

OPTIMIZED ENSEMBLED CONCEPT DRIFT DETECTION

João Pedro Mauad Nogueira

Programa de Pós-Graduação em Engenharia de Produção e Sistemas – Pontifícia Universidade
Católica de Paraná

R. Imac. Conceição, 1155 - Prado Velho, Curitiba, PR, CEP 80.215-901, Brazil

`joao.mauad@pucpr.edu.br`

Gilberto Reynoso Meza

Programa de Pós-Graduação em Engenharia de Produção e Sistemas – Pontifícia Universidade
Católica de Paraná

R. Imac. Conceição, 1155 - Prado Velho, Curitiba, PR, CEP 80.215-901, Brazil

`g.reynosomeza@pucpr.br`

ABSTRACT

Machine learning is a branch of artificial intelligence that is crucial for predicting trends in various industries. In most real-world scenarios their underlying patterns constantly change over time, resulting in the degradation of model's performances. This is labeled as concept drift. The following article introduces an approach to detect drifts, named Optimized Ensembled Concept Drift Detection (OECDD). For the testing stage there was the usage of both artificially made and real datasets containing sudden, gradual and incremental drifts. By evaluating the proposed approach against the Naive Bayes classifier and a variant containing the classic drift detection detection method (DDM), it was possible to observe that OECDD achieved the highest accuracy, while presenting a significantly longer processing time - with RAM consumption rates being similar across all alternatives. Concluding, the developed method is suitable for more complex and less time demanding scenarios. Future work will explore optimization improvements and real-world applications.

KEYWORDS. Concept Drift Detection. NSGA-II. Weighted Majority Voting Ensemble.

1. Introduction

The usage of machine learning is a determining factor in the prediction of current and future tendencies, aiding in industry applications such as maintenance, logistics and product quality. Real life scenarios constantly changes over time. The stipulation of stock market values, trends in product consumption, viewership of content present in different platforms and so on and so forth. Considering this, the performance of machine learning models may decrease after an instance or sequence of significantly impact-full events. This describes the phenomenon known as concept drift. Although concept drift applications are vast, there remains potential for further improvement in this area. Other approaches, that are outside the realm of concept drift, may be implemented in order to achieve better results. For example, ensemble combinations and genetic algorithms. The following article aims to present a resulting approach named Optimized Ensembled Concept Drift Detection, testing its classification capacity and results.

2. Previous work

In "EOCD: An ensemble optimization approach for concept drift applications", [Feitosa Neto and Canuto, 2021] developed an approach of concept drift detection that utilizes genetic algorithm as an optimization tool, explicit and implicit mechanisms of adaptation in the ensembles. This article was the main inspiration for the Optimized Ensembled Concept Drift Detection, more specifically in the combination of genetic algorithm and ensemble learning to improve concept drift detection. The articles of [Lu et al., 2019], [Verma et al., 2021] and [Dogan and Birant, 2019] detail the mechanisms and overall knowledge of, respectively, concept drift, genetic algorithm (NSGA-II) and the used ensemble configuration (Weighted Majority Voting Ensemble) - them being reviews or the source main article for each component. Other applications that combine at least two of the appointed mechanisms can be visualized in the works of [Singh et al., 2024], [L. Lobo et al., 2018] and [Noori et al., 2023]. Said authors denoting content, respectively, content about network traffic anomaly detection, evolutionary diversity generation method for concept drift adaptation and an intrusion detect system. The articles of [Brzezinski and Stefanowski, 2014], [Bifet et al., 2010a], [Bifet et al., 2010b] present the real data sets and framework that produce synthetic data to be used in testing environments. Although the primary focus of the referenced articles is on developing new techniques or improving existing ones, the main contribution of this paper lies in structuring and testing a broader concept. For instance, in the works of [Feitosa Neto and Canuto, 2021], notable results are achieved in, for instance, both drift detection and drift adaptation, albeit with a dense and complex implementation. In contrast, this article presents a simpler configuration by associating a classic optimization technique with well established machine learning models. As such, this framework provides an easier entry point to more advanced concept drift algorithms, facilitating the visualization and understanding of the potential - in terms of gain - of combining optimization algorithms with concept drift.

3. Concept Drift Detection

According to [Lu et al., 2019], the concept drift phenomenon is described as the change of probabilities of a certain variable, given a certain period of time (equation 1), the main branches of this area being the detection, understanding and adaptation of changing scenarios. Considering applications of machine learning, the performance of models may drop considerably since the used data for training presented different tendencies and patterns before the concept drift. Real-world systems present relevant complexity and dynamicity. With that in mind, some examples of the implementation of concept drift detection are the usage of product usage such as mobile phone usage pattern, initiating with "audio calls", to "camera" and currently "mobile internet", intrusion detection systems like in [Noori et al., 2023] or in more simple events, for instance, the changing of seasons where is the variation of temperature, humidity and vegetation. Drift monitoring may lead

to the triggering and substitution of inadequate models with recently trained ones, maintaining the functioning of a predictive system.

$$P_{t+1}(X, C) \neq P_t(X, C) \quad (1)$$

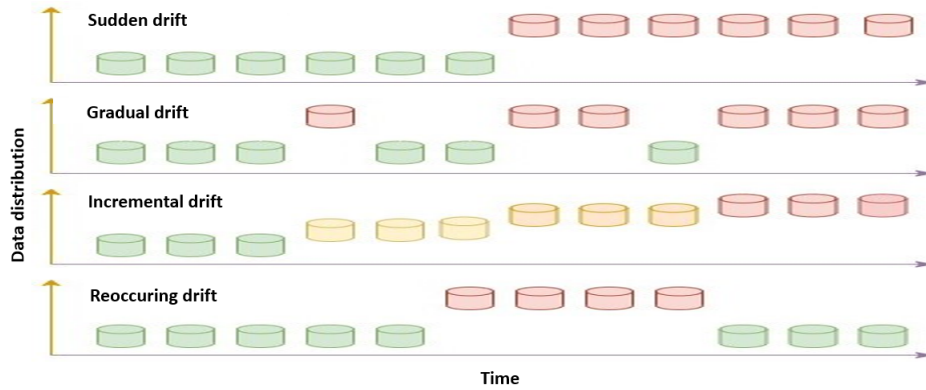


Figure 1: Types of drift.

Considering the transition between two data concepts, there are four main drift types: Sudden, Gradual, Incremental and Reoccurring. In the sudden drift the new concept abruptly changes the previous one, the underlying distribution of both are clearly distinct being the less complex concept drift to detect. As the label states, the gradual drift consists on the progressive change between two data concepts. In it, the frequency of the initial concept decreases while the frequency of the sequenced concept increases. In the incremental drift there is a continuous transition. One way to visualize it is the accuracy of a decreasing performance wise model in the given scenario, it presents the aspect of an exponential curve. As such, the incremental drift could be considered a subtype of the gradual drift. The remaining reoccurring drift is described as the interchange of two or more concepts in a seasonal manner - i.e with the same characteristic after a certain time (Figure 1). [Lu et al., 2019] states that in order to detect said concept drifts, there were developed detection methods. The earliest developed one was the Drift Detection Method (DDM). It identifies whether the overall online error rate within a predetermined period had a significant increase or not. If the monitored changing level hits a warning threshold, it instantiates a new machine learning model that is used in incremental learning, while the predictions of its predecessor are still being used. If the change reaches the drift threshold, the old model is replaced for further predictions.

4. Non-Dominated Sorting Genetic Algorithm

One of the most popular algorithms to solve multi-objective optimization problems is the Non-Dominated Sorting Genetic Algorithm also known as NSGA-II. According to [Verma et al., 2021] the sequence executed by the algorithm consists, initially, on the generation of an initial population P_t with N size. After that, there is a mutation of the produced population, receiving a new label Q_t . Then, both of them are combined into R_t , with their members being the ranking denoted by their non-domination levels. The subsequent step involves choosing N members from R_t to make a new population, P_{t+1} . If the size of the first front is greater than or equal to N , only N members from the least crowded region of this front are selected to make part of P_{t+1} . Conversely, if the first front's size is less than N , its members are directly transferred to the next generation and new additional members are selected from the least crowded region of the second front, in order to compose P_{t+1} . Should it still fall short of N , the same method is applied to the following fronts until

the population reaches the required size of N . This procedure is repeated for creating populations P_{t+2} , P_{t+3} , P_{t+4} , and so on and so forth, for subsequent generations, until the stopping criteria are met.

5. Weighted Majority Voting

Weighted majority voting is an approach for classification of ensemble learning. Within it, there is a combination of multiple machine learning model classifiers in order to predict the value of a certain instance. In such method, each classifier inputs their prediction, giving a weighted vote for one of the labels to said instance - said weight is determined by the precision or reliableness of the machine learning model. The predicted class for the ensemble will be the one that presents the most votes. Such approach brings more solid results in a predicting task, considering that different models may be more optimal given different scenarios. In other words, each model contributes with their strengths in order to complement the others weaknesses. It then provides results with higher accuracies for more complex scenarios. As stated by [Dogan and Birant, 2019], there are three phases in the synthesis of a Weighted Majority Voting Ensemble (WMVE): training of models, determination of weights and the combination of all individual classifiers outputs. Taking into account w_t as the equivalent weight of a said classifier and $d_{t,j}$ as its decision or vote made. The equation that describes the elected label being (2).

$$\hat{y} = \max_{1 \leq j \leq k} \sum_{t=1}^N w_t \cdot d_{t,j} \quad (2)$$

6. Optimized Ensembled Concept Drift Detection approach

As it was already stated, the discussed approach was already applied by [Feitosa Neto and Canuto, 2021] in the Ensemble Optimization for Concept Drift. The following is a general structure for future usages in concept drift detection realm and its replication, presenting some differences like the usage of NSGA-II, grid search and randomized search. With that in mind, the Optimized Ensembled Concept Drift Detection approach is described in Figure 2. It essentially consists on the usage of NSGA-II with machine learning models to compose the most optimal weighted majority voting ensemble in order to achieve better prediction results.

Said approach utilizes batches with instances of a data stream. There are two named batches, the execution batch and training batch. The first one denotes the input that must be predicted by the configured optimized ensemble, while the latter is used to synthesize new models that will compose a new ensemble. In the first step, there is a sampling of the first batch that will be used in the training of twelve different models: naive bayes, stochastic gradient descend, decision tree, multilayer perceptron classifiers and all of the previous alternatives using either the grid search algorithm or the randomized search algorithm to perform an initial optimization. After that, the selected data is inputted in the training of all possible models. Continuing, there is the synthesis of all the possible combinations for each model in an array containing four slots. With all the newly formed ensembles, the main optimization process is executed by applying NSGA-II to the multi-objective optimization problem (MOOP) that represents this scenario. Equation (3) formulates the optimization problem where x is the accuracy for the next execution batch, y the mean execution time and z complexity models, a scale of scores that gives the complexity level associated with the classifiers, with naive bayes presenting the lowest complexity and the multilayer perceptron upholding the highest complexity of all the selected classifiers. As for the variables utilized in the genetic algorithm, the population size consists in all the possible ensemble combinations. Float random sampling was used as the sampling method, applied to the tournament selection of the binary type and survival. After the execution of NSGA-II, the best optimized ensemble is determined. Then

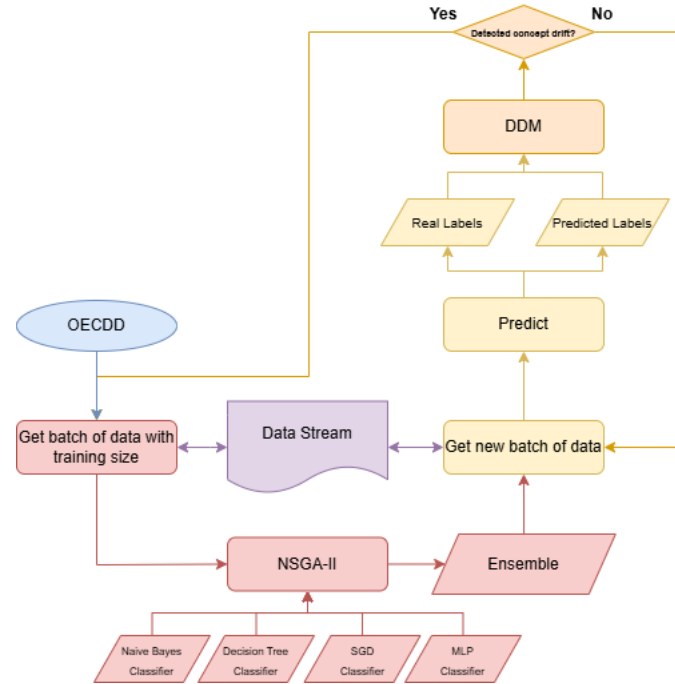


Figure 2: OECDD framework flow chart.

there is the retrieval and prediction of a new execution batch. Both input and output of the ensemble are inserted in the drift detection method, after which the same batch is used to make a partial fit on the ensemble. If there is a detection of drift, there must be the instance of a new training batch and, then, all of the previous steps must be executed providing the new best ensemble configuration. This cycle is repeated for all the existence of the data stream. It is important to explicit that the best usage of NSGA-II consists on the creation of a pareto front, but for the current article it will be only used to optimize the MOOP stated in equation (3).

$$\begin{aligned} \text{Minimize } X &= (f_1(x, y, z)) \\ f_1(x, y, z) &= x^{\frac{1}{8}} + y + z \\ 0.0 &\leq x \leq 1.0 \\ 4 &\leq y \leq 40 \\ 0.0 &\leq z \leq 25.0 \end{aligned} \quad (3)$$

6.1. Designed test

For the design of the experiment test to the OECDD approach, there will be the comparison of a naive bayes classifier, naive bayes classifier with drift detection method and a OECDD ensemble. They will all be tested in both synthetic and real data sets. For the synthetic data sets, by using the MOA framework [Bifet et al., 2010b], there was the creation of data with sudden, gradual and incremental drift types. Considering the sudden drift test, five data streams were generated, each with 100,000 samples, ensuring that the transition window between concepts is zero, meaning the drift is instantaneous, the drift change point being set at 50,000 position. In the case of the gradual drift, another five data streams, each with 100,000 samples, were generated. However, to simulate a gradual group concept drift, it was established a transition drift window of 20,000

samples, starting from sample number 40,000 to sample 60,000. The SEA generator was used to obtain these data. Finally, to simulate multiple data streams with incremental group drift, a hyper-plane generator was used. Ten dimensions were established, with a change magnitude of 0.001 per sample. The probability of a reverse direction change was set to 10. As to the real data sets presented in the works of [Brzezinski and Stefanowski, 2014] and [Bifet et al., 2010a], Electricity was collected from the Australian New South Wales Electricity Market, containing 45,312 observations to denote the change in price relative to the moving average of the last 24 hours. SPAM represents the results from the text mining process within an online news dissemination system. This work aimed to develop an incremental filter to classify emails as either spam or not. Furthermore, based on this classification, the intention is to assess whether an email is relevant for sharing among users. USENET₁, similar to an email dataset, contains the same concept changes but with only 100 attributes. Similarly, USENET₂ also resembles an email dataset with 100 attributes. However, both the first and last drifts are identical, as well as the second and fourth ones. This means there are two abrupt and recurring drifts in the fourth and final cases, respectively. For all of the three approaches, there was the measurement of the accuracy, using the prequential evaluation method, mean execution time and RAM consumption. The produced results were established of the mean of eleven executions. All of the alternatives use partial fit of the execution batch after they make a prediction.

Table 1: Results - Accuracy (%).

Data set	Naive Bayes	Naive Bayes with Drift Detection Method	OECD
elecNormNew	72.81	75.25	71.93
spam	89.22	90.96	91.16
usenet ₁	58.40	58.40	59.17
usenet ₂	71.10	71.12	68.38
SEA12 _{gradual}	85.92	85.92	87.52
SEA21 _{gradual}	87.44	87.47	87.77
SEA23 _{gradual}	85.36	85.41	87.46
SEA31 _{gradual}	86.22	86.23	87.19
SEA32 _{gradual}	83.83	83.85	86.44
hyper ₁	84.00	84.09	86.93
hyper ₂	85.72	85.81	86.63
hyper ₃	82.96	83.07	85.87
hyper ₄	85.34	85.37	86.78
hyper ₅	86.80	86.81	85.95
SEA12 _{sudden}	85.91	85.93	87.92
SEA21 _{sudden}	87.47	87.48	87.77
SEA23 _{sudden}	85.44	85.46	87.68
SEA31 _{sudden}	86.18	86.22	87.80
SEA32 _{sudden}	83.76	83.93	87.70

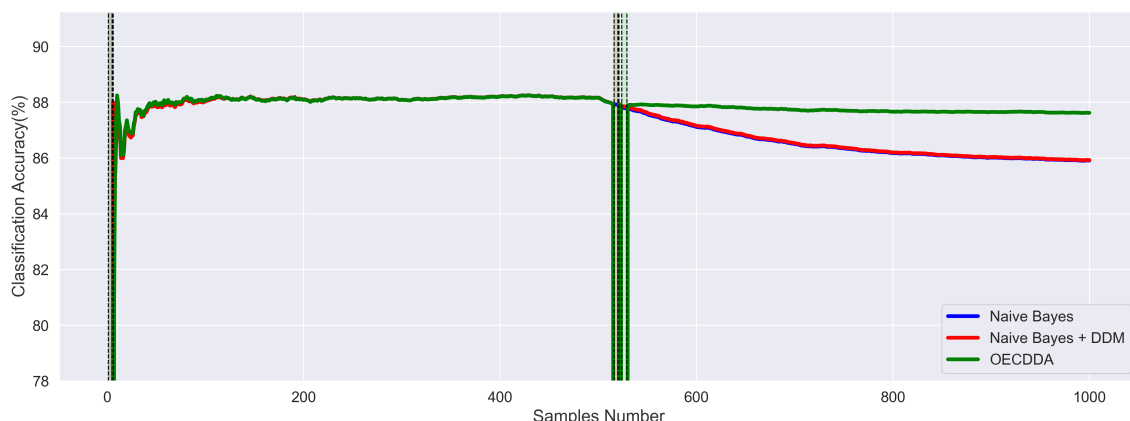


Figure 3: Accuracy plot of data with sudden drift.

Table 2: Results - RAM-Hours (GB/h).

Data set	Naive Bayes	Naive Bayes with Drift Detection Method	OECDD
elecNormNew	0.10628307	0.10259056	0.09400857
spam	0.10318961	0.10305554	0.09329027
usenet ₁	0.10315951	0.10252413	0.09350761
usenet ₂	0.10309125	0.10246194	0.0935735
SEA12 _{gradual}	0.10244394	0.10242659	0.09406454
SEA21 _{gradual}	0.10370328	0.10289568	0.0943915
SEA23 _{gradual}	0.10444134	0.10300769	0.09464951
SEA31 _{gradual}	0.10408284	0.10297368	0.09433547
SEA32 _{gradual}	0.10316281	0.10273066	0.09449328
hyper ₁	0.10306248	0.10271317	0.09445784
hyper ₂	0.10296246	0.10254842	0.09435996
hyper ₃	0.10294432	0.10248178	0.09431667
hyper ₄	0.10286972	0.10247085	0.09535468
hyper ₅	0.10278091	0.1025894	0.09612873
SEA12 _{sudden}	0.10264983	0.10258772	0.09578953
SEA21 _{sudden}	0.10251756	0.10271537	0.09570795
SEA23 _{sudden}	0.10242088	0.10265064	0.09568606
SEA31 _{sudden}	0.10237911	0.10258316	0.09582998
SEA32 _{sudden}	0.10246043	0.10248663	0.09586537

6.2. Results

By applying said approach in the established data, a few observation can be pointed out. In regards to the general mean accuracy, OECDD presents the highest results for sixteen of the nineteen data sets. By analysing said output, the three exceptions where in the elecNormNew, usenet₂ and hyper₅ data sets. Although, this could point out to the possibility of said method presenting better results in artificial data, in half of the real data sets OECDD had the best performance accuracy wise (Table 1). Continuing, as expected, OECDD, by far, presents the slowest mean execution time with

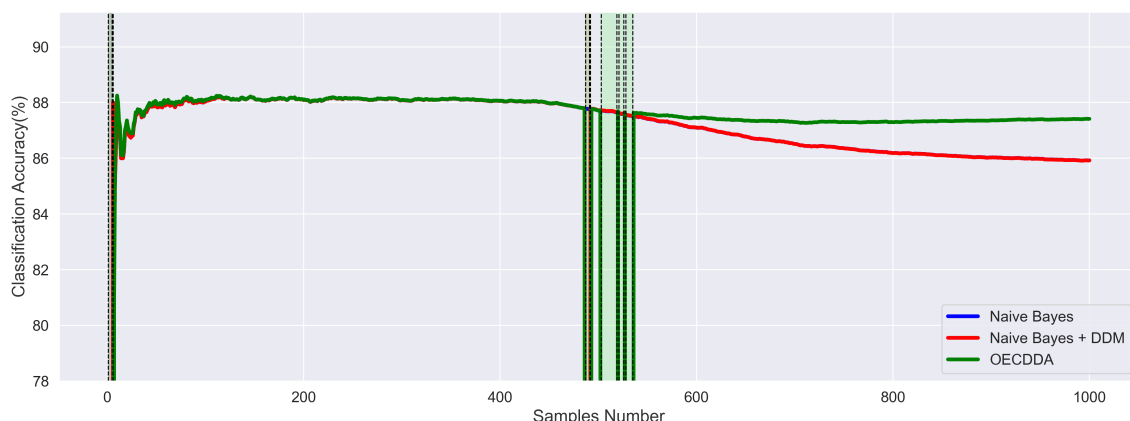


Figure 4: Accuracy plot of data with gradual drift.

the average

Table 3: Results - Mean execution time per batch (s).

Data set	Naive Bayes	Naive Bayes with Drift Detection Method	OECDD
elecNormNew	0.1777	0.0678	343.5170
spam	0.0440	0.0300	16.1618
usenet ₁	0.0033	0.0014	5.1007
usenet ₂	0.0028	0.0014	4.9287
SEA12 _{gradual}	0.3172	0.08855	11.5172
SEA21 _{gradual}	0.0838	0.0904	7.4936
SEA23 _{gradual}	0.0898	0.0909	11.3528
SEA31 _{gradual}	0.1081	0.0871	11.4067
SEA32 _{gradual}	0.0766	0.0913	11.4233
hyper ₁	0.0852	0.0908	20.5138
hyper ₂	0.0781	0.0917	16.3448
hyper ₃	0.0810	0.0932	12.0086
hyper ₄	0.0782	0.0937	20.6887
hyper ₅	0.0795	0.0942	16.3740
SEA12 _{sudden}	0.0781	0.0871	7.5089
SEA21 _{sudden}	0.0782	0.0914	7.5382
SEA23 _{sudden}	0.0753	0.0871	7.5879
SEA31 _{sudden}	0.0768	0.0889	11.3257
SEA32 _{sudden}	0.0853	0.0928	7.5115

time of the framework being up to ten to a hundred times the mean execution times of classic naive bayes machine learning model and naive bayes added with drift detection method (Table 2). Said lethargy derives from the time spent to generate the most proximate ensemble combination via genetic algorithm. Finally, regarding the mean ram consumption rate in RAM-Hours (Table 3), all three approaches have close results with the framework exhibiting mildly lower values. With that

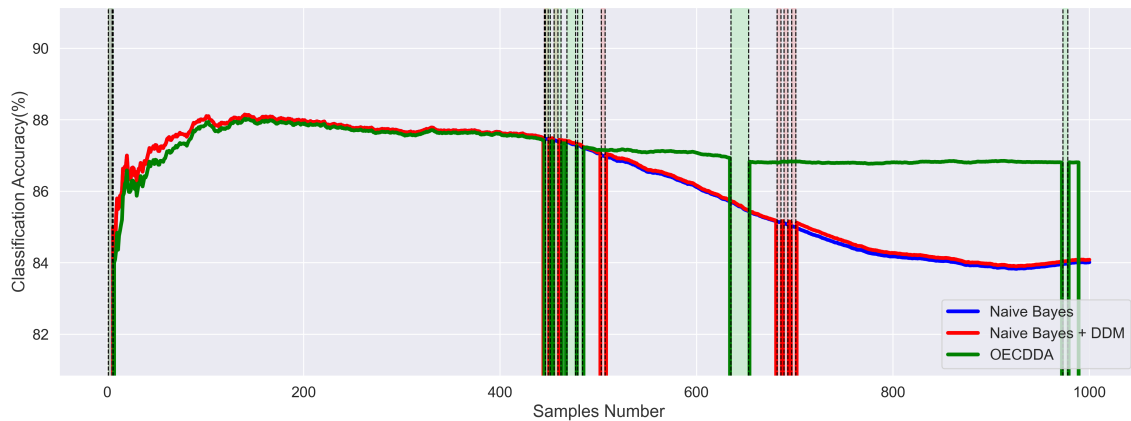


Figure 5: Accuracy plot of data with incremental drift.

regard, due to the longer execution time of OECDD, the total consumption of RAM is also up to ten to a hundred times more than the other alternatives. Considering the accuracy plots for all the data sets, the scale of which having one instance representing the mean of one hundred instances, for the sudden drift data it was possible to observe that de detection was close to the 500 instance i.e 50.000 (Figure 3). OECDD is more capable for detecting gradual drifts, described in Figure 4, maintaining a higher and constant accuracy after instance 550, or 55.000. With the most discrepant results of OECDD compared to the other alternatives being in the incremental drift data, presented in Figure 5. One other aspect is that the naive bayes approach with or without drift detection method did share close results.

7. Conclusions

In the stated results, it was possible to observe the overall better performance, accuracy wise, of the Optimized Ensembled Concept Drift Detection approach. This is due to the execution of two optimization processes, uplifting the models performances. As such, the processing consumption and total execution time is greater when compared to both the approaches that utilize naive bayes. Considering this, the main focus of the developed structure may be more interesting for more complex and less speed demanding scenarios. For example support in decision making and information oriented application.

8. Future works

For future improvements, testing wise, there may be the change of measurement for time and computational resources consumption - perhaps the total time per drift type or total time per drift. Also new variables in the NSGA-II optimization problem could better represent the priorities of a certain context. With that in mind, as stated in the OECDD section NSGA-II best use is for making pareto fronts and, as such, there lies the need to implement this application based on the objectives described also in the same section. Another important aspect is the comparison of the OECDD approach not only with using other machine learning models, but also with state-of-the-art concept drift detectors currently found in the literature. At last, there is a necessity to test and verify the usage of OECDD approach in real life cases.

References

- Bifet, A., Holmes, G., and Pfahringer, B. (2010a). Leveraging bagging for evolving data streams. In *Machine Learning and Knowledge Discovery in Databases*, p. 135–150.
- Bifet, A., Holmes, G., Kirkby, R., and Pfahringer, B. (2010b). Moa: massive online analysis. *Journal of Machine Learning Research*, 11:1601–1604.
- Brzezinski, D. and Stefanowski, J. (2014). Combining block-based and online methods in learning ensembles from concept drifting data streams. *Information Sciences*, 265:50–67.
- Dogan, A. and Birant, D. (2019). A weighted majority voting ensemble approach for classification. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*, p. 1–6.
- Feitosa Neto, A. and Canuto, A. M. (2021). Eocd: An ensemble optimization approach for concept drift applications. *Information Sciences*, 561:81–100.
- L. Lobo, J., Del Ser, J., Bilbao, M. N., Perfecto, C., and Salcedo-Sanz, S. (2018). Dred: An evolutionary diversity generation method for concept drift adaptation in online learning environments. *Applied Soft Computing*, 68:693–709.
- Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2019). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31:2346–2363.
- Noori, M. S., Sahbudin, R. K. Z., Sali, A., and Hashim, F. (2023). Feature drift aware for intrusion detection system using developed variable length particle swarm optimization in data stream. *IEEE Access*, 11:128596–128617.
- Singh, R., Srivastava, N., and Kumar, A. (2024). Anomaly detection in network traffic scenarios by resampling and majority voting with concept drift: A hybrid approach. In *2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, p. 1–7.
- Verma, S., Pant, M., and Snasel, V. (2021). A comprehensive review on nsga-ii for multi-objective combinatorial optimization problems. *IEEE Access*, 9:57757–57791.