Assessing Batch and Online Learning for Delivery in Full and On Time Predictions

Adriano Alves de Lima
Department of Informatics
Federal University of Paraná
Curitiba, Brazil
adrianolima@ufpr.br

Márcio Venâncio Batista
Department of Informatics
Federal University of Paraná
Curitiba, Brazil
marcio.venancio@ufpr.br

Jean Paul Barddal
Graduate Program in Informatics (PPGIa)
Pontifícia Universidade Católica do Paraná
Curitiba, Brazil
jean.barddal@ppgia.pucpr.br

Danilo Sipoli Sanches
Department of Computer Science
Federal University of Technology of Paraná
Cornélio Procópio, Brazil
danilosanches@utfpr.edu.br

Luiz Eduardo Soares de Oliveira
Department of Informatics
Federal University of Paraná
Curitiba, Brazil
luiz.oliveira@ufpr.br

***Abstract*—Improving results by optimizing process execution is one objective of major companies. For these corporations, the main point for achieving better results is the good maintenance of supply chain management. The most important supply chain metric is Delivery in Full and On Time (DIFOT). DIFOT measures how well a supply chain delivers value to the customer. In this work, we bring forward an analysis of DIFOT prediction from large Brazilian food company. More specifically, we compare a batch and online learning algorithm for DIFOT prediction and depict why the latter is suitable for this problem. Furthermore, we report a feature drift analysis to identify whether there are considerable shifts along with the dataset timespan. As a byproduct of this research, we make the dataset used in this analysis publicly available for future research in DIFOT prediction.**

***Index Terms*—Dataset, Data Streams, Feature Drift, DIFOT**

I. Introduction and Motivation

The role of the manufacturing industry is to create wealth by adding value and selling products. Common to all manufacturing companies is the need to control material flow from suppliers through the value-adding processes and distribution channels. The supply chain is the connected series of activities that concerning planning, coordinating, controlling material, parts, and finished goods from suppliers to the customer [1]. It is worried about two distinct flows over the organization: material and information. The supply chain's scope begins with the source of the supply and ends at the point of consumption. It extends much further than merely a concern with the physical movement of material and is just as much concerned with management, purchasing, facility planning, customer service, and information flow as with transport and physical distribution. Therefore, the supply chain is the series of steps and processes by which value is added to a product and through which it is delivered to an end customer [2], [3].

A significant supply chain Key Performance Indicator (KPI) metric is Delivery in Full and On Time (DIFOT). This metric is the ultimate performance measure of a supply chain. DIFOT directly measures how well a supply chain is fulfilling the delivery of value to the customer. It would be inconceivable for business not to measure profit or cash flow. It should be equally inconceivable for manufacturing or distribution businesses not to manage DIFOT. DIFOT, in its simplest form, is the ratio between the number of orders that were delivered on time, with the ordered items supplied in the quantity required on the day that the customer required them, and the total number of orders shipped [3].

Given the relevance of the DIFOT KPI in a supply chain, this work contributes with a comparative study of machine learning techniques batch and online supervised classification for DIFOT predictions. To the best of our knowledge, there are no studies have dealt with DIFOT prediction using machine learning techniques, specifically using data stream mining, only some barely studies that waste minimization being studied and taxonomy in Food Supply Chain (FSC) [4], [5] and risk prediction [6]. Thus, we propose the use of traditional batch and online learning approaches to predict the occurrence of DIFOT for a sales order using a priori data, i.e., data that is obtained during the creation of the sales order. Consequently, it is possible to know if the sales order will achieve the DIFOT, thus allowing the manager to mitigate weakness that may cause the failure of that sales order, which ultimately contributes to the DIFOT final company score.

The dataset of this paper was obtained from food company historical data. The data were collected from the period of 2018-01 to 2019-08. Since it is aim to predict the occurrence of DIFOT at the time of registering a new sales order, the problem becomes difficult to solve because the features used are all obtained a priori. Thus, at the moment when a sales order is created, some information is still undefined, such as the carrier that will deliver, or if it is possible to optimize the production to meet the demand, or even if it will be possible to deliver the entire quantity requested. In this paper we investigate how each feature, such as these mentioned, can impact in different ways the occurrence of DIFOT. Thus, we intend to answer two main research questions:

- Are data stream approaches better performing than batch ones in this real-world dataset?
- Considering the intrinsic time series in this dataset, are there feature drifts in its data?

The remainder of this paper is organized as follows. We provide a discussion about some papers that applied machine learning techniques to the supply chain and a brief review of the related studies in Section II. Next, we describe the characteristics of the dataset and details about the real-world database proposed in Section III. Next, we show our experimental protocol, indicating the batch and online methods used, validation process, metrics to evaluation, and hyper-parameter tuning approaches in Section IV. We present the main results obtained and comments on the discoveries on the dataset in Section V and conclusion in Section VI.

II. Literature Review

In this section, we present a research review of machine learning methods applied to supply chains (Section II-A) and data stream classification applications related to this work (Section II-B).

A. Supply Chain Management

The supply chain management occurs along with a network of upstream and downstream organizations, of both relationships and flows of material, information, and resources [2]. Supply chain professionals struggle to handle large amounts of structured and unstructured data. They are surveying new techniques to investigate how data are produced, captured, organized, and analyzed to give valuable insights into industries. Big data analytics is a popular approach for overcoming such problems [7].

Recently, some studies have indicated the benefits of using big data methods in logistics and supply chain management. Mishra and Singh [4] proposed a big data analytics approach for waste minimization in food supply chains. Shukla and Kiridena [8] introduced a fuzzy rough sets-based multi-agent model for configuring supply chains in dynamic environments. Along with these studies, there are many areas within supply chain management that could benefit from big data methods and technologies.

For instance, Baryannis et al. [6] proposed a risk prediction framework that uses data-driven Artificial Intelligence (AI) techniques and relies on the collaboration and interactivity between AI and supply chain experts. The authors defend a trade-off between the interpretability of the models and the best choice to measure the results, which sometimes is necessary a decrease on overall result over something like a black box. Zhang et al. [9] proposed an improved Random Forest to deal with online supply chain finance for risk evaluation and provide a scientific basis for risk assessments.

Angarita-Zapata et al. [5] designed a taxonomy for FSC that categorizes Computational Intelligence approaches and their relationship with FSC. According to them, our work is a distribution problem that could be solved with knowledge discovery and function approximation. Angarita-Zapata et al. highlight the challenges involving intersecting FSC and machine learning methods, mainly incremental learning, which has many gaps to be solved. Thus, in the next subsection we show the most relevant and useful methods for this task.

B. Data Streams Classification Domain Applications

Ramírez-Gallego et al. [10] survey the literature summarizing, categorizing, and analyzing the contributions on data preprocessing related to streaming data and some batch methods. The experiments were performed in synthetic data using Massive Online Analysis tool (MOA) and real datasets in the textual domain, and they conclude with claims to needed feature selection evolution in data streaming scenarios. Similarly, Barddal et al. [11] evaluated different credit score datasets using the most relevant methods for data stream classification. They also analyzed the feature importance in the credit scoring problem over time and reported comparable results to batch approaches in two of three datasets tested.

Won et al. [12] evaluated several feature selection techniques and provided empirical drift adaptation results via active learning. The method is composed of a drift adaptation system with subsequent active learning for performance recovery. The dataset used from different domain applications such as conflict detection, airline delay prediction, and text classification. In the last similar context, De Moraes and Gradvohl [13] presented a comparative study of six feature selection methods for text stream classification with the presence of feature drift. They also proposed the Online Feature Selection with Evolving Regularization algorithm, which uses regularization to dynamically correct the model complexity, thus reducing feature drift impacts.

Melidis et al. [14] proposed a method to tackle concept and feature drift using two components: sketch to maintain an updated feature space and an ensemble to average out potential drift on features. The domain application of their work is related to the textual classification of data streams. In the same domain, Shivakumaraswamy et al. [15] introduce a framework for active feature selection, designed to adapt the feature space over a stream of opinionated documents from the Amazon dataset, and this framework shows benefits compared to the default model. Also, in a similar context, Fahy and Yang [16] evaluate an algorithm-independent in textual and image streams for dealing with feature drift to be used with any of the density-based clustering.

Holmberg and Xiong [17] proposed a method to deal with feature drift in non-stationary data streams benchmark in MNIST public dataset. The method lies in deep reconstruction networks that are continuously updated with new instances. The networks are used to detect the changes and also to dynamically rank the importance of features selection. Sahmoud and Topcuoglu [18] proposed a framework to deal with the classification of data streams with feature drift. Their framework builds a dynamic multi-objective evolutionary algorithm called Dynamic Filter-Based Feature Selection with an Artificial Neural Network to classify the data streams by using only the features selected. Sahmoud and Topcuoglu evaluated their framework using four synthetic datasets, a common approach in the data stream classification domain due to scarcity of real-world databases essentially ephemeral.

Chamby-Diaz et al. [19] presented an algorithm called Dynamic Correlation-based Feature Selection (DCFS) that determines which features are the most important in a data stream. The DCFS uses an adaptive strategy based on a drift monitor to update the relevant features subsets dynamically. The researchers benchmark four real datasets and eight synthetic. In the same way, Duda et al. [20] proposed the Random Forest with Features Importance (RFFI), which uses the measure of feature importance as a drift detector. The RFFI implements solutions inspired by the Random Forest algorithm to the data stream scenarios. The authors evaluated their method in Random Tree Generator (RTG) and Electricity prediction.

Duarte and Gama [21] presented a study on feature ranking from data streams in online learning models. They proposed three new online feature ranking algorithms designed for Hoeffding-based methods. They also implemented three approaches in AMRules, a streaming regression method for learning rules. Ferone and Maratea [22] elaborated a variation of the QuickReduct algorithm suitable for processing data streams: it builds an evolving reduct that dynamically selects the features in the stream, removing the redundant ones and adding the newly relevant ones as soon as they become such.

Zhao and Koh [23] presented a framework to detect and describe feature drift in an unsupervised way using Wasserstein and Energy distance measures. To the best of our knowledge, no one study has been proposed to apply online approaches in Supply Chain domain application. Thus, we show the most relevant known techniques to benchmark in the proposed DIFOT dataset.

III. A NEW DIFOT DATASET

The dataset brought forward in this paper[1] was obtained from a Brazilian multinational food company's historical data. The data are dated from the period of January 2018 to August 2019.

Date and time features have given rise to new features such as day, month, week, and weekday, similar to [6]. At the end of this preprocessing phase, the dataset contained a total of 54 features. Among all the features, some are day, week, and month of order creation date; day, week, and month of preparation date for delivery; distance between the distribution center and customer. Some features classify the quality of roads in the location of the distribution center and client. There are also weather characteristics for the preparation date and the possible delivery date. The categorical features, which are: sales-type, sales document type, sales organization, sales channel, customer segmentation code, and distribution channel, all were transformed using a simple encoding strategy. This assigns an integer value for each distinct feature category, that is, if a categorical feature has only three states, the encoding process will generate the values 0, 1 and 2. As a final step for feature preparation, all features were subject to a scaling standardization algorithm. The entire dataset comprises 1,198,059 instances distributed into two imbalanced classes: DIFOT (985,052) and Non-DIFOT (213,007). Figure 1 shows the distribution class monthly with its respective amount of instances and Table I details the features available in the dataset alongside their types and descriptions.

TABLE I
DESCRIPTION OF THE FEATURES AVAILABLE IN THE PROPOSED DATASET (I: INTEGER, C: CATEGORICAL, F: FLOAT, DC: DISTRIBUTION CENTER).

Features Numbers	Data Types	Description/Group
{1, 2, 3, 4}	I	Order shipping {day, month, week, weekday}
5	I	Days between order creation and shipment
6	C	Sale Type
7	C	Distribution channel
8	C	Sales document type
9	C	Sales organization
10	C	Sales channel
11	F	Order average gross weight
12	F	Distance between the DC and the customer
13	C	Customer segmentation code
{14, 15, 16, 17}	I	Order creation {day, month, week, weekday}
{18, 19, 20, 21}	F	Parts preparation {day, month, week, weekday}
{22, 23, 24, 25}	F	Historical score for the occurrence of Difot in the month the order was {created, created grouped by sales type, created grouped by distribution channel, created grouped by document type}
26	I	Quantity different items in the sales order
27	F	Historical average gross weight of materials sold in the month that the order was created
28	I	Historical average of orders created in the month that the order was created
{29, 30, 31, 32}	F	Quality score set as {GOOD, REGULAR, BAD, AWFUL} for highways in the client's state
33	F	Final score for highways in the client's state
{34, 35, 36, 37}	F	Quality score set as {GOOD, REGULAR, BAD, AWFUL} for roads in the DC state
38	F	Final score for highways in the DC state
39	F	Climatic altitude on the first day after the shipping date in the vicinity of the DC
{40, 41}	F	{Precipitation, Humidity} on the first day after the shipping date in the vicinity of the DC
42	F	Wind speed on the first day after the delivery date in the vicinity of the DC
{43, 44, 45, 46}	F	{Climatic altitude, Precipitation, Humidity, Wind speed} on the second day after the shipping date in the vicinity of the DC
{47, 48, 49, 50}	F	{Climatic altitude, Precipitation, Humidity, Wind speed} on the first day after the shipping date in the vicinity of the customer
{51, 52, 53, 54}	F	{Climatic altitude, Precipitation, Humidity, Wind speed} on the second day after the delivery date in the vicinity of the customer
55	I	Label [1=DIFOT, 2=NON-DIFOT]

[1] Available at: https://web.inf.ufpr.br/luizoliveira/difot-dataset/

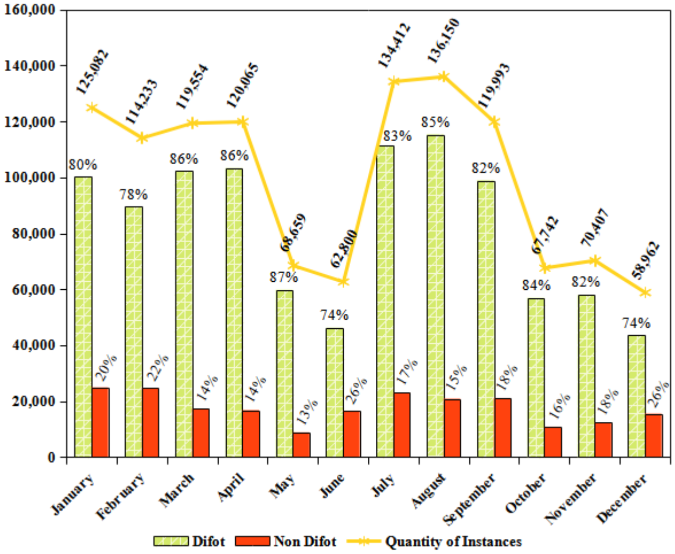

Fig. 1. Distribution of instances per month of creation order.

IV. Experimental Configuration

In this section, we present the batch and online learning approaches used to train the classifiers in DIFOT (Section IV-A). Section IV-B describes the metric used to evaluate the algorithms and to measure the feature importance. Section IV-C details the validation methods applied to benchmark the classifiers. Finally, in Section IV-D are the hyper-parameters used in the experimentation.

A. Learning algorithms

1) Batch methods: The classical algorithms from Sklearn[2] were experimented to allow us to compare with online learners: probabilistic Naive Bayes (NB), decision tree J48 classifier, Random Forest (RF) ensemble, and Balanced Bagging Classifier (BBC).

2) Very Fast Decision Tree: The *Hoeffding Tree* (HT) or VFDT [24] is a classifier based on tree decision, which shows better results than traditional methods (i.e. C4.5) in data streams. The use of the Hoeffding threshold along the node decision tree, without the entire information about the dataset, allows its application in online learning.

3) Hoeffding Adaptive Tree: This classifier is an extension of VFDT [25] which uses the concept drift detector *Adaptive Sliding Window* (ADWIN) [26]. Same as VFDT, the HAT may uses different base classifiers on its leaves decision trees built.

4) Leveraging Bagging: It is an OzaBag extension with some improvements [27]. In summary, the difference among the methods is on Poisson distribution, in which LevBag uses $\lambda = 6$, besides using ADWIN to concept drift detection. Therefore, the authors' experimental results showed better results compared to OzaBag.

5) Adaptive Random Forest: It is a modification of RF which deals with data streams [28]. The ARF uses decision trees on its base, and thus, it achieves satisfactory results in data streams and naturally deals with feature drifts.

6) Adaptive Random Forest with Resampling: It is a classifier designed to deal with imbalanced datasets [29]. ARFRE resamples instances based on the current class label distribution.

7) Cost-sensitive Adaptive Random Forest (CSARF): The CSARF is an ARF variant tailored to handle class imbalance in online learning tasks [30]. The main CSARF traits include: the assignment of weights to each internal tree; the addition of a sliding window to observe the classes distribution; a modification in the learning process to ensure that all trees train with minority class; and the assignment of cost sensitivity with two strategies (local and global).

8) Kappa Updated Ensemble: The KUE is an ensemble method that is a combination of online and block-based approaches that uses Kappa statistic for dynamic weighing and selection of base classifiers [31].

B. Evaluation metric and feature importance

To compare batch and online learning methods in DIFOT imbalanced dataset, we used F1 score or F-measure, which is the harmonic average between precision and recall with equal importance to them, see Equation (1). In our preliminary experiments, we benchmark some methods using accuracy, but we notice trend to majority class. Thus, and according to [6], the F1 score is suitable for imbalanced datasets, then we compute the macro, which sums the F1 score for each class and takes the mean for each of them.

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

In this work, we analyzed the feature importance of the features using Mean Decrease Impurity (MDI) [32]. MDI ranks features according to their position in RF. It is a weighted sum of values obtained during the split that accounts for the number of samples, see Equation (2). t_i is a tree inside the ensemble of trees $T = t_1, t_2, ..., t_E$, N_t is the number of examples observed in a split node b, N is the total number of samples in the entire tree, $J(b)$ is the goodness-of-fit computed during the split of a node b, and $\Omega(b)$ is a function that returns the feature $X_i \in X$ got in b.

$$\text{MDI}(x_i) = \frac{1}{|T|} \sum_{t_i}^{T} \sum_{b}^{t_i} \begin{cases} \frac{N_t}{N} \times J(b), & \text{if } \Omega(b) = x_i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

C. Validation methods

We adopted two types of validation: holdout and test-then-train. First, in holdout approach validation, we used different ratios of months for training and testing, including 40%-60%, 50%-50%, and 60%-40%. We experimented with some other splits like 70%-30% and 80%-20%, but the final results did not change so much. The main reason for testing different

[2] Available at: https://scikit-learn.org/stable/

proportions of the dataset here is to check the impact of more or less training data and whether the behavior of DIFOT changed over the months.

Second, the monthly test-then-train validation scheme considers that each month is used for training right after it is used for evaluation, except for the first month that is used solely for training. The main investigation here is to compare the results obtained using a classifier that is constantly updated with new instances against a model that was trained until a specific time.

D. Hyper-parameter tuning

The learning algorithms used possess different hyper-parameters that may impact the performance of the models to be created. The methods were tuned so that the F-measure metric during the test was optimized. The hyper-parameters tested during this process for both batch and online learning methods are given in Table II. The only classifier unreported was NB, as no hyper-parameters are available for tuning.

TABLE II
HYPER-PARAMETERS TWEAKED DURING THE TUNING PROCESS FOR BATCH AND ONLINE CLASSIFIERS.

Method	**Hyper-Parameter**	**Values**
J48	Criterion	{Gini, Entropy}
	Minimum samples split	{2, 50, 100, 200}
	Maximum features	{2, $\sqrt{n\_features}$}
	Class weight	{None, Balanced}
RF	Number of estimators	{10, 100}
	*Plus same others of J48	
BBC	Number of estimators	{10, 100}
	Sampler	{RandomUnder, RandomOver, SMOTE, TomekLinks, SMOTETomek}
HT/HAT	Numeric estimator	{Gaussian with 4, 7, 10, 12 bins}
	Grace period	{2, 50, 100, 200, 500}
	Criterion	{Infogain, Gini}
	Pre-prune	{True, False}
	Leaf prediction	{NBAdaptive, NB}
LEVBAG	Base learners	{HT, ARFHT}
	Ensemble size	{10, 50, 100, 150}
	Delta ADWIN	{0, 0.002}
	*Plus same others of HT/HAT	
ARF	Ensemble size	{10, 50, 100, 150}
	Features mode	{$\sqrt{n\_features}$, Percent, Int}
	Features per tree	{2, 3, 5, 10}
	Drift detection	{ADWIN with delta 0, 0.002}
	Warning detection	{ADWIN with delta 0, 0.002}
	*Plus same others of HT/HAT	
ARFRE	Tree learner	{HT, ARFHT}
	*Plus same others of ARF	
CSARF	Imbalance window	{1000, 10000}
	Threshold mode	{Local, Global}
	*Plus same others of ARF	
KUE	Learner	{HT, ARFHT}
	Member count	{10, 50, 100, 150}
	Chunk size	{1000, 10000}
	*Plus same others of HT/HAT	

V. RESULTS AND ANALYSIS

In this section, we analyze both batch and data stream learning algorithms in the context of the DIFOT dataset previously presented in Section III. The results obtained will be discussed separately and in 3 steps: first, we report and analyze the F-measure results obtained using batch learning algorithms when trained and tested using different proportions of the dataset using holdout validation. The goals with this analysis are (i) to verify whether more training data directly translates to higher F-measure value on test data, and (ii) to pick the best performing classifier for further comparisons against data stream approaches; Second, we compare different data stream mining algorithms in terms of F-measure. The goal is to identify whether continuously updating the predictive models using a monthly test-then-train validation process leads to higher prediction rates without damaging F-measure; finally, we analyze feature importance according to MDI.

A. Results Monthly

We initiate our analysis by batch methods monthly using under and oversampling according to Table III. We experimented with some others sampling techniques, but under and oversampling has been performed better in this imbalanced dataset. The best macro F-measure achieved was 0.75 in May using RF-Under, while the mean of all periods was 0.71 and a standard deviation of 0.04 (see Figure 2 to summarized results). However, the worst result using RF was 0.56 in March using oversampling, but the NB poorly achieved only 0.13 in the same time. Notice that all other batch methods showed poor performance, including BBC with a mean of less than 0.51.

TABLE III
MAIN RESULTS MONTHLY FOR BATCH METHODS.

Month	**NB-Under**	**J48-Under**	**RF-Under**	**BBC-Under**	**NB-Over**	**J48-Over**	**RF-Over**	**BBC-Over**
02	0.61	0.67	**0.71**	0.61	0.59	0.60	***0.67***	0.58
03	0.13	0.53	**0.63**	0.36	0.13	0.55	***0.56***	0.38
04	0.12	0.66	**0.72**	0.46	0.12	0.59	***0.70***	0.53
05	0.13	0.69	**0.75**	0.50	0.13	0.61	***0.72***	0.50
06	0.22	***0.67***	**0.71**	0.52	0.24	0.60	0.64	0.52
07	0.19	0.53	**0.66**	0.41	0.19	0.57	***0.63***	0.41
08	0.71	0.70	**0.74**	0.61	0.71	0.61	***0.72***	0.55
09	0.16	0.68	**0.73**	0.52	0.17	0.60	***0.70***	0.54
10	0.18	0.67	**0.72**	0.47	0.23	0.60	***0.71***	0.50
11	0.60	0.68	**0.73**	0.55	0.59	0.62	***0.69***	0.55
12	0.29	0.66	**0.73**	0.55	0.27	0.59	***0.67***	0.54

Table IV are shown the results obtained using online learners monthly. The ARF method achieved the best results for every month, and LevBag, CSARF, and KUE appeared with competitive F-measure values. Considering that CSARF and KUE are projected for imbalanced datasets, we believe that an average of 0.74 of F-measure (see Figure 3 to summarized results) are satisfactory results that overcome batch methods. Surprisingly, for incremental approaches, the hyper-parameter tuning does not deliver improvements, and all the results were obtained using the default configuration.

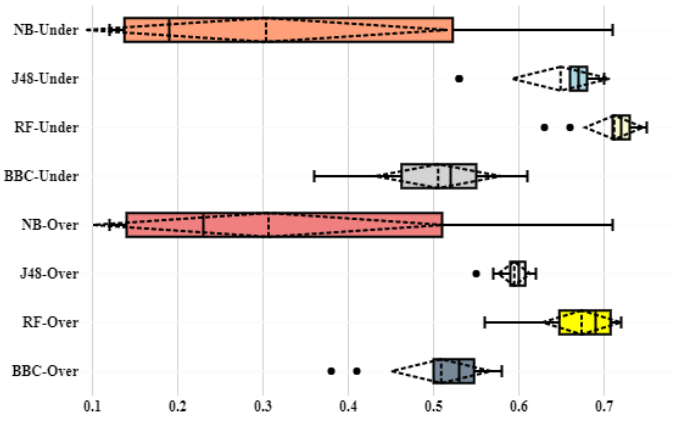

Fig. 2. Summarizing results monthly for batch methods.

TABLE IV
MAIN RESULTS MONTHLY FOR ONLINE APPROACHES.

Month	**NB**	**HT**	**HAT**	**LEVBAG**	**ARF**	**ARFRE**	**CSARF**	**KUE**
02	0.67	0.70	0.69	***0.79***	**0.81**	0.77	0.78	0.76
03	0.56	0.64	0.64	***0.74***	**0.75**	0.70	0.72	0.73
04	0.51	0.71	0.69	***0.76***	**0.76**	0.72	0.73	0.73
05	0.71	0.74	0.74	0.75	**0.77**	0.72	0.72	***0.75***
06	0.61	0.63	0.67	***0.79***	**0.81**	0.76	0.78	0.76
07	0.54	0.62	0.73	***0.75***	**0.76**	0.73	0.73	0.74
08	0.71	0.72	0.74	***0.75***	**0.76**	0.73	0.73	0.75
09	0.71	0.72	0.72	***0.75***	**0.76**	0.73	0.74	0.74
10	0.69	0.70	0.70	***0.73***	**0.74**	0.71	0.71	0.73
11	0.67	0.69	0.71	***0.74***	**0.75**	0.71	0.72	0.73
12	0.63	0.66	0.67	0.74	**0.76**	0.73	***0.74***	0.72

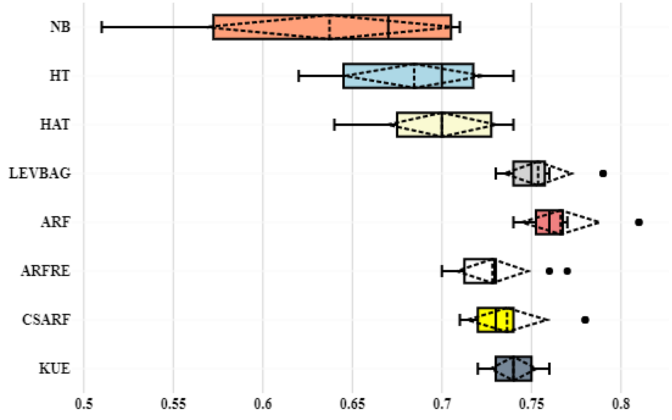

Fig. 3. Summarizing results monthly for online methods.

It is important to highlight that ARFRE also achieved good results with a macro F-measure mean of 0.73. We notice a stable behavior during the months for every method with the best results for February (0.81 with ARF, 0.79 for LevBag, and 0.78 using CSARF) and June (0.81, 0.79, and 0.78 for ARF, LevBag and CSARF, respectively).

B. Results for Training with Forty Percent of Data

Another validation approach applied in this work was a holdout method using 40% of the dataset to train the algorithms and 60% to test the classifier. In the first validation, we experimented with batch methods using under and oversampling, see Table V. The best classifier was RF-Under with 0.73 of average F-measure (see Figure 4 to summarized results) achieving until 0.75 in May and August. NB-Under also achieved interesting results with 0.69 of average F-measure. We notice that the BBC had a poor performance since the method is designed to deal with imbalanced data.

TABLE V
MAIN RESULTS WITH FORTY PERCENT TRAINING FOR BATCH METHODS.

Month	**NB-Under**	**J48-Under**	**RF-Under**	**BBC-Under**	**NB-Over**	**J48-Over**	**RF-Over**	**BBC-Over**
05	0.74	0.70	***0.75***	0.58	0.74	0.68	**0.75**	0.58
06	0.71	0.68	***0.73***	0.66	0.71	0.67	**0.73**	0.67
07	0.71	0.68	**0.74**	0.64	0.71	0.66	***0.74***	0.64
08	0.71	0.69	**0.75**	0.63	0.71	0.67	***0.74***	0.66
09	0.68	0.68	***0.73***	0.63	0.68	0.67	**0.73**	0.63
10	0.67	0.67	**0.72**	0.58	0.66	0.65	***0.72***	0.60
11	0.65	0.67	**0.72**	0.60	0.65	0.65	**0.72**	0.60
12	0.66	0.65	***0.71***	0.60	0.64	0.64	**0.71**	0.62

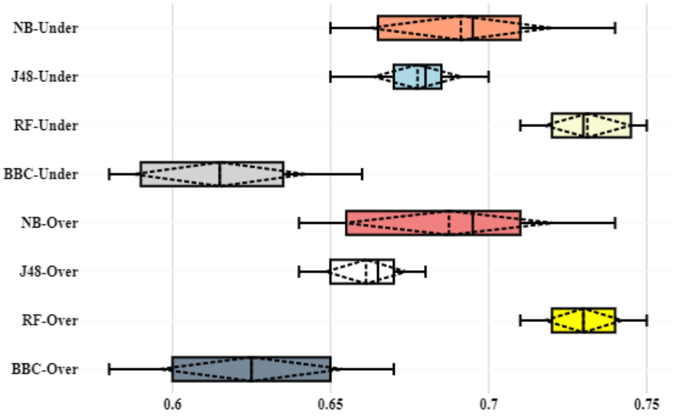

Fig. 4. Summarizing results with forty percent training for batch methods.

Table VI contains the results for online methods using 40% of the data for training and 60% to test using the interleaved test-then-train approach. The best method was ARF with 0.76 of average F-measure (see Figure 5 to summarized results), roughly 3% better than the RF-Under batch algorithm. LevBag also achieves satisfactory results with 0.75 of mean and 0.02 of standard deviation. We notice that KUE, CSARF, and ARFRE also achieved enough results, respectively 0.74, 0.73, and 0.72 of average F-measure. Nevertheless, besides these methods being designed to deal with imbalanced datasets, ARF shows up with an alternative way to DIFOT predictions.

TABLE VI
MAIN RESULTS WITH FORTY PERCENT TRAINING FOR ONLINE METHODS.

Month	**NB**	**HT**	**HAT**	**LEVBAG**	**ARF**	**ARFRE**	**CSARF**	**KUE**
05	0.71	0.74	0.74	***0.76***	**0.77**	0.70	0.71	0.74
06	0.70	0.68	0.70	***0.80***	**0.81**	0.78	0.79	0.78
07	0.68	0.71	0.71	***0.75***	**0.76**	0.71	0.73	0.74
08	0.72	0.72	0.72	***0.76***	**0.77**	0.73	0.73	0.75
09	0.70	0.71	0.71	***0.75***	**0.76**	0.73	0.74	0.74
10	0.67	0.69	0.69	***0.74***	**0.74**	0.71	0.71	0.73
11	0.68	0.71	0.71	***0.74***	**0.75**	0.71	0.73	0.73
12	0.65	0.66	0.69	0.74	**0.76**	0.73	***0.74***	0.73

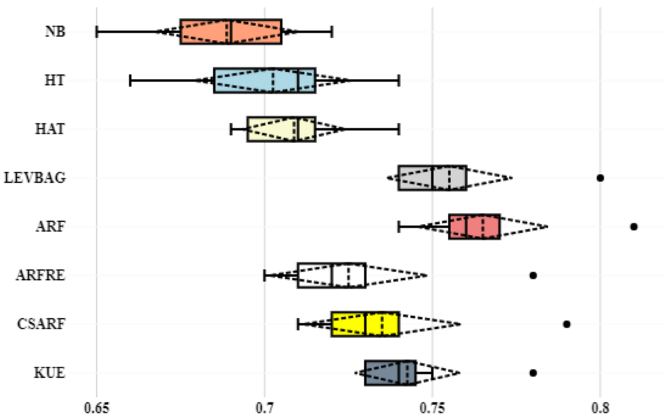

Fig. 5. Summarizing results with forty percent training for online methods.

C. Results for Training with Fifty Percent of Data

In order to verify the impact of more train data for performance classification, we used the holdout validation with 50% of the instances to train the batch methods and 50% of the samples to test. The results are in Table VII. Similar results to 40%-60% approach were obtained in this validation process using 50% of the to train, which RF was the best classifier, achieving 0.72 of average F-measure, but using the undersampling technique (see Figure 6 to summarized results). The NB also achieved enough results with 0.69 of F-measure. The J48 and BBC performed poorly, with mean F-Measure results below 0.60.

Again we compare batch methods with online classifiers using the same parts of the dataset to train and test using the interleaved test-then-train approach, according to Table VIII. The ARF achieved the best average F-measure value with 0.76 and 0.02 of standard deviation (see Figure 7 to summarized results), followed by CSARF with 0.75 average F-measure with 0.02 standard deviation. These results show the robust performance of tree ensemble-based classifiers. However, LevBag, KUE, and ARFRE achieved satisfactory results with 0.74, 0.74, and 0.72 of average F-measure, respectively.

TABLE VII
MAIN RESULTS WITH FIFTY PERCENT TRAINING FOR BATCH METHODS.

Month	**NB-Under**	**J48-Under**	**RF-Under**	**BBC-Under**	**NB-Over**	**J48-Over**	**RF-Over**	**BBC-Over**
06	**0.72**	0.58	0.72	0.57	***0.72***	0.61	0.67	0.57
07	0.71	0.58	**0.73**	0.54	0.71	0.62	***0.72***	0.57
08	0.72	0.58	**0.74**	0.56	0.72	0.61	***0.73***	0.57
09	0.69	0.55	**0.72**	0.52	0.70	0.59	***0.70***	0.55
10	0.67	0.54	**0.72**	0.57	0.67	0.59	***0.69***	0.59
11	0.67	0.56	**0.72**	0.54	0.67	0.60	***0.69***	0.55
12	0.66	0.56	**0.69**	0.55	***0.66***	0.60	0.66	0.59

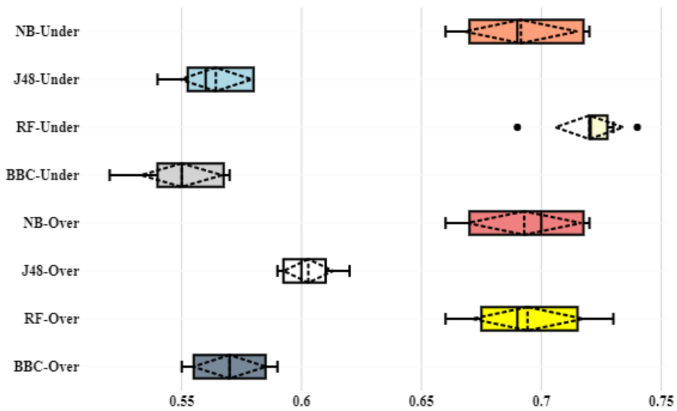

Fig. 6. Summarizing results with fifty percent training for batch methods.

TABLE VIII
MAIN RESULTS WITH FIFTY PERCENT TRAINING FOR ONLINE METHODS.

Month	**NB**	**HT**	**HAT**	**LEVBAG**	**ARF**	**ARFRE**	**CSARF**	**KUE**
06	0.72	0.70	0.67	0.77	**0.80**	0.77	***0.80***	0.76
07	0.73	0.71	0.71	***0.74***	**0.77**	0.71	0.72	0.73
08	0.73	0.72	0.70	0.74	**0.76**	0.73	0.74	***0.74***
09	0.71	0.70	0.68	0.74	**0.76**	0.71	***0.74***	0.73
10	0.70	0.69	0.69	0.74	**0.76**	0.71	***0.75***	0.73
11	0.68	0.70	0.70	0.72	**0.74**	0.70	***0.74***	0.73
12	0.70	0.65	0.68	0.73	***0.75***	0.72	**0.76**	0.73

D. Results for Training with Sixty Percent of Data

In Table IX are the F-measure results for batch methods using sixty percent of the dataset to train and forty percent to test. We notice some decrease in performance for the algorithms experimented compared to other validation approaches like forty and fifty percent of the dataset for training. The exception was RF-Over, which achieved 0.71 of the average F-measure (0.01 of standard deviation) with the best result in August (mean of 0.73 for F-measure). In addition, according to Figure 8 it is possible to view outliers in RF-Under and also J48-Over which indicates instability in tree-based methods with more data information.

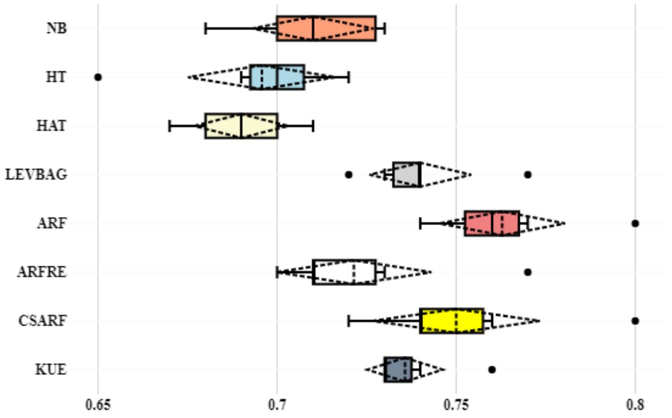

Fig. 7. Summarizing results with fifty percent training for online methods.

TABLE IX
MAIN RESULTS WITH SIXTY PERCENT TRAINING FOR BATCH METHODS.

Month	NB-Under	J48-Under	RF-Under	BBC-Under	NB-Over	J48-Over	RF-Over	BBC-Over
07	***0.69***	0.54	0.66	0.39	0.69	0.56	**0.71**	0.47
08	0.71	0.49	***0.71***	0.53	0.71	0.59	**0.73**	0.50
09	0.67	0.50	**0.73**	0.56	0.67	0.57	***0.72***	0.51
10	0.56	0.49	**0.71**	0.57	0.55	0.55	***0.70***	0.50
11	0.37	0.50	***0.70***	0.56	0.35	0.56	**0.71**	0.50
12	0.26	0.55	**0.71**	0.47	0.26	0.56	***0.69***	0.47

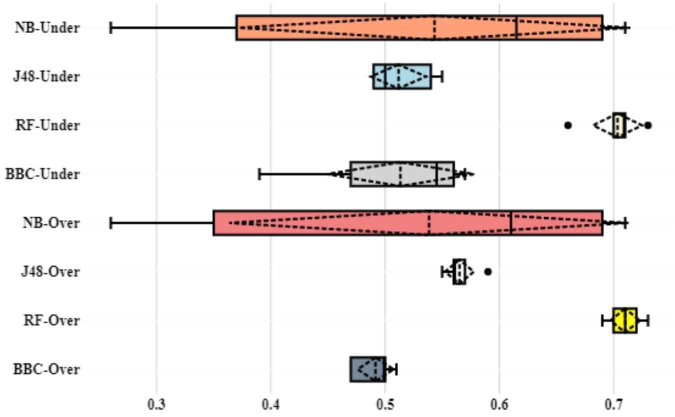

Fig. 8. Summarizing results with sixty percent training for batch methods.

We continue our analysis with Table X which presents the main F-measure results for online classifiers using sixty percent of the dataset for training. The main observation is about the stable performance with more data differently of batch methods that decreased the results. The ARF was maintained as the best learner (0.77 of average F-measure) followed by CSARF with 0.75 of mean. Another relevant highlight is the standard deviation smaller than those observed for batch methods, i.e., 0.01 for all online approaches, except NB and HT. In addition, using sixty percent to train online learners methods, only CSARF have outlier results (see Figure 9) showing a more stable behavior.

TABLE X
MAIN RESULTS WITH SIXTY PERCENT TRAINING FOR ONLINE METHODS.

Month	NB	HT	HAT	LEVBAG	ARF	ARFRE	CSARF	KUE
07	0.72	0.71	0.66	0.74	***0.77***	0.72	**0.75**	0.74
08	0.73	0.71	0.66	0.75	***0.78***	0.73	**0.76**	0.74
09	0.71	0.70	0.66	0.75	***0.78***	0.73	**0.76**	0.73
10	0.68	0.68	0.63	0.73	***0.76***	0.72	**0.73**	0.72
11	0.67	0.67	0.64	0.73	***0.76***	0.72	**0.75**	0.72
12	0.69	0.67	0.66	0.73	***0.77***	0.73	**0.76**	0.72

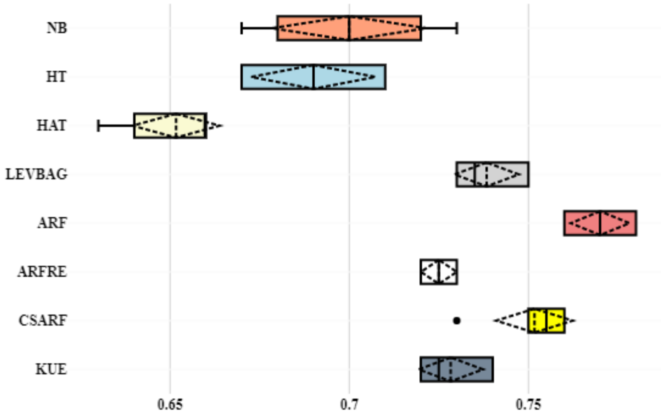

Fig. 9. Summarizing results with sixty percent training for online methods.

E. Feature Importance Analysis

In this work, we are interested in answering the following question: Considering intrinsic time series in this dataset, are there feature drifts in its data? We experimented evaluate the dataset month to month and with different train-test split percentages. For such evaluation, we used the RF classifier to measure feature importance according to the MDI metric in different dataset partitions. We notice that besides the timespan characteristics, there are no feature drifts in the DIFOT database. In general, the feature importance ranking suffers only a few changes, and three of them remain in the top for all approaches and the end of the ranking by shifting between periods or splits. However, the robust results reported using online learners show the relevance of these approaches to deal with DIFOT comparing to traditional batch methods. We believe that this finding is related to the less important characteristics, that is, although there is no change in the best attributes, in a few months (e.g. February and March) certain dimensions (feature forty) lose relevance and this scenario can be better adapted through online methods.

VI. Conclusion and Perspectives

In this work was presented a brief literature review of batch and online classification barely related to Supply Chain Management. More specifically, we were interested in comparing these approaches applied to DIFOT prediction using a dataset proposed for this end. The public DIFOT dataset is expected to support extensible researches in supply chain and machine learning, mainly for data stream mining considering the scarcity of this type of real data. We benchmark several portion of the DIFOT dataset evaluating the results equally among batch and online learners with a robust hyper-parameter tuning. The results obtained using data stream classifiers were inspiring for future research in DIFOT prediction, mainly using tree-based ensemble adaptive approach like ARF and its improved precursors. We do not detect feature drifts using the holdout validation method with month to month or percentages of the dataset. Considering these discoveries, we plan to continue researching online classification methods according to recent studies such as deep learning [17], [18] and rough sets theory [22]. In addition, methods to detect feature drift could enrich the scientific community focused in supply chain.

References

[1] G. C. Stevens and M. Johnson, "Integrating the supply chain... 25 years on," *International Journal of Physical Distribution & Logistics Management*, vol. 46, pp. 19–42, feb 2016.

[2] M. Christopher, *Logistics & supply chain management*, 3rd ed. Great Britain: Pearson, 2011.

[3] T. McLean, *On Time, in Full: Achieving Perfect Delivery with Lean Thinking in Purchasing, Supply Chain, and Production Planning*, M. Sinocchi, Ed. New York, United States of America: Productivity Press, 2017.

[4] N. Mishra and A. Singh, "Use of twitter data for waste minimisation in beef supply chain," *Annals of Operations Research*, vol. 270, pp. 337–359, sep 2018.

[5] S. J. Angarita-Zapata, A. Alonso-Vicario, A. D. Masegosa, and J. Legarda, "A taxonomy of food supply chain problems from a computational intelligence perspective," *Sensors*, vol. 21, no. 20, p. 6910, oct 2021.

[6] G. Baryannis, S. Dani, and G. Antoniou, "Predicting supply chain risks using machine learning: The trade-off between performance and interpretability," *Future Generation Computer Systems*, vol. 101, pp. 993–1004, dec 2019.

[7] S. Tiwari, H.-M. Wee, and Y. Daryanto, "Big data analytics in supply chain management between 2010 and 2016: Insights to industries," *Computers & Industrial Engineering*, vol. 115, pp. 319–330, jan 2018.

[8] N. Shukla and S. Kiridena, "A fuzzy rough sets-based multi-agent analytics framework for dynamic supply chain configuration," *International Journal of Production Research*, vol. 54, pp. 6984–6996, feb 2016.

[9] H. Zhang, Y. Shi, and J. Tong, "Online supply chain financial risk assessment based on improved random forest," *Journal of Data, Information and Management*, vol. 3, no. 1, pp. 41–48, feb 2021.

[10] S. Ramírez-Gallego, B. Krawczyk, S. García, M. Woźniak, and F. Herrera, "A survey on data preprocessing for data stream mining: Current status and future directions," *Neurocomputing*, vol. 239, pp. 39–57, 2017.

[11] J. P. Barddal, L. Loezer, F. Enembreck, and R. Lanzuolo, "Lessons learned from data stream classification applied to credit scoring," *Expert Systems with Applications*, vol. 162, p. 113899, dec 2020.

[12] D. Won, P. Jansen, and J. Carbonell, "Minimizing and recovering from the effect of concept drift via feature selection," in *24th European Conference on Artificial Intelligence 2020*, ser. Frontiers in Artificial Intelligence and Applications. IOS Press, 2020, pp. 1611–1617.

[13] M. B. de Moraes and A. L. S. Gradvohl, "A comparative study of feature selection methods for binary text streams classification," *Evolving Systems*, pp. 1–17, oct 2020.

[14] D. P. Melidis, M. Spiliopoulou, and E. Ntoutsi, "Learning under feature drifts in textual streams," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 527–536.

[15] R. Shivakumaraswamy, C. Beyer, V. Unnikrishnan, E. Ntoutsi, and M. Spiliopoulou, "Active feature acquisition for opinion stream classification under drift," in *CEUR Workshop Proceedings 2444 (2019)*, vol. 2444. Aachen: RWTH, 2019, pp. 108–111.

[16] C. Fahy and S. Yang, "Dynamic feature selection for clustering high dimensional data streams," *IEEE Access*, vol. 7, pp. 127 128–127 140, jul 2019.

[17] J. Holmberg and N. Xiong, "Online feature selection via deep reconstruction network," in *International Conference on Harmony Search Algorithm*. Springer, 2019, pp. 194–201.

[18] S. Sahmoud and H. R. Topcuoglu, "A general framework based on dynamic multi-objective evolutionary algorithms for handling feature drifts on data streams," *Future Generation Computer Systems*, vol. 102, pp. 42–52, jan 2020.

[19] J. C. Chamby-Diaz, M. Recamonde-Mendoza, and A. L. C. Bazzan, "Dynamic correlation-based feature selection for feature drifts in data streams," in *8th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2019, pp. 198–203.

[20] P. Duda, K. Przybyszewski, and L. Wang, "A novel drift detection algorithm based on features' importance analysis in a data streams environment," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 10, pp. 287–298, may 2020.

[21] J. Duarte and J. Gama, "Feature ranking in hoeffding algorithms for regression," in *Proceedings of the Symposium on Applied Computing*, 2017, pp. 836–841.

[22] A. Ferone and A. Maratea, "Adaptive quick reduct for feature drift detection," *Algorithms*, vol. 14, no. 2, p. 58, 2021.

[23] D. Zhao and Y. S. Koh, "Feature drift detection in evolving data streams," in *International Conference on Database and Expert Systems Applications*. Springer, 2020, pp. 335–349.

[24] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. of the sixth ACM International Conference on Knowledge Discovery and Data Mining*, Boston, United States of America, 2000, pp. 71–80.

[25] A. Bifet and R. Gavaldà, "Adaptive learning from evolving data streams," in *International Symposium on Intelligent Data Analysis*, 2009, pp. 249–260.

[26] A. Bifet and R. Gavalda, "Learning from time-changing data with adaptive windowing," in *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 2007, pp. 443–448.

[27] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2010, pp. 135–150.

[28] H. M. Gomes, A. Bifet, J. Read, J. P. Barddal, F. Enembreck, B. Pfharinger, G. Holmes, and T. Abdessalem, "Adaptive random forests for evolving data stream classification," *Machine Learning*, vol. 106, no. 9-10, pp. 1469–1495, jun 2017.

[29] L. E. B. Ferreira, H. M. Gomes, A. Bifet, and L. S. Oliveira, "Adaptive random forests with resampling for imbalanced data streams," in *2019 International Joint Conference on Neural Networks*. IEEE, 2019, pp. 1–6.

[30] L. Loezer, F. Enembreck, J. P. Barddal, and A. de Souza Britto Jr, "Cost-sensitive learning for imbalanced data streams," in *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, 2020, pp. 498–504.

[31] A. Cano and B. Krawczyk, "Kappa updated ensemble for drifting data stream mining," *Machine Learning*, vol. 109, no. 1, pp. 175–218, oct 2020.

[32] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," *Advances in neural information processing systems*, vol. 26, pp. 431–439, 2013.