.005
WAV2			
SVPVM	1.3741	0.0197	0.0415
LF	6.2144	0.7883	0.1659
MF	9.4321	18.2377	2.3721
WAV3			
SVPVM	4.9000	2.3897	0.6813
LF	19.1250	28.5736	1.7603
MF	21.8500	93.5325	4.7069

Table 3: Error in each window of SEA1 dataset

Window number	Error measures MSE	MAE	MAX
1	0.626844	0.626844	1
2	0.183581	0.354168	1.02573
3	0.000149	0.009568	0.048859
4	2.92×10^{-7}	0.000453	0.001564
5	7.47×10^{-10}	2.20×10^{-5}	9.79×10^{-5}
6	2.81×10^{-13}	3.87×10^{-7}	1.69×10^{-6}
7	6.01×10^{-17}	5.77×10^{-9}	2.47×10^{-8}
8	5.86×10^{-20}	1.78×10^{-10}	8.72×10^{-10}
9	1.10×10^{-22}	7.41×10^{-12}	3.19×10^{-11}
10	9.49×10^{-26}	2.11×10^{-13}	1.40×10^{-12}

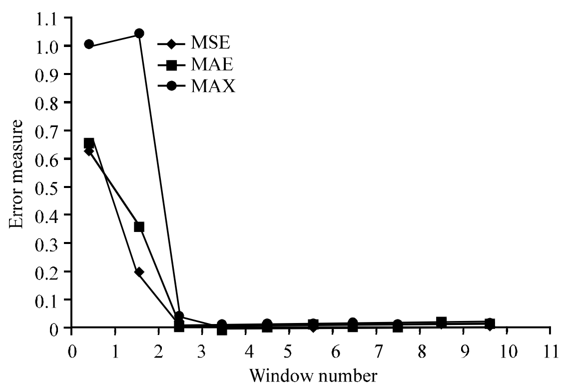

Fig. 5: Prediction error for windows in SEA 1

ARNN decrease error in SEA1 dataset from one window to the next window, one can note that error oscillation between decreasing and at some point of time increasing, this may give indication that there is drift in that point of time. In Table 3, the Maximum error value (MAX), Mean Square Error (MSE) and Mean Absolute Error (MAE) are listed for ten windows in SEA1 dataset. And curve of error is shown in Fig. 5.

The class used in drift detection stage is to detect change between different classes and inside class itself. The feature vectors in the proposed system are classified by KNN classifier to specify class type in the dataset. The accuracy of KNN classifier is higher than other classifiers

Table 4: Classifiers accuracy

Classifies	KNN	NB	LDA	QDA
Accuracy	97.3	30.57	16.38	15.68

Table 5: Delay time in drift detection

Delay in MPH	Delay in paper	TA in MPH	TA in paper	Actual drift
1556	1609	96.4%	93.2%	1500

Table 6: Performance measures in SEA3 and SEA4

Data set	G-mean	Precision	Sensitivity	F1-measure	Specificity	Accuracy (%)
SEA3	0.99	1	0.99	0.99	1	99
SEA4	0.99	1	0.99	0.99	1	99

when it is compared with them. One file from PAMAP2 dataset is used to compare the accuracy of KNN, Naive Bayse, Linear discriminant analysis LDA and Quadratic Discriminant Analysis QD Aclassifiers. Table 4 shows the accuracy result.

The drift is detected for each class for two files of the dataset as in Fig. 6 and 7. The class represents one of used activities. In standing activity two types of drift are founded sudden and gradual in lying activity sudden drift sudden is detected.

The changing from one activity to another produce sudden drift, since, it change the class type. In lying activity more than one sudden drift detected in this file of dataset. The drift detected in ELEC2 is gradual drift as shown in Fig. 8.

The SEA1 dataset contains sudden drift at point 1500, the implementation of drift detection method is shown in Fig. 9, also, the warning for drift is displayed as dotted line for the drift area, no gradual drift is founded and the system does not give any result in gradual drift indicator. The drift is evaluated in SEA1 data set and compared with study by Gama *et al.* (2009) where the best result is obtained in this study at delay time (1609), the result of proposed MPH gives delay time at (1556), the actual location of drift at point is 1500, so, the proposed MPH is with less delay time. Time accuracy is calculated according to Eq. 4 and is shown in Table 5.

Confusion matrix is generated as in Table 6 by assuming a change point which can be considered the "positive" class while no change can be considered the "negative" class. Some of useful performance metrics are employed to evaluate the MPH method for SEA3 and SEA4 as listed in Table 6.

Many researchers have generated their own dataset to test the performance of their proposed system which will be kept for themselves and not published in public, so, SIN dataset has been generated from known equations with a specific number of points to simulate stream data. A drift has been added to some points in order to test the MPH method. The drift is tested in two directions depending on drift types (real drift and virtual drift). The

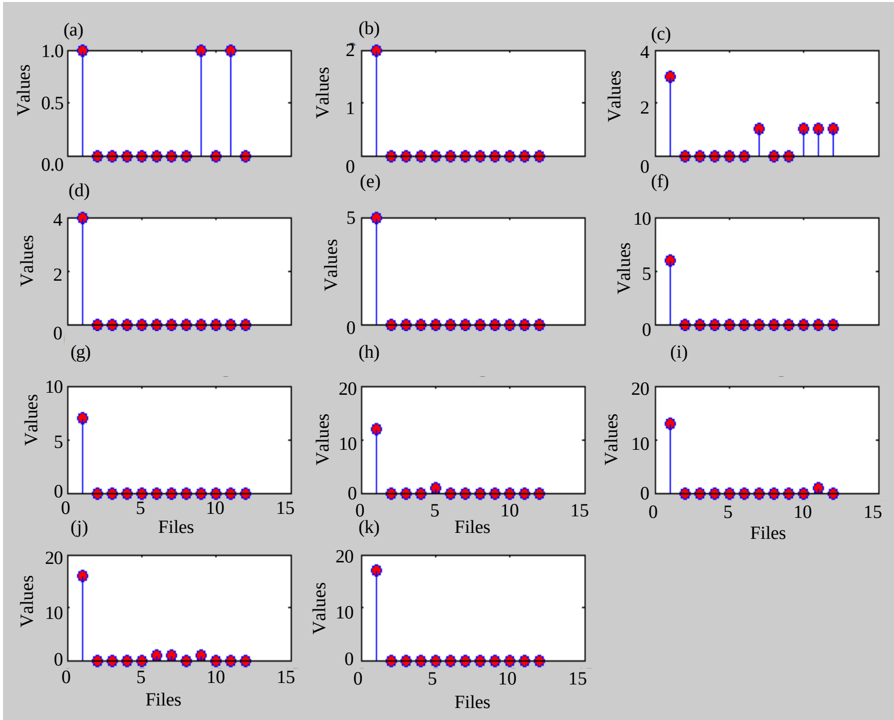

Fig. 6: Gradual and sudden drift in file1 of PAMAP2: a) Lying; b) Sitting; c) Standing; d) Walking; e) Running; f) Cycling; g) Nordic walking; h) Asceding stairs; i) Decending stairs; j) Vacuum cleaning and k) Ironing

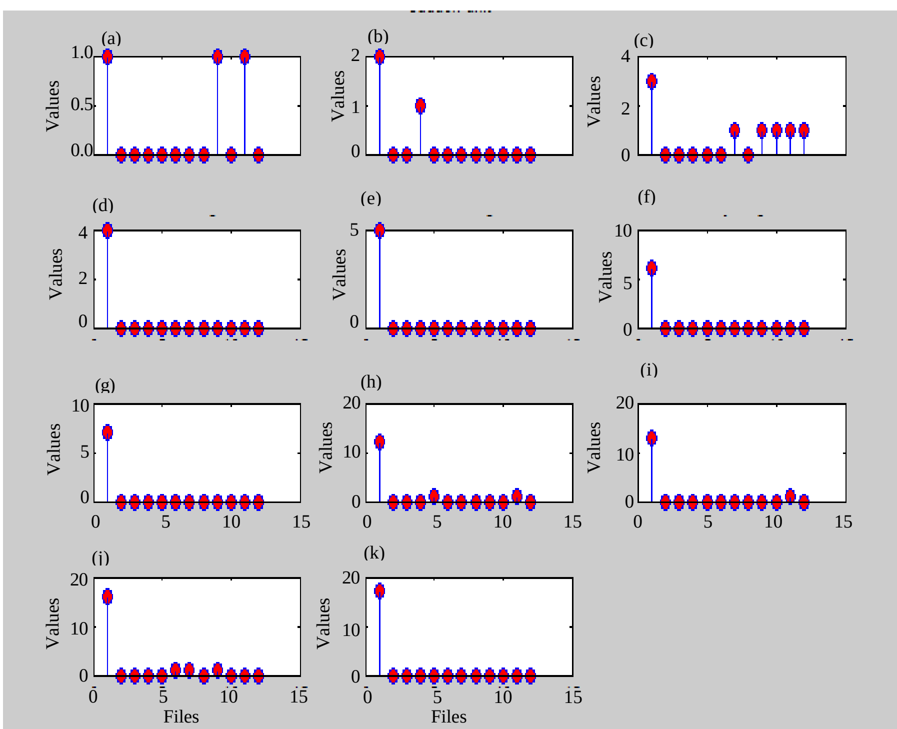

Fig. 7: Gradual and sudden drift in another PAMAP2: a) Lying; b) Sitting; c) Standing; d) Walking; e) Running; f) Cycling; g) Nordic walking; h) Asceding stairs; i) Decending stairs; j) Vacuum cleaning and k) ironing

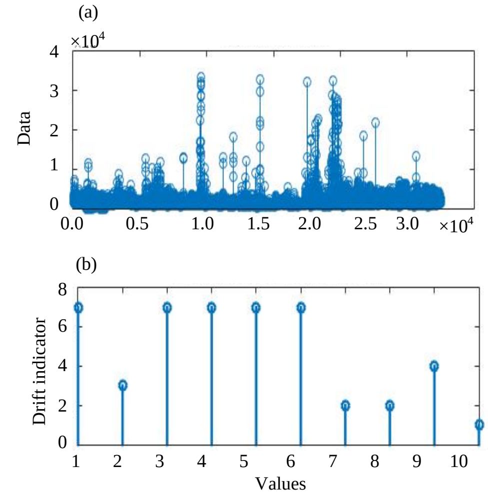

Fig. 8: Dected drift in ELEC2: a) ELEC2 dataset and b) Drift detected in ELEC2 dataset

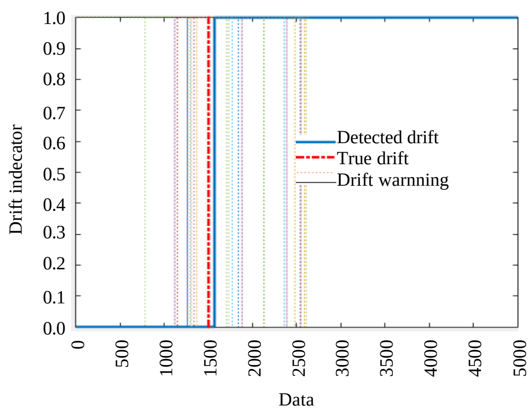

Fig. 9: Sudden drift in SEA 1 (Detected drift in SEA 1 dataset)

Table 7: Drift locations in SIN

Dataset	Real drift location	TP	TN	FP	FN
SIN1	200-500	290	1	0	11
SIN2	400-700	289	1	0	12
SIN3	600-900	287	1	0	14
SIN4	500-800	286	1	0	15

real drift is generated by changing the point values while virtual drift is generated by changing the distribution of data. By means of MPH methods one can distinguish between these drift types with high accuracy. Table 7 shows the drift locations and Table 8 shows performance measures for detected drift.

The delay time and time accuracy for different drift in SIN data is listed in Table 9. The parameter (з: Increasing step factor) is used in drift detection algorithm to reduce the effect of past values (pervious stream points) in the decision of current drift. The effect of past stream has changed depending on з. If its value increases, the effect, also, increases which mean that current points

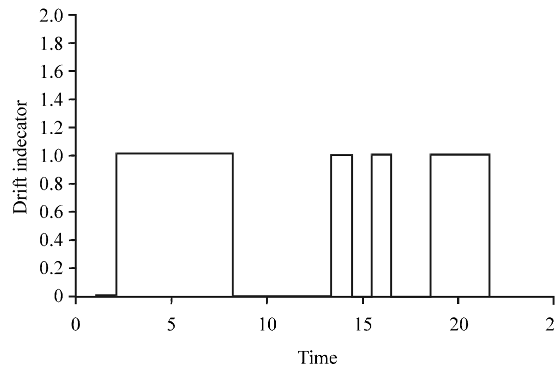

Fig. 10: Gradual drift in hyper plan

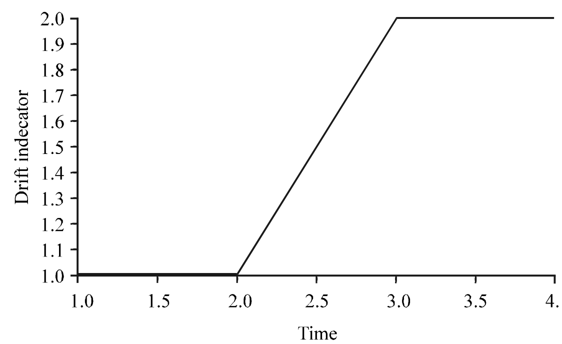

Fig. 11: Incremental drift in hyper plan data

Table 8: Evaluation of SIN dat aset

Data set	G-mean	Precision	Sensitivity	F1-measure	Specificity	Accuracy (%)
SIN1	0.98	1	0.96	0.98	1	96
SIN2	0.97	1	0.96	0.97	1	96
SIN3	0.97	1	0.95	0.97	1	95
SIN4	0.97	1	0.95	0.97	1	95

Table 9: Delay time and time accuracy in SIN data.

Real drift location	False drift	True drift start	Delay time	Time accuracy (%)
SIN1	67	211	11	94.78
SIN2	67	412	12	97.08
SIN3	67	614	14	97.77
SIN4	67	515	15	97.08

tightly related to past points and vice versa. If з is decreases the effect of past points is reduced and the mismatch between current and previous drifts also, decreases. Table 10 shows different values of з and its effect in the performance measures of drift detection.

The moving hyper plan dataset contains gradual drift that moves the classes gradually from its original distribution to another one and then returns to original state. The result of gradual drift by MPH on this dataset is shown in Fig. 10.

Also, this dataset considered incremental drift dataset where the data incrementally changed from start to end of it. Figure 11 shows the detected incremental drift.

Table 10: Effect of different

Data set	G-mean	Precision	Sensitivity	F1-measure	Specificity	Accuracy (%)
10^{-5}	0.98	1	0.96	0.98	1	96
10^{-4}	0.98	1	0.97	0.98	1	97
10^{-3}	0.96	1	0.94	0.96	1	94

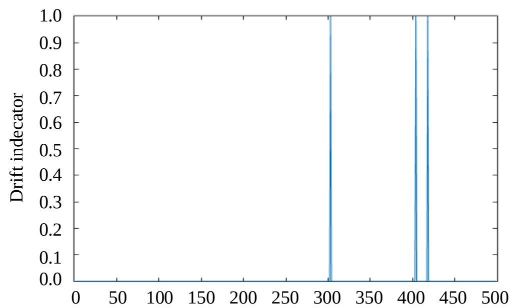

Fig. 12: Gradual drift in circle dataset

The circle dataset change the size of three circles gradually by increments the radius. Figure 12 displays the detected drift.

CONCLUSION

The growing number of modern applications produces evolving data stream, there is need for detecting changes. Drift detection is important in many applications such as alarm from diseases also can be used in forecasting system. In this study, a new model ARNN is proposed that reduces the error by a feedback to the input. Error reduction can help to detect real drift that changes the behavior of data rather than their distribution which is a virtual drift. Also, MPH method is used to detect drift and identify its type sudden, gradual incremental and give alarm when drift is detected. The MPH method has been evaluated by using many synthetic dataset with known drift locations. The MPH can detect drift and specify its type with high accuracy when evaluated with different types of evaluation measures. The selection of suitable increasing step factor helps in forgetting past values and give more importance to current data.

REFERENCES

Al-A'araji, N.H., E. Al-Shamery and A.H. Alyaa, 2016. A new polynomial curve fitting based on segmentation of variable point and variable modes for reconstructing missing values. Res. J. Appl. Sci., 11: 1089-1094.

Andrade, H.C., B. Gedik and D.S. Turaga, 2014. Fundamentals of Stream Processing: Application Design, Systems and Analytics. Cambridge University Press, New York, USA., ISBN:9781107015548, Pages: 530.

Bach, S.H. and M.A. Maloof, 2008. Paired learners for concept drift. Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08), December 15-19, 2008, IEEE, Pisa, Italy, ISBN:978-0-7695-3502-9, pp: 23-32.

Carney, D., U. Cetintemel, M. Cherniack C. Convey and S. Lee *et al.*, 2002. Monitoring streams: A new class of data management applications. Proceedings of the 28th International Conference on Very Large Data Bases (VLDB'02), August 20-23, 2002, VLDB Endowment, Hong Kong, China, pp: 215-226.

Chen, K., Y.S. Koh and P. Riddle, 2016. Proactive drift detection: Predicting concept drifts in data streams using probabilistic networks. Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN'16), July 24-29, 2016, IEEE, Vancouver, British Columbia, Canada, ISBN:978-1-5090-0621-2, pp: 780-787.

Gama, J., I. Zliobaite, A. Bifet, M. Pechenizkiy and A. Bouchachia, 2014. A survey on concept drift adaptation. ACM. Comput. Surv. CSUR., 46: 1-37.

Gama, J., R. Sebastiao and P.P. Rodrigues, 2009. Issues in evaluation of stream learning algorithms. Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09), June 28-July 1, 2009, ACM, New York, USA., ISBN:978-1-60558-495-9, pp: 329-338.

Ienco, D., A. Bifet, B. Pfahringer and P. Poncelet, 2014. Change detection in categorical evolving data streams. Proceedings of the 29th Annual ACM Symposium on Applied Computing (SAC'14), March 24-28, 2014, ACM, Gyeongju, Republic of Korea, ISBN:978-1-4503-2469-4, pp: 792-797.

Kadwe, Y. and V. Suryawanshi, 2015. A review on concept drift. J. Comput. Eng, 17: 20-26.

Khamassi, I. and M. Sayed-Mouchaweh, 2014. Drift detection and monitoring in non-stationary environments. Proceedings of the 2014 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), June 2-4, 2014, IEEE, Linz, Austria, ISBN:978-1-4799-3347-1, pp: 1-6.

Minku, L.L., A.P. White and X. Yao, 2010. The impact of diversity on online ensemble learning in the presence of concept drift. IEEE. Trans. Knowl. Data Eng., 22: 730-742.

Nabeel, A.A., E. Al-Shamery and A. Abdul-Hussein, 2017. An optimal stream prediction using adaptive regression neural network. J. Eng. Appl. Sci., 12: 8844-8850.

Pinage, F.A. and E.M.D. Santos, 2015. A dissimilarity-based drift detection method. Proceedings of the 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), November 9-11, 2015, IEEE, Vietri sul Mare, Italy, ISBN:978-1-5090-0163-7, pp: 1069-1076.

Ross, G.J., N.M. Adams, D.K. Tasoulis and D.J. Hand, 2012. Exponentially weighted moving average charts for detecting concept drift. Pattern Recogn. Lett., 33: 191-198.

Thakre, A.A. and S. Dongre, 2016. Review on concept drift detection techniques. Intl. J. Recent Innovation Trends Comput Commun., 4: 404-407.

Yao, Y. and L.B. Holder, 2016. Detecting concept drift in classification over streaming graphs. Proceedings of the KDD Workshop on Mining and Learning with Graphs (MLG), August 14, 2016, ACM, San Francisco, California, USA., pp: 2134-2142.

Zliobaite, I., M. Pechenizkiy and J. Gama, 2016. An Overview of Concept Drift Applications. In: Big Data Analysis: New Algorithms for a New Society, Japkowicz, N. and J. Stefanowski (Eds.). Springer, Switzerland, ISBN:978-3-319-26987-0, pp: 91-114.