

Real-Time Concept Drift Detection and Its Application to ECG Data

<https://doi.org/10.3991/ijoe.v17i10.25473>

Ketan Sanjay Desale^(✉), Swati Shinde
PCET's Pimpri Chinchwad College of Engineering, Pune, India
ketan.desale@pcceopune.org

Abstract—Prediction of cardiac disease is one the most crucial topics in the sector of medical info evaluation. The stochastic nature and the variation concerning time in electrocardiogram (ECG) signals make it burdensome to investigate its characteristics. Being evolving in nature, it requires a dynamic predictive model. With the presence of concept drift, the model performance will get worse. Thus learning algorithms require an apt adaptive mechanism to accurately handle the drifting data streams. This paper proposes an inceptive approach, Corazon Concept Drift Detection Method (Corazon CDDM), to detect drifts and adapt to them in real-time in electrocardiogram signals. The proposed methodology results in achieving competitive results compared to the methods proposed in the literature for all types of datasets like synthetic, real-world & time-series datasets.

Keywords—concept drift, Electrocardiogram signal, drift detection method, data streams

1 Introduction

In the real world, data is normally nonstationary. In much difficult data evaluation uses, data develops throughout the period and needs to be examined in near real-time [1]. This triggers complications since the forecasts may turn much less correct as the point in time goes by or possibilities to enhance the precision may be skipped. Concept Drifts relates to scenarios in the event that the relationship among the insight data and the focus on shifting, that any model is usually attempting to alter over time [2]. The challenge of concept drift is certainly of strengthening significance as a growing number of data is structured in the mode of data streams alternatively as opposed to stationary sources, and it is impractical to foresee that data allocation stays steady through a prolonged time frame. It is not confounded that the issue of concept drift is analyzed in different research areas including machine learning, data mining, data recuperation, and so on recommender systems [3].

Let us consider that data stream is a set of samples (X_k, y_k) where $k = 1, 2, 3, \dots$ where X is a feature vector, and y is the label that belongs to k number of classes. In evolving data streams, the data distribution is changing timely. Concept drift between time t_0 and t_1 is defined as,

$$\exists X: P_{t_0}(X, y) \neq P_{t_1}(X, y) \quad (1)$$

Where P_{t_0} indicates the joint probability between the input variable set X and the target variable y at time t .

Generally, concept drifts are distinguished concerning their behavior and speed of changes. The drift speed [4] is portrayed through the contrary to a transition period of drift between adjacent concepts. Abrupt concept drift occurs due to faster speed whereas the slower speed leads to the gradual concept drift. Moreover, new concepts may reoccur. The methodology of concept drift adaptation is binarized into active and passive approaches [5]. The active approach first finds the change points in the input data streams and then adapts the model, which is trigger-based adaptation. A proactive approach continuously adapts the model every time new instances arrive. The drift detectors are mechanisms that identify the change points. The concept drift detection methods are classified in following categories [6]:

- a) sequential analysis methods - SPRT [7], PH [8] and CUSUM [8]
- b) error rate-based detection methods - DDM [9], EDDM [10], ECDD [11] and RDDM [12],
- c) data distribution-based detection methods - ADWIN [13] and FHDDM [14].

2 Related Work

This section focuses on a few drift detection algorithms. Every algorithm's systematic review, framework, and logic are discussed.

2.1 Drift Detection Method (DDM)

The DDM is proposed by Gama et al. [9]. DDM is the most refereed drift detection algorithm which supervises the error rate of the base learner. Two levels are defined for concept drift detection in DDM, warning level, and drift level. The error rate p_t and its standard deviation s_t at time t are estimated. Updation of the variables p_{min} and s_{min} is done if,

$$p_t + s_t < p_{min} + s_{min} \quad (2)$$

It signals the warning state when,

$$p_t + s_t \geq p_{min} + 2s_{min} \quad (3)$$

and signals the drift state when,

$$p_t + s_t \geq p_{min} + 3s_{min} \quad (4)$$

The algorithm modifies itself every time drift is observed based on stored instances.

2.2 Early Drift Detection Method (EDDM)

Overcoming the limitations of DDM, a new algorithm EDDM was proposed by Baena-García et al. [10], that estimates the distance between two adjacent classification errors p_t , standard deviation s_t at time t and updates the variables p_{max} and s_{max} if,

$$p_t + 2s_t > p_{max} + 2s_{max} \quad (5)$$

The warning state is alarmed when,

$$(p_t + 2s_t)/(p_{max} + 2s_{max}) < \alpha \quad (6)$$

and the drift state is alarmed when,

$$(p_t + 2s_t)/(p_{max} + 2s_{max}) < \beta \quad (7)$$

After detecting the concept drift, the stored instances are used to relearn a new model.

2.3 Adaptive Windowing (ADWIN)

The ADWIN is a two-time window-based drift detection approach proposed by Bifet et al. [13]. Without defining the size of windows in advance, ADWIN takes the total size n of a window W . Then two sub-window sizes referring to the variation rate between two sub-windows w_{old} and w_{new} are computed.

2.4 Accurate Concept Drift Detection Method (ACDDM)

This Algorithm proposed by M.M.W. Yan [15] in 2020 detects the inconsistency of error rates using Hoeffding's inequality theory by analyzing the prequential error rates. The current error rate of the base learner is used to trigger concept drift. Let us assume a sequence of instances (X_j, y_j) where $j = 1, 2, 3, \dots$, instances that are used to test the model. The error is calculated as follows:

$$p_i = \sum_{k=1}^i L(y_k, \hat{y}_k) \quad (8)$$

where, L = loss function, \hat{y}_i = prediction and y_i = observed values.

3 Corazon Concept Drift Detection Method (Corazon CDDM)

Fig. 1 describes the proposed Corazon Concept Drift Detection Method (Corazon CDDM) methodology for real-time detection and adapting to drift in input data. The drift detector is responsible for detecting the variations in stream data. After detecting the drift feedback mechanism is applied to retrain the model. When a change has been detected by drift detector, the adaptation process is activated and it is labeled as model

retraining manager. This retraining manager is responsible for considering the previously learned model before proceeding with retraining. The objective is to don't discard previously learned model and fit it partially in retraining.

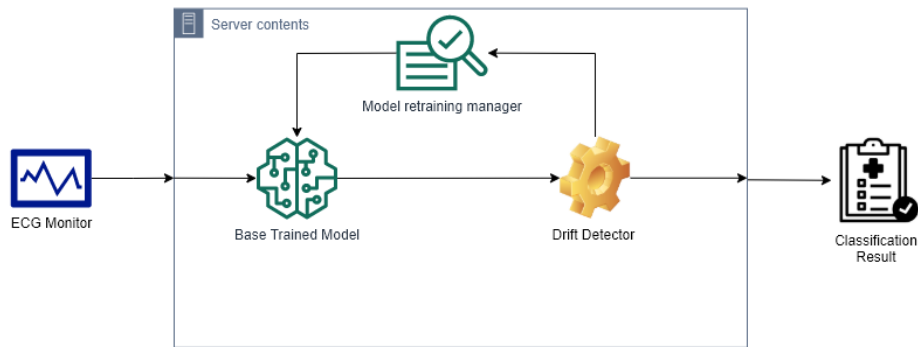


Fig. 1. Corazon Concept Drift Detection Method (CDDM) Methodology

3.1 Dataset descriptions

Synthetic datasets are created using Agrawal generators in Massive Online Analysis (MOA). For experimentation following real-world and synthetically generated datasets are used.

Agrawal. This stream generator [16] has 6 numeric attributes and 3 categorical attributes. These attributes indicate the random loan applications. Ten different functions can be produced commenting on the viability of loan approval. Variation in functions leads to concept drift. We used the six functions and kept perturbation noise 5%.

Electricity. This is a widely used dataset [17] and was collected from the Australian New South Wales Electricity Market. The dataset contains a total of 45,312 instances. Every example of the dataset refers to a period of 30 minutes. The price variation (increase or decrease) for the corresponding average of the last 24 hours is reflected in the class label.

MIT-BIH Arrhythmia Dataset. This database [18] contains 48 records of heartbeats at 360 Hz of 47 patients. Each record contains two ECG leads. Characteristics of the dataset are as follows:

- Number of Samples: 109446
- Number of Categories: 5
- Sampling Frequency: 125Hz
- Classes: 5 classes (N - 0, S - 1, V - 2, F - 3, Q - 4)

We have generated the synthetic streams by introducing the drifts at regular time intervals on a total number of instances. The details are described in Table 1. The advantage of generating the evolving data streams is it introduces heterogeneous concept changes with variable durations.

Table 1. Synthetic & Real-world datasets

Dataset	Instances	Attributes	Classes	Drift Width
Agrawal-60,000	60,000	9	2	100
Agrawal-600,000	6,00,000	9	2	1000
Electricity	45,312	8	2	--
MIT-BIH	1,09,446	2	5	--

3.2 Algorithm implementation

Algorithm 1 describes the comprehensive model used for training the input data. The model m is trained over Input data and put it in $\text{fit}(x,y)$ function. Algorithm 2 takes care of detecting the drift in the input stream and adapt to it in real-time. Once the drift is detected, the previously learned model is cloned and relearning continues on newly arriving data.

Algorithm 1: Trains the model on training data

```

1: m = MODEL
2: data = TRAINING DATA
3: while epoch  $\neq$  0 do
4:   while data  $\neq$  0 do
5:     x  $\leftarrow$  data.loadX()
6:     y  $\leftarrow$  data.loadY()
7:     m.fit(x, y)
8:   end while
9: end while

```

Algorithm 2: Executes the model over data

```

1: m = MODEL
2: stream = INCOMING STREAM
3: detector = DRIFT DETECTOR
4: while stream do
5:   x  $\leftarrow$  stream.loadX()
6:   y  $\leftarrow$  stream.loadY()
7:   p  $\leftarrow$  m.predict(x)
8:   if detector.detect(p, y) then
9:     m  $\leftarrow$  clone(m)
10:    m.fit(x, y)
11:   end if
12:   m.partial fit(x, y)
13:   yield p
14: end while

```

4 Experimental Results

Naive Bayes, Stochastic Gradient Descent & Very Fast Decision Tree (VFDT) [19] are used as base learners. For synthetic datasets, different random sequences are generated after running the drift detection program 5 times. Whenever concept drift is found, the algorithm relearns the new model as mentioned in Algorithm 2, and then, the precision, recall, and prequential accuracy were computed. Precision and recall are two significant model evaluation metrics. Precision, which is also known as a positive predictive value, is the percentage of your results that are relevant, while recall, which is also known as sensitivity, refers to the percentage of total relevant results correctly classified by an algorithm.

Table 2 & Table 3 are listing the results when SGD is used as a base classifier. ACDDM gives the highest accuracy for Agrawal-60,000, Agrawal-600,000 dataset & MIT-BIH datasets, while ADWIN is performing better for Electricity dataset. Table 4 & Table 5 are listing the results when Naive Bayes is used as a base classifier. Overall Naive Bayes is not able to produce quantifiable results except for Electricity dataset. For Agrawal-60,000 and Agrawal-600,000 datasets, all drift detection methods result in almost the same accuracy of 68.64%. Being a multiclass dataset, Naive Bayes does not work properly on the MIT-BIH dataset.

Table 2. Drift Detection with SGD as base learner (Part 1)

Dataset	Detector	Precision	Recall	Accuracy (%)
Agrawal-60,000	DDM	0.51 ± 0.10	0.55 ± 0.069	63.89 ± 6.79
	ADWIN	0.50 ± 0.09	0.52 ± 0.08	63.05 ± 4.98
	ACDDM	0.50 ± 0.11	0.55 ± 0.09	64.17 ± 6.41
Agrawal-600,000	DDM	0.45 ± 0.23	0.49 ± 0.07	56.75 ± 3.74
	ADWIN	0.47 ± 0.10	0.48 ± 0.06	54.95 ± 3.75
	ACDDM	0.44 ± 0.23	0.49 ± 0.07	57.53 ± 3.07
Electricity	DDM	0.82	0.82	84.93
	ADWIN	0.82	0.83	85.28
	ACDDM	0.77	0.80	82.24

Table 3. Drift Detection with SGD as base learner (Part 2)

Data set	Detector	Precision					Recall					Accuracy (%)
		N	S	V	F	Q	N	S	V	F	Q	
MIT-BIH	DDM	0.96	0.11	0.32	0.049	0.81	0.91	0.31	0.43	0.18	0.86	87.57
	ADWIN	0.96	0.12	0.31	0.038	0.81	0.91	0.38	0.45	0.15	0.86	87.86
	ACDDM	0.97	0.13	0.33	0.028	0.84	0.91	0.48	0.51	0.26	0.92	89.11

Table 4. Drift Detection with Naïve Bayes as base learner (Part 1)

Dataset	Detector	Precision	Recall	Accuracy (%)
Agrawal-60,000	DDM	0.55 ± 0.04	0.61 ± 0.05	68.64 ± 3.13
	ADWIN	0.55 ± 0.04	0.61 ± 0.05	68.64 ± 3.13
	ACDDM	0.55 ± 0.04	0.61 ± 0.05	68.64 ± 3.13
Agrawal-600,000	DDM	0.59 ± 0.19	0.60 ± 0.05	66.89 ± 2.5
	ADWIN	0.59 ± 0.19	0.60 ± 0.05	66.86 ± 2.53
	ACDDM	0.60 ± 0.18	0.60 ± 0.05	66.89 ± 2.5
Electricity	DDM	0.59	0.65	69.30
	ADWIN	0.68	0.72	75.54
	ACDDM	0.45	0.78	71.61

Table 5. Drift Detection with Naïve Bayes as base learner (Part 2)

Data Set	Detector	Precision					Recall					Accuracy (%)
		N	S	V	F	Q	N	S	V	F	Q	
MIT-BIH	DDM	0.026	0.046	0.075	0.66	0.35	0.89	0.09	0.15	0.007	0.092	5.90
	ADWIN	0.46	0.097	0.088	0.35	0.047	0.83	0.026	0.071	0.007	0.12	39.41
	ACDDM	0.085	0.11	0.19	0.20	1.0	0.93	0.41	0.19	0.028	0.092	16.04

Table 6 & Table 7 describes the performance results of experiments using VFDT as a base learner. For Agrawal-60,000, Agrawal-600,000 and Electricity datasets ACDDM archives the highest accuracy, but DDM also shows competitive results compared to ACDDM. For the MIT-BIH and Electricity dataset results of DDM & ACDDM are the same.

Table 6. Drift Detection with VFDT as base learner (Part 1)

Dataset	Detector	Precision	Recall	Accuracy (%)
Agrawal-60,000	DDM	0.94 ± 0.03	0.94 ± 0.04	95.41 ± 1.86
	ADWIN	0.91 ± 0.06	0.78 ± 0.19	83.88 ± 13.28
	ACDDM	0.94 ± 0.03	0.94 ± 0.04	95.45 ± 1.89
Agrawal-600,000	DDM	0.94 ± 0.03	0.95 ± 0.03	95.52 ± 1.91
	ADWIN	0.92 ± 0.06	0.78 ± 0.19	83.99 ± 13.39
	ACDDM	0.94 ± 0.03	0.94 ± 0.03	95.54 ± 1.92
Electricity	DDM	0.75	0.80	81.28
	ADWIN	0.68	0.69	73.30
	ACDDM	0.75	0.80	81.28

Table 7. Drift Detection with VFDT as base learner (Part 2)

Data set	Detector	Precision					Recall					Accuracy (%)
		N	S	V	F	Q	N	S	V	F	Q	
MIT-BIH	DDM	0.98	0.62	0.74	0.21	0.88	0.96	0.82	0.88	0.42	0.92	94.97
	ADWIN	0.98	0.40	0.56	0	0.81	0.94	0.36	0.85	0	0.95	92.35
	ACDDM	0.98	0.62	0.74	0.21	0.88	0.96	0.82	0.88	0.42	0.92	94.97

Fig. 2 shows the accuracy results on electricity dataset. Real-world data streams are prone to frequent changes in data distributions. For electricity dataset, ADWIN drift detector with SGD as base learner produces the highest accuracy of 85.28%. Overall, it is observed that Corazon CDDM method with ADWIN drift detector gives little bit increased accuracy of 85.28% than any other drift detectors experimented in literature. Whereas, for the MIT-BIH dataset, as shown in Fig. 3, ACDDM detector achieves accuracy of 94.97% & DDM detector achieves accuracy of 89.11% when VFDT is used as base learners using Corazon CDDM approach.

After analyzing all experimental results on Agrawal synthetic dataset, we found that Corazon CDDM methodology with ACDDM drift detector achieves maximum accuracy, precision and recall. If we compare the ACDDM drift detector performance in [15] with proposed methodology, Corazon CDDM shows remarkable improvement in accuracy when VFDT is used as base learner. Table 8 is showing comparative summary of these results. We observed remarkable growth in performance of ACDDM drift detector, where accuracy is increased to $95.45 \pm 1.86\%$ from $76.02 \pm 1.09\%$ for Agrawal-60,000 dataset and for Agrawal-6,00,000 dataset accuracy increased from $84.85 \pm 1.27\%$ to $95.54 \pm 1.92\%$.

Table 8. Comparison with existing methods

Dataset	ACDDM	Corazon CDDM
Agrawal-60,000	76.02 ± 1.09	95.45 ± 1.86
Agrawal-6,00,000	84.85 ± 1.27	95.54 ± 1.92
Electricity	84.81	81.29
MIT-BIH	NA	94.97

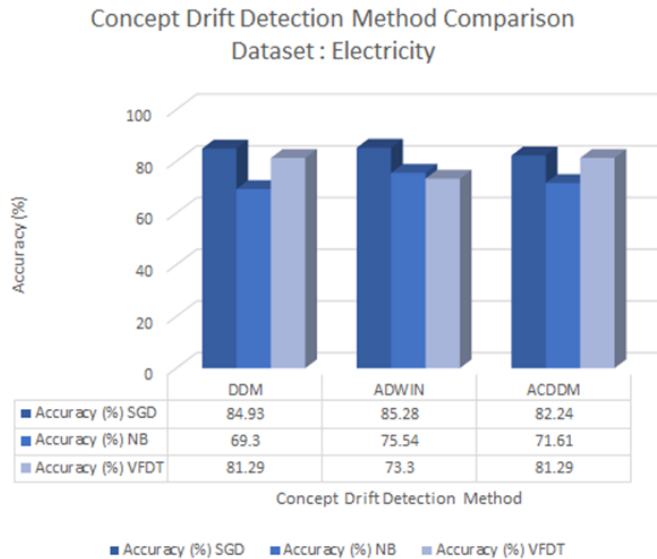


Fig. 2. Base Learner Performance (accuracy) on Drift Detection Methods for Electricity Dataset

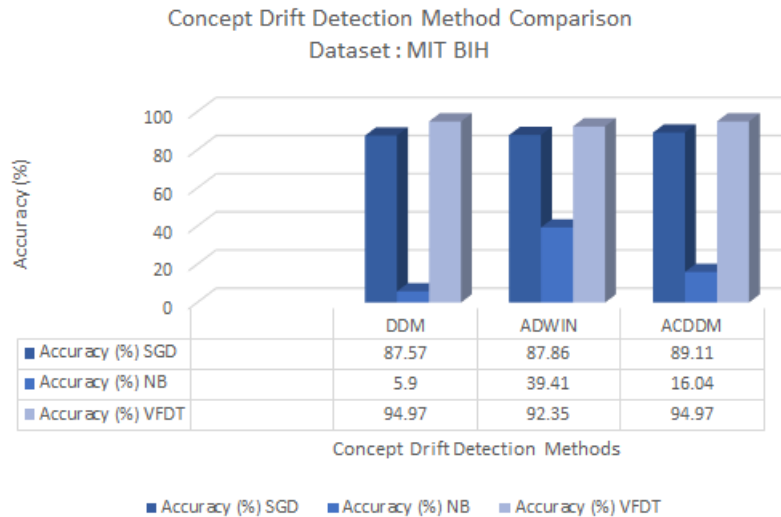


Fig. 3. Base Learner Performance (accuracy) on Drift Detection Methods for MIT-BIH dataset

In summary, the proposed approach shows notable performance for synthetic datasets and real-world MIT-BIH dataset. As these datasets contains sufficient number of instances for learning, the performance of Corazon CDDM is outstanding. But on the contrary, real world dataset like electricity didn't produce expected accuracy with proposed approach, due to lack of sufficient learning instances. Overall combination of incremental learning process of VFDT and feedback mechanism of Corazon CDDM befit perfectly for both synthetic and real world stream datasets.

5 Conclusion and Future Work

In this paper, the inception approach of concept drift detection and adaption in time series data i.e., MIT-BIH is introduced and experimented with various detectors. To date ECG signals were not analyzed for the presence of drift thus no methodology was proposed for the detection of drift in it. The Corazon CDDM approach uses feedback mechanism after detection of drift and considers previously learned model into consideration while progressing further with retraining. Experimental results on synthetic datasets and MIT-BIH dataset show the effectiveness of the proposed adaptive approach. Future work will be focused on experimenting adaptive approach using Deep Learning algorithms as base learners to improve the performance.

6 References

- [1] Jaka Demšar, Zoran Bosnić, Detecting concept drift in data streams using model explanation. *Expert Systems with Applications*. Volume 92, 2018, pp. 546-559, <https://doi.org/10.1016/j.eswa.2017.10.003>

- [2] K. S. Desale, C. N. Kumathekar and A. P. Chavan, "Efficient Intrusion Detection System Using Stream Data Mining Classification Technique," 2015 International Conference on Computing Communication Control and Automation, 2015, pp. 469-473, <https://doi.org/10.1109/iccubea.2015.98>
- [3] A. Liu, J. Lu and G. Zhang, "Diverse Instance-Weighting Ensemble Based on Region Drift Disagreement for Concept Drift Adaptation," in IEEE Transactions on Neural Networks and Learning Systems, <https://doi.org/10.1109/tnnls.2020.2978523>
- [4] L.L. Minku, A.P. White, X. Yao, The impact of diversity on online ensemble learning in the presence of concept drift, IEEE Trans. Knowl. Data Eng. 22 (5) (2010) 730–742, <https://doi.org/10.1109/tkde.2009.156>
- [5] L.I. Kuncheva, Classifier ensembles for changing environments, in: Proceedings of the International Workshop on Multiple Classifier Systems, Lecture Notes in Comput. Sci. 3077 (2004) 1–15, https://doi.org/10.1007/978-3-540-25966-4_1
- [6] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Comput. Surv. 46 (4) (2014) 37, 44:1-44, <https://doi.org/10.1145/2523813>
- [7] A. Wald, Sequential Analysis, John Wiley and Sons, Inc, 1947.
- [8] E.S. Page, Continuous inspection schemes, Biometrika 41 (1–2) (1954) 100–115, <https://doi.org/10.1093/biomet/41.1-2.100>
- [9] J. Gama, P. Medas, G. Castillo, P. Rodrigues, Learning with drift detection, in: Proceedings of the Brazilian Symposium on Artificial Intelligence, SBIA'04, Lecture Notes in Comput. Sci. 3171 (2004) 286–295, https://doi.org/10.1007/978-3-540-28645-5_29
- [10] M. Baena-García, J. Del Campo-Ávila, R. Fidalgo, A. Bifet, Early drift detection method, in: Proceedings of the ECML PKDD International Workshop on Knowledge Discovery from Data Streams, IWKDDs'06, 2006, pp. 77-86.
- [11] G.J. Ross, N.M. Adams, D.K. Tasoulis, D.J. Hand, Exponentially weighted moving average charts for detecting concept drift, Pattern Recognit. Lett. 33 (2012) 191–198, <https://doi.org/10.1016/j.patrec.2011.08.019>
- [12] R.S.M. Barros, D.R.L. Cabral, Goncalves Jr. P.M., S.G.T.C. Santos, RDDM: Reactive Drift Detection Method, Expert Syst. Appl. 90 (30) (2017) 344–355, <https://doi.org/10.1016/j.eswa.2017.08.023>
- [13] A. Bifet, R. Gavalda, Learning from time-changing data with adaptive windowing, in: Proceedings of the SIAM International Conference on Data Mining, 2010, pp. 443-448, <https://doi.org/10.1137/1.9781611972771.42>
- [14] A. Pesaraghader, H. Viktor, Fast hoeffding drift detection method for evolving data streams, in: Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Comput. Sci. 9852 (2016) 96–111. https://doi.org/10.1007/978-3-319-46227-1_7
- [15] M.M.W. Yan, Accurate detecting concept drift in evolving data streams, ICT Express (2020), <https://doi.org/10.1016/j.ict.2020.05.011>
- [16] R. Agrawal, T. Imielinski, A. Swami, Database mining: A performance perspective, IEEE Trans. Knowl. Data Eng. 5 (6) (1993) 914–925, <https://doi.org/10.1109/69.250074>
- [17] M. Harries, N.S. Wales, Splice-2 Comparative Evaluation: Electricity Pricing, Technical Report, University of New South Wales, Australia, 1999.
- [18] Massachusetts Institute of Technology, MIT-BIH ECG database, 2011, Available at: <http://ecg.mit.edu/>
- [19] M. Somvanshi, P. Chavan, S. Tambade and S. V. Shinde, "A review of machine learning techniques using decision tree and support vector machine," 2016 International Conference on Computing Communication Control and Automation (ICCUBEA), 2016, pp. 1-7, <https://doi.org/10.1109/iccubea.2016.7860040>

7 Authors

Ketan Sanjay Desale is currently a PhD student at Department of Computer Engineering, PCET's Pimpri Chinchwad College of Engineering (PCCOE), Pune, India. He completed his ME Degree in Computer Networks at D. Y. Patil School of Engineering Y technology, Pune University, Pune between years 2013 and 2015. Between the year 2007 and 2011 he completed his BE Degree in Information Technology from PCET's Pimpri Chinchwad College of Engineering (PCCOE), Pune, India. (Personal Website: <https://ketandesale.weebly.com>)

Dr. Swati Shinde is currently working as Professor in Department of Computer Engineering, PCET's Pimpri Chinchwad College of Engineering (PCCOE), Pune, India. She completed her PhD in Computer Science & Engineering. She is also working as Principal Investigator - DST Funded Research Project, NVIDIA Ambassador and Certified Instructor for Deep Learning. Her research interest is in Artificial Neural Networks, Intelligent Systems, Deep Learning and Fuzzy Systems. (Personal Website: <https://sites.google.com/view/swatishinde/home>).

Article submitted 2021-07-03. Resubmitted 2021-08-04. Final acceptance 2021-08-05. Final version published as submitted by the authors.