An Improved under Sampling Approaches for Concept Drift and Class Imbalance Data Streams using Improved Cuckoo Search Algorithm

Tirupathi Rao Gullipalli[a], and Dr. Bhanu Prakash Battula[b]

[a]
Research Scholar in Department of CSE, Acharya Nagarjuna University, Guntur
[b]Principal, GVR&S College of Engineering and Technology Guntur.

Article History: Received: 11 January 2021; Accepted: 27 February 2021; Published online: 5 April 2021

Abstract: One of the biggest challenges in the recent times in the field of data stream learning is to mitigate the presence of concept drift. There are numerous challenges in overcoming the concept drift, such as changing class ratio, huge volume of data and real time processing for effective knowledge discovery. Evolutionary search techniques are one of the new paradigms to handle huge dimensionality and scalability of the data streams. One of the finest and least applied evolutionary search approaches is the cuckoo search technique for data streams. To solve both the concept drift and class imbalance issues simultaneously, in this paper we have proposed an approach using nature inspired evolutionary optimizing technique known as Cuckoo Feature and Instance Selection (CFIS) algorithm. The performance evaluation of the proposed approach is done on an exclusive experimental setup of 15 data streams formed and compared with two data stream approach. Moreover, a set of six evaluation criteria's are considered for showing overall better performance of the proposed approach in the presence of concept drift and class imbalance.

Keywords: Data Streams, evolutionary learning, optimization technique, Cuckoo Feature and Instance Selection (CFIS).

1. Introduction

Data streams are the data which are generated continuously and collected with different change in the characteristics of the data. Data streams have some unique characteristics such as large volume of data, change in the nature of the data i.e concept drift and huge storage capability before processing [1]. In this scenario, the analytical processing of data streams for knowledge discovery requires algorithmic approaches suitable for streaming platforms. There are different challenges for data stream which include classification, clustering, Association analysis etc. [2]. Evolutionary algorithmic approaches are one of the best techniques to deal with data streams of concept drift and class imbalance nature. One of the evolutionary search techniques is cuckoo search.

The details of the cuckoo search are provided in the following subsections. The Cuckoo Optimization Algorithm is motivated by the life of the cuckoo bird [3]. This novel optimization algorithm is focused on the bird's unique breeding and egg laying behavior. In this model, adult cuckoos and eggs were included. Adult cuckoos lay their eggs in the nests of other species. If the eggs are not found and removed by the host birds, they may develop into a mature cuckoo. Cuckoo movement and environmental conditions can ideally lead them to converge and find the best position for reproduction and breeding. Yang and Deb created Cuckoo Optimization in 2009 [3], which was influenced by nature. Rajabioun developed the Cuckoo Optimization Algorithm in 2011[4]. The Cuckoo Optimization Algorithm (COA) is a brand-new continuous all-aware search algorithm inspired by the life of a cuckoo pigeon.COA, like other meta heuristics, continues with a primary community of cuckoos. Other host birds' environments are used by these cuckoos to deposit their nests. The habitat in COA is defined by a random group of possible solutions.

Cuckoo Search

The Cuckoo Quest is a global optimization method for finding critical non-circular slip surfaces that is very easy and effective. It doesn't need any trial surfaces or search artifacts from the consumer.

The criteria for the Cuckoo Hunt are as follows:

• Number of Surface Vertices at the Start

• Number of Surfaces to Store

Initial Number of Surface Vertices

The Cuckoo Quest produces each trial slip surface with an initial number of vertices. The final number of vertices on a slip surface may vary from the initial number due to the Cuckoo Search algorithm and subsequent Optimization.

The default value of 8 vertices is a reasonable amount to use in general. The true global minimum surface can not be found if this amount is too big. If the amount is too high, the computation can take longer without actually improving the outcome. The maximum value for the initial number of surface vertices should not surpass half of the Number of Slices specified in Project Settings, according to a rule of thumb.

Number of Surfaces to Store

The Cuckoo Quest decides the generation of each new surface dependent on the effects of previously determined slip surfaces. The maximum amount of previously measured slip surface effects that are processed in order to decide the next surface to compute is defined by the Number of Surfaces to Store. It is proposed that you use the default value of 1000.

Allow Surface with entrance and exit at same elevation

We typically avoid generating slip surfaces with the first and last points on the same elevation in a traditional slope stability issue (e.g. slip surfaces on a horizontal surface). In certain instances, you may want to measure certain surfaces, so check this box to enable the Cuckoo Quest to evaluate them.

Optimize Surfaces

Cuckoo Search has the Optimize Surfaces option allowed by example. This provides an extra optimization to the Cuckoo Search's minimum safety factor surface, typically resulting in a lower safety factor. For a Cuckoo Hunt, it is suggested that this choice be allowed at all times. For more details, see the Optimize Surfaces issue. The normal flow map for the manuscript is shown in Figure 1.

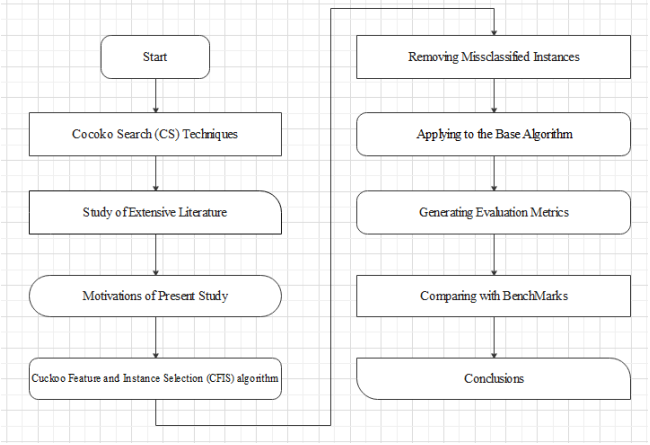

Figure 1 Flow Chart of Manuscript

The remainder of this paper is structured in the following manner. The proposal's motivation and associated work are discussed in Section 2. Section 3 discusses the proposed methods for coping with idea drift in the presence of imbalance results. Section 4 explains the experimental simulation testing approach. Section 5 includes the experiment findings and discussion, while Section 6 contains the conclusion.

2. Motivation

Yang & Deb (2009) [3] suggested a global optimization approach focused on the actions of cuckoos. The breeding activity styles are laid their eggs in host nests; if the eggs are not found and discarded, the hosts hatch the eggs into chicks.

The cuckoo search algorithm (CSA) is focused on the obligate brood parasitism of some cuckoo types, which means they lay their eggs in the nests of other birds. In this reproductive operation, there are two probable consequences for a cuckoo egg. The cuckoo egg will hatch and move on to the next generation if the host bird does not notice the cuckoo egg, or if the host bird notices the cuckoo egg and either throws it away or abandons its nest to create a new one. The CSA approach was influenced by the two described phenomena for two phases of new solution generation: discovery via Levy flights (the first phenomenon) and exploitation via substitution of a fraction of eggs (the second phenomenon) (the second phenomenon). Cuckoo quest has been included in many articles published in the literature. In the next segment, we'll look at a couple of the publications that were chosen.

Cuckoo Quest (CS) is a modern meta-heuristic algorithm developed by Xin-She Yang et al. [4] for solving optimization problems focused on the obligate brood parasitic activity of some cuckoo species coupled with the Levy flight behavior of some birds and fruit flies. A modern Cuckoo search algorithm focused on dynamically increasing swapping parameters has been suggested by M. Mareli et al. [5]. The flipping parameter is used to hold the local and global random walks in check. They came to the conclusion that an increasingly increasing switching parameter is a desirable function for improved algorithm efficiency. Cuckoo quest (CS) is a modern metaheuristic optimization algorithm introduced by Amir Hossein Gandomi et al. [6] for solving structural optimization tasks.

Rohit Salgotra a et al.,[7] have proposed different versions of cuckoo search algorithm using Cauchy operator to generate the step size, division of population and division of generations are also used for improving the exploration and exploitation during the search process. Venkata Vijaya Geeta et al.,[8] have reviewed different categories of cuckoo search algorithm which are used for increases the efficiency, accuracy, and convergence rate of the process. The different life stages of cuckoo bird such as cuckoo egg laying and breeding are used as phases in the algorithm for optimization. Yang, X et al.,[9] have developed a multi species cuckoo search method for effective nonlinear multi modal design case applications. In this scenario, different multi modal designs are considered as multiple cuckoo species interacting with each other and co-host for evolution and survival of fittest.

Jaddi NS et al.,[10] have used cuckoo search optimization technique for assigning weights for different links of artificial neural networks. Choosing proper weights is the toughest part in the building of artificial neural networks and it was well handled by using the cuckoo search optimization technique. Yung C Shih [11] have proposed a cuckoo search algorithm for development of maximally distributed physical Arrangements of different parameters such as egg weight and the acquired learning of host birds for obtaining low degree of distribution. A. Bustamam et al.,[12] have used cuckoo search optimization algorithm for finding inflation factor in the Markov clustering process for stochastic flow simulation.

Xin-She Yang et al.,[13] have developed a new stochastic function for solving engineering design

optimization problems. The developed solution is combined with the standard model test function for effective design of springs and welded beam structures. Celso A. G. S et al.,[14] have developed a novel optimization technique using cuckoo evaluation search and watershed erosion simulation programme for global optimization solutions in different problems of real world. Zhigang Lian et al.,[15] have proposed a new cuckoo search optimization technique and combined with PSO to solve optimization problems.

3. Proposed Approaches

The proposed approach consists of the following phases:

In the initial phase, the continuous flows of data sources are mapped to form a data stream of class imbalance in nature. The individual chunks of data are class imbalance in nature and the combined incremental data stream is also in the state of class imbalance nature with concept drift. The class imbalance issue can be overcome using different strategies of oversampling, under sampling and hybrid sampling etc. More specifically, in this paper we are using under sampling technique for class imbalance using the evolutionary techniques.

In the next phase, the required under sampling for the arriving chunks to be performed by using cuckoo search techniques. The cuckoo search techniques is described in the below sub sections.

3.1 Cuckoo Breeding Behavior Strategy

3.1.1 The Egg

An egg in a nest represents one person in the community, whereas a cuckoo egg represents a different approach for a particular place in the population. A Hamiltonian course is the same as an egg. The dealer is in charge of the tour's course.

3.1.2 The Nest:

The number of nests does not change. A nest is a single part of the group, and the number of nests is proportional to the population's scale. A rid nest is where one of the population's representatives is replaced by a different one. A nest may have several eggs, but each nest only has one egg for simplicity's sake.

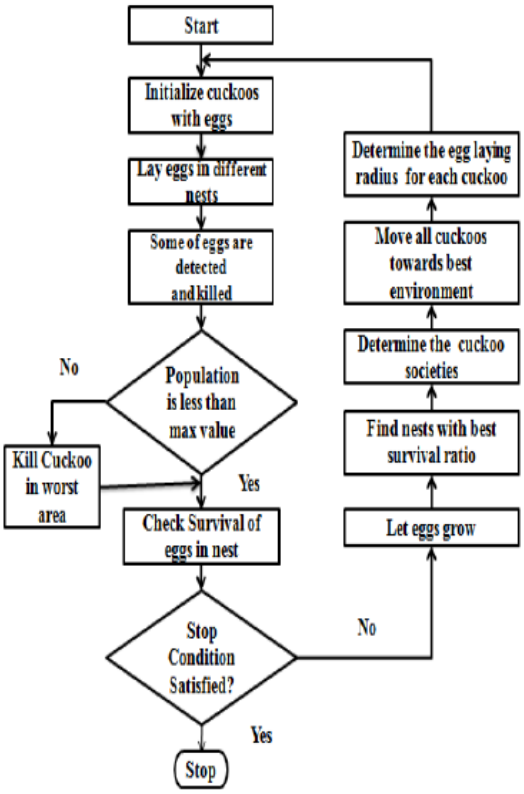

Figure 2 Flow Chart of Cuckoo Search Evolutionary Algorithm

3.1.3 Objective Function

The quality of a solution in the travelling salesman problem is proportional to the length of the Hamiltonian road. The shortest Hamiltonian route is the best alternative.

3.1.4 Search Space

Adjust the exact values of a nest's coordinates to change its position. Changing nests or nest sites do not place any real restrictions. This is particularly popular in continuous optimization problems, and it can be thought of as a value that eliminates functional obstacles to transferring a solution from one neighborhood to another, such as the representation of coordinates in TSP's solution space. The visiting order between cities may be modified since the city coordinates are set coordinates of the visited cities.

3.2 Cuckoo Search Mechanism

3.2.1 Lévy Flights mechanism-

The cuckoo hunt has been applied based on the three laws. Lévy flight is used to create a new approach x t+1 I for the I th cuckoo. This move is known as a global random walk, and it is supported by
The levy flight random global walk

$$x_i^{t+1} = x_i^t + \alpha \otimes Le'ıy(\lambda)\left(x_{best} - x_i^t\right)$$

The levy flight random local walk

$$x_i^{t+1} = x_i^t + \alpha \otimes H(p - \epsilon) \otimes \left(x_j^t - x_k^t\right)$$

where *x t i*is the previous solution, α> 0 is the step size related to problem scales and _is entry wise multiplication. Here *x t j and x t k* are randomly selected solutions and *x best* is the current best solution. In present work, the random step length via Lévy flight is considered due to more efficiency of Lévy flights in exploring the search space and is drawn from a Lévy distribution having infinite variance and mean.

$$Le'ıy \sim \left\{ \frac{\lambda\Gamma(\lambda)\sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda,}} (s \gg s_0 > 0) \right.$$

3.2.2 Cuckoo Search Implementation steps-
The cuckoo search can be well implemented as a effective search mechanism for choosing better individuals form the existing population. The process can also be implemented as a optimization problem from NP hard problems.
3.2.3 Generating initial cuckoo habitat
In the Initial state, cuckoo birds migrate from other environment and choose the host bird nest for laying eggs. This process of selecting nests for laying eggs depends on the initial properties of the cuckoo birds and the nature of the host bird. The rate of success of the cuckoo birds also depends on the best strategy followed by the cuckoo birds.
3.2.4 Cuckoos' style for egg laying
Cuckoo birds have the capability to produce the eggs which imitate the host eggs. This similar eggs laying technique will help the cuckoo birds to reduce the chances for detection of foreign eggs in the nest.
3.2.5 Immigration of cuckoos
Some of the cuckoo birds will migrate to the new environment for lesser competition or greater success. The migration policy is also one of the crucial decisions by the cuckoo birds for their success. There is a risk of complete elimination of the cuckoo bird of the new environment of migrating is not suitably selected.
3.2.6 Eliminating cuckoos in worst habitats
If the cuckoo birds are not successful in a specific environment and as a result the population of the cuckoos' gets reduced and finally elimination of cuckoo birds occurs.

Cuckoo Feature and Instance Selection (CFIS) algorithm

1. *Create a node N*
2. ***If** samples in N are of same class, C **then***
3. *return N as a leaf node and mark class C;*
4. ***If** A is empty**then***
5. ***return** N as a leaf node and mark with majority class;*
6. ***else***
7. **Begin**
8. *Objective function* f(x), x = (x1, ..., xd)T
9. *Generate initial population of*
10. n *host nests* xi (i = 1, 2, ..., n)
11. **while**(t <*MaxGeneration) or (stop criterion)*
12. *Get a cuckoo randomly by L´evy flights*
13. *evaluate its quality/fitness* Fi
14. *Choose a nest among* n *(say,* j*) randomly*
15. **if***(*Fi > Fj*),*
16. *replace* j *by the new solution;*
17. **end**
18. *A fraction (*pa*) of worse nests*
19. *are abandoned and new ones are built;*
20. *Keep the best solutions*
21. *(or nests with quality solutions);*
22. *Rank the solutions and find the current best*
23. **end while**
24. *Post process results and visualization*
25. **End**
26. *apply Gain Ratio(* D_w , A_w *)*
27. *label root node N as f(A)*

28. ***for*** *each outcome j of f(A)****do***
29. *subtree j =New Decision Tree(Dj,A)*
30. *connect the root node N to subtree j*
31. ***endfor***
32. ***endif***
33. ***endif***
34. *Return N*

---

4. Research Methodology

Table 1 UCI datasets and their properties

S.no. Dataset	Inst	Attributes	IR
1. car	1728	7	3.15
2. german_credit+	1000+	21	2.33
3. hypothyroid+	3772+	30	17.94
4. mfeat+	2000+	217	1.00
5. nursery+	12960+	9	1.01
6. page-blocks+	5473+	11	14.93
7. segment+	2310+	20	1.00
8. sick+	3772+	30	15.32

+ indicates the continuous data forming the data streams which are in class imbalance nature due to the high class imbalance ratio.

The experimental methodology used in the study consists of 8 class imbalance data sets [16] used for evaluating the performance of proposed and compared algorithms. For the purpose of making the experimental set up difficult, we have made the data chunks arriving from different data sources with different sizes.

5. EXPERMENTAL RESULTS

The experimental results of the proposed approaches CFIS are compared with Hoeffding Tree [17] and CS Forest [18]. The evaluation metrics used in the experimental validation are accuracy, AUC, TP Rate, TN Rate, FP Rate and FN Rate.

Table 2 Summary of Accuracy results on UCI datasets

Datasets	HoeffdingTree	CSForest	CFIS
Car	85.46	70.02	**98.47**
german_credit+	80.26	62.65	**92.89**
Hypothyroid+	87.80	77.47	**95.01**
mfeat+	90.11	86.56	**93.94**
nursery+	92.20	88.72	**92.19**
page-blocks+	91.13	92.69	**95.71**
segment+	85.42	94.82	**96.63**
sick+	89.59	94.35	**97.12**

Table 3 Summary of Area Under ROC results on UCI datasets

Datasets	HoeffdingTree	CSForest	CFIS

car	0.982	0.977	**1.000**
german_credit+	0.883	0.864	**0.960**
hypothyroid+	0.908	0.682	**0.975**
mfeat+	0.950	0.840	**0.981**
nursery+	0.975	0.920	**0.990**
page-blocks+	0.888	0.954	**0.987**
segment+	0.936	0.977	**0.990**
sick+	0.744	0.738	**0.962**

Table 4 Summary of True Positive Rate results on UCI datasets

DatasetsHoeffdingTreeCSForest		CFIS	
car	0.958	**1.000**	0.992
german_credit+	0.910	0.705	**0.945**
hypothyroid+	0.951	0.852	**0.965**
mfeat+	0.935	0.906	**0.971**
nursery+	0.967	0.953	**0.985**
page-blocks+	0.960	0.973	**0.990**
segment+	0.962	0.982	**0.990**
sick+	0.977	**0.991**	0.988

Table 5 Summary of False Positive Rate results on UCI datasets

DatasetsHoeffdingTreeCSForest		CFIS	
car	0.199	1.000	**0.016**
german_credit+	0.354	0.557	**0.122**
hypothyroid+	0.412	0.778	**0.127**
mfeat+	0.208	0.389	**0.065**
nursery+	0.104	0.194	**0.032**
page-blocks+	0.297	0.221	**0.038**
segment+	0.174	0.112	**0.021**
sick+	0.537	0.556	**0.109**

Table 6 Summary of True Negative Rate results on UCI

DatasetsHoeffdingTree	CSForest	CFIS	
car	0.801	0.000	0.984
german_credit+	0.645	0.442	0.877
hypothyroid+	0.587	0.221	0.872
mfeat+	0.791	0.610	0.934
nursery+	0.895	0.805	0.967
page-blocks+	0.702	0.778	0.961
segment+	0.825	0.887	0.978
sick+	0.462	0.443	0.890

Table 7 Summary of False Negative Rate results on UCI

Datasets	HoeffdingTree	CSForest	CFIS

car	0.042	**0.000**	0.008
german_credit+	0.090	0.294	**0.054**
hypothyroid+	0.049	0.147	**0.034**
mfeat+	0.064	0.093	**0.028**
nursery+	0.032	0.046	**0.014**
page-blocks+	0.039	0.026	**0.009**
segment+	0.037	0.017	**0.009**
sick+	0.022	**0.008**	0.011

Table 2 presents the experimental summary of accuracy for proposed approaches and compared approaches. In most of the datasets CFIS have improved on other algorithms in terms of accuracy. The accuracy values of german credit, mfeat and nursery data chunks combined performance is better for hoeffding tree as the internal constitution of dataset lead to good performance of hoeffding tree.

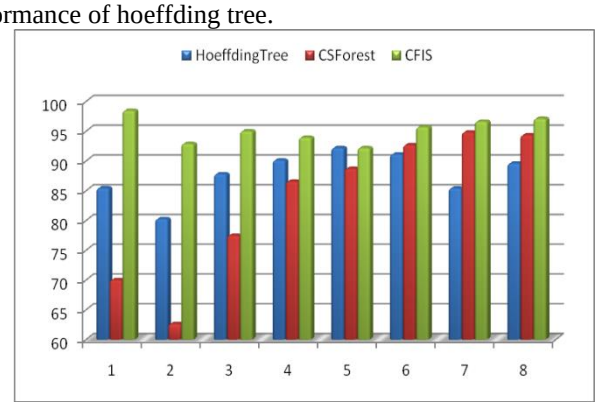

Fig. 3Trends in Accuracy ofHoeffdingTree, CS Forest versus CFIS on data stream

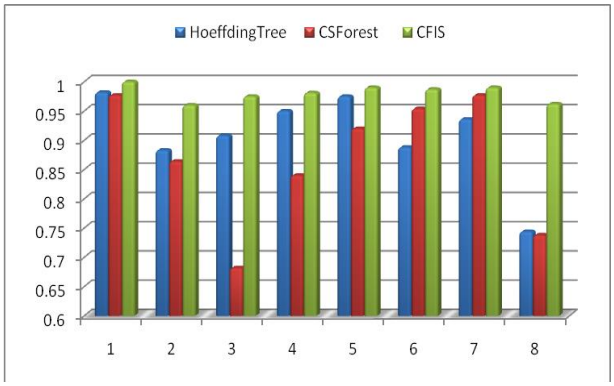

Fig. 4Trends in AUC ofHoeffdingTree, CS Forest versus CFIS on data stream

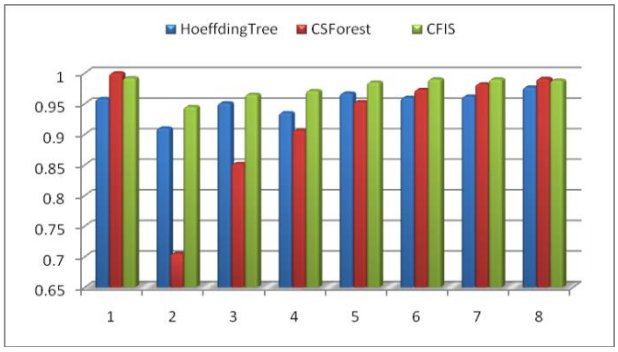

Fig. 5Trends in TP Rate ofHoeffdingTree, CS Forest versus CFIS on data stream

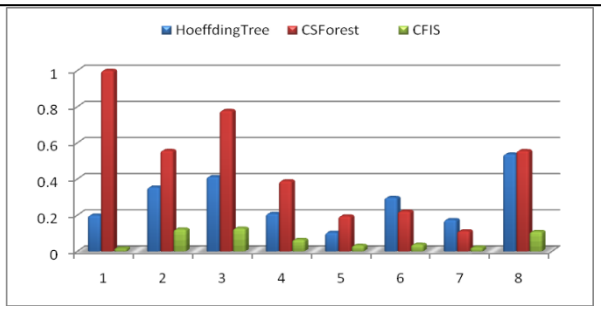

Fig. 6Trends in FP Rate ofHoeffdingTree, CS Forest versus CFIS on data stream

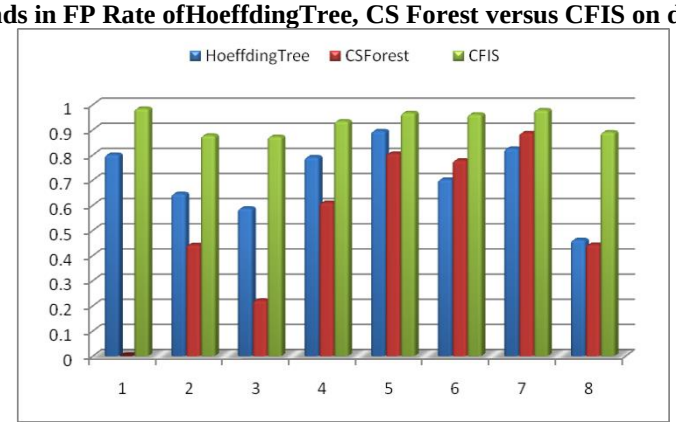

Fig. 7Trends in TN Rate ofHoeffdingTree, CS Forest versus CFIS on data stream

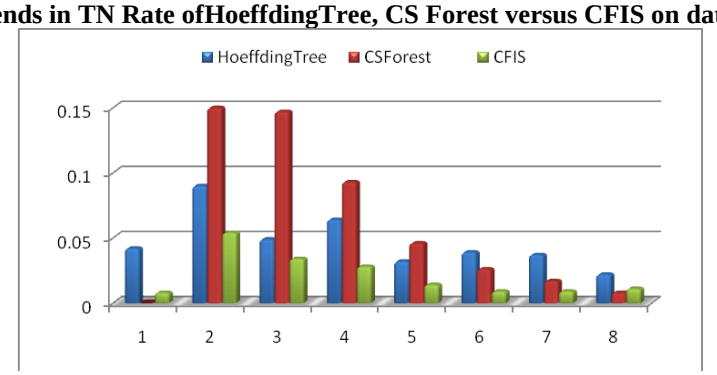

Fig. 8Trends in FN Rate ofHoeffdingTree, CS Forest versus CFIS on data stream

The summary details of AUC, TP Rate, FP Rate, FN rate and TN Rate are presented in the Table 2, 3, 4, 5, 6 and 7. The experimental details of AUC of CFIS versus Hoeffding Tree and CSForest on all the data sets are presented in Table 3. Figure 3 to 8 presents the graphical representation of accuracy, AUC, TP Rate, FP Rate, FN rate and TN Rate metric for the compared and proposed algorithm and the results are encouraging for the proposed CFIS algorithm.

6. Conclusion

Data stream mining in the presence of concept drift and class imbalance is one of the emerging challenges. In this paper, a Cuckoo Feature and Instance Selection (CFIS) algorithm is proposed for handling the presence of class imbalance and concept drift. The evolutionary cuckoo search algorithm is compared with hoeffding tree and CS forest approaches and the results indicate the improvement of the proposed approach.

References:

1. Gama J (2010) Knowledge discovery from data streams. Chapman & Hall/CRC, London
2. Aggarwal CC (2007) An Introduction to Data Streams. In: Aggarwal CC (ed) Data streams. Advances in database systems, vol 31. Springer, Boston.
3. Xin-She Yang, Suash Deb," Cuckoo Search via L´evy Flights", Conference: Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on
4. RaminRajabioun," Cuckoo Optimization Algorithm", Applied Soft Computing 11(8):5508-5518, DOI: 10.1016/j.asoc.2011.05.008
5. M. Mareli, B. Twala," An adaptive Cuckoo search algorithm for optimisation", Applied Computing and Informatics 14 (2018) 107–115.
6. Amir Hossein Gandomi, Xin-She Yang, Amir Hossein Alavi," Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems", Engineering with Computers (2013) 29:17–35, DOI 10.1007/s00366-011-0241-y.

7. Rohit Salgotraa ,∗, Urvinder Singh a , SriparnaSaha," New cuckoo search algorithms with enhanced exploration and exploitation properties", Expert Systems With Applications 95 (2018) 384–420.
8. Venkata Vijaya Geeta. Pentapalli, Ravi Kiran Varma P," Cuckoo Search Optimization and its Applications: A Review", International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 5, Issue 11, November 2016.
9. Yang, X., Deb, S. & Mishra, S.K. Multi-species Cuckoo Search Algorithm for Global Optimization. *CognComput* 10, 1085–1095 (2018). https://doi.org/10.1007/s12559-018-9579-4
10. Jaddi NS, Abdullah S, Abdul Malek M (2017) Master-Leader-Slave Cuckoo Search with Parameter Control for ANN Optimization and Its Real-World Application to Water Quality Prediction. PLoS ONE 12(1): e0170372. doi:10.1371/journal. pone.0170372
11. Yung C Shih," A cuckoo search algorithm: Effects of coevolution and application in the development of distributed layouts", Journal of Algorithms & Computational Technology Volume 13: 1–19.
12. Bustamam, V. Y. Nurazmi, and D. Lestari,:" Applications of Cuckoo Search Optimization Algorithm for Analyzing Protein-Protein Interaction Through Markov Clustering on HIV", *Proceedings of the 3rd International Symposium on Current Progress in Mathematics and Sciences 2017 (ISCPMS2017)* AIP Conf. Proc. 2023, 020232-1–020232-6; https://doi.org/10.1063/1.5064229.
13. Xin-She Yang, Suash Deb," Engineering optimisation by cuckoo search", Int. J. Mathematical Modelling and Numerical Optimisation, Vol. 1, No. 4, 2010.
14. Celso A. G. Santos, Paula K. M. M. Freire and Sudhanshu K. Mishra,:" CUCKOO SEARCH VIA LÉVY FLIGHTS FOR OPTIMIZATION OF A PHYSICALLY-BASED RUNOFF-EROSION MODEL", Journal of Urban and Environmental Engineering, v.6, n.2, p.123-131.
15. Zhigang Lian, Lihua Lu, Yangquan Chen. A New Cuckoo Search. 2nd International Conference on Intelligence Science (ICIS), Oct 2017, Shanghai, China. pp.75-83, 10.1007/978-3-319-68121-4_8. hal-01820910
16. HamiltonA. Asuncion D. Newman. (2007). *UCI Repository of Machine Learning Database* (School of Information and Computer Science), Irvine, CA: Univ. of California [Online]. Available: http://www.ics.uci.edu/ ~mlearn/MLRepository.html
17. Geoff Hulten, Laurie Spencer, Pedro Domingos (2001). Mining time-changing data streams. In: ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 97-106, 2001.
18. Michael J. Siers& Md Zahidul Islam (2015). Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem. Information Systems. 51:62-71