Approaches to Robust Online Explanations

***Abstract*—Increasing use of black-box models such as Deep Neural Networks (DNNs) in practice prompted search for ways to explain the behavior of these systems. Widely used in this area are model-agnostic local attribution methods like LIME and SHAP. They explain model predictions on individual data points by assigning scores to features. It is, however, questionable whether use of local explanation methods is appropriate in an online learning environment. Continuous arrival of novel data, possible noise in the channel and the resulting changes in the model make explanations noisy and vulnerable to concept drift in the data. In this paper we propose various approaches to this problem, including the use of a feature selection framework which keeps track of the explained model distribution, ensemble learning and probabilistic reasoning to increase the robustness of explanations over time. We place emphasis on efficiency and stability as well as local and global faithfulness of explanations. Furthermore, we present a possible way of representing global explanations of data streams as well as evaluation metrics for robustness and local quality of explanations. We evaluate the presented approaches on data sets from two domains and two synthetically generated data sets. Results show that combining global with local explanations improves stability without decreasing the quality of explanations and even though local explainers show comparably high global ranking similarity, stability and efficiency of global explainers is significantly higher.**

***Index Terms*—SHAP, LIME, XAI, data streams, robustness, explainable AI, ensemble, evaluation metrics**

I. Introduction

As machine learning enters all pores of our society, the importance of explainability of used models increases. Explainable AI (XAI) field aims to make the process and reasoning behind decisions of black-box models understandable to domain experts and analysts or, ideally, to everyone with no need for an intermediary [1]. Explainability is of crucial importance in critical applications like medicine and autonomous driving where consequences of wrong predictions do not allow for mistakes. For this reason, both in domains like these and other (for instance, finance and marketing) which process personal data, regulatory agencies increasingly require explanations to accompany the decisions of black-box models [2].

Once a black-box model is deployed, it has to be constantly updated to reflect changes in the environment and the data [3]. Distribution which generated the data used as input for the model changes over time. This is called *concept drift* and is a major challenge that has to be taken into account in practice. With regard to frequency of model updates, in the most extreme case models are used in an online learning environment where they are retrained at every time step. In an online learning environment data points from the data set are not all accessible at once but arrive in a stream. At every time step t a classifier sees data point x_t and possibly a label y_t, depending on whether it is in a training or test mode.

Due to unavailability of the whole data set, explainability methods used with data streams have to be robust to noise in model changes. Since the model adapts to the incoming data through retraining, local explanations behind its decisions change as well. Noise in the channel or outliers in the data could make the explainer model temporarily produce widely differing individual explanations. Such unreliable explanations would be undesirable because they would describe only a transient state of the explained model and that way render it untrustworthy. Consequently, explanations that were valid in the past might not be valid in the future.

These problems bring us to the desired properties for explainability methods in data streams. Lundberg and Lee define the properties of *Local Accuracy*, *Missingness* and *Consistency* as desirable for local attribution methods [4]. However, considering the online environment, missingness seems to be particularly vulnerable to the problem of noisy channel and local accuracy has to be considered in terms of the existing trade-off between the desired smoothness and local faithfulness of explanations.

Building on this, apart from the interpretability of explanations which can be achieved through feature attributions, we consider the following to be the goals of explainability methods in an online environment:

- *Local Faithfulness* to the actual attributions of the model. Depending on the underlying model they could be calculated in various ways. Although explanations should not fit to noise in the model, they should still be in correspondence with the way the explained classifier attributes features locally.
- *Global Faithfulness* because, while individual attributions could differ, attributions of the explainability method should correspond to the attributions of the explained model over longer periods of time.
- *Efficiency* in view of the fact that it could allow the use of the method in real-time.
- *Stability* because produced explanations should be usable over time and not vary widely in short intervals. That being said, the explainability method should adapt to model changes that result from model retraining when there is a concept drift in the data.

Here we note that local and global faithfulness goals correspond to the desired criteria mentioned in [5].

In this paper we will concentrate on post-hoc explanations with model-agnostic attribution methods that assign importance scores to features of the data. The main contributions of this paper are:

- Creation of more robust explanations in data streams through various combinations of local and global ex-

plainer models.
- Experimental results and evaluation of the presented approaches using metrics for measuring the quality and robustness of explainability methods in an online environment.

The rest of the paper is structured as follows: In the following section we will look into the related works. Subsequently different approaches to the problem of robust explanations in data streams will be presented. Results of experiments with evaluations of the described approaches are then presented and discussed. Finally, in conclusion we discuss key results and possible future work.

II. Related Works

Related works can roughly be classified according to whether they look at the local or global explanations [2]. Local explanations refer to the explanations produced to explain classification of individual data points, while global explanations take into account the model and try to explain its behavior without special consideration of individual data points. Local explanations are produced by the surrogate models which approximate the classifier locally around a data point [2]. Most of the research on explainability is concentrated on local surrogate models and the most popular methods are LIME and SHAP.

LIME (Local Interpretable Model-agnostic Explanations) is a local surrogate explainability method which approximates a black-box model in the neighborhood around an individual data point by sampling data points around it from the given black-box model and fitting a linear model to the sampled data [5]. Weights of the trained local linear model are then considered to be feature attributions which represent local explanation for the point in question. Ribeiro et al. also propose SP-LIME (Submodular pick LIME), that is using LIME explanations at a number of important data points to create a global explanation of a model. They put a particular emphasis on the selection of representative data points.

SHAP (SHapley Additive exPlanations) is a local surrogate explainability method which approximates Shapley values for features and considers them to represent an importance score of a feature [4]. Shapley values are weighted differences between a classifier output with and without all possible sets of features. To represent missing features it computes a baseline computed from a number of already seen data points. SHAP fulfills local accuracy, missingness and consistency properties [4].

Compared to local surrogate explainability methods which mostly use linear models, apart from SP-LIME, there do not seem to be many proposed global surrogate methods that use linear models [2]. Most of the approaches learn rules or trees that approximate the behavior of the model. For example, a soft decision tree approach proposed by Frosst and Hinton trains a decision tree model based on the inputs and outputs of a DNN [6]. Because it is trained with stochastic gradient descent, it can be updated alongside the explained DNN in an online fashion using the outputs of the retrained DNN.

Explainable online learning is still a nascent field and accordingly not much research has been done in the area. Nevertheless, online approaches to feature selection could offer useful insights for explainability because they have to keep track of the feature quality over time. A particular framework that we apply to explanations in this work is FIRES [7].

FIRES (Fast, Interpretable and Robust feature Evaluation and Selection) is a framework for online feature selection [7]. To choose the best features at every time step, FIRES keeps track of a probability distribution underlying the explained model parameters and uses the estimated parameters of the model distribution to assign scores to features. Since FIRES assumes that model parameters are normally distributed, through calculation of feature scores using mean and variance, it incorporates both estimated importance and uncertainty. Notably, consideration of importance and uncertainty agrees with desired properties of a globally faithful and stable explainer model.

III. Approaches

In this section we propose different approaches for generation of explanations in data streams used in the experiments.

1) Local Explainer: Our first approach to explain the model in an online environment is to use one of the local attribution methods as an explainer model. At every time step t we calculate the feature attributions a_t from the output of a classifier f and the data point x_t using either LIME or SHAP as an explainer l, that is:

$$a_t = l(x_t, f(x_t), X) \tag{1}$$

Since SHAP needs to establish a baseline value for missing features and implementation of LIME can use training data to approximate the distribution used to draw local samples, these models also need a baseline sample set (X in (1)).

2) Global Explainer: Because local explainers might be vulnerable to noise, we look at global explainers as well. For this, we use weights which FIRES assigns to features at every time point [7]. We reinterpret feature weights as attributions, and since we concentrate on post-hoc explanations, we reinterpret the distribution of model parameters that FIRES keeps track of as the distribution of feature attributions. Interpreted this way, FIRES could be considered to be a global surrogate model. In particular, we use FIRES-GLM version which uses a generalized linear model with a probit link function as the surrogate model because it makes an assumption that the number of model parameters is the same as the number of features which is appropriate and convenient for the explainer interpretation of FIRES.

More concretely, to use FIRES as a global surrogate explainer predictions of the explained classifier f are used as inputs, and attributions at time t are:

$$a_t = \text{FIRES}(x_t, f(x_t)) \tag{2}$$

3) Ensemble of Explainers: Since FIRES keeps track of the probability distribution of model parameters, it allows us to consider the uncertainty in the underlying model and thereby make explanations more stable. To do this, an ensemble of local explainers is used to generate explanations at every time step.

Explanations at time t are obtained by extracting μ_t and Σ_t parameters of model distribution from FIRES, sampling N model parameters $\{\theta_1, \theta_2, \ldots, \theta_N\}$ s.t. $\theta_i \sim \mathcal{N}(\mu_t, \Sigma_t)$ from the model distribution, creating classifiers f_{θ_i} from each set of the sampled model parameters and aggregating local explanations (generated with SHAP or LIME like in (1)) of sampled models:

$$a_t = \frac{1}{N} \sum_{i=0}^{N} e(x_t, f_{\theta_i}(x_t), X) \tag{3}$$

4) Weighted Global and Local Explainer: Another way to combine global and local explainability methods is by explicitly weighting the explanations generated by a global and local explainer. Let global explainer be g and l be a local explainer as above, generated attributions are:

$$a_t = \frac{w_l \cdot l(x_t, f(x_t)) + w_g \cdot g(x_t, f(x_t))}{w_l + w_g} \tag{4}$$

5) Weighted Gaussian Explainer: Weighted Gaussian explainer is a linear global surrogate method in which we assume a_t to be a Gaussian random variable and define a likelihood by assuming that output y_t of the classifier depends linearly on the feature attributions a_t and data points x_t.

$$P(y_t|x_t, a_t) = \mathcal{N}(y_t; x_t^T a_t, \Lambda) \tag{5}$$

We now put a Gaussian prior on a_t. At initialization (a_0) we use a standard Gaussian distribution.

$$P(a_t) = \mathcal{N}(a_t; \mu_{t-1}, \Sigma_t) \tag{6}$$

Since conjugate prior of a Gaussian distribution is also a Gaussian distribution and because there is only a linear relationship among the random variables, the posterior is then $P(a_t|x_t, y_t) = \mathcal{N}(a_t; \mu_t, \Sigma_t)$ where [8]:

$$\Sigma_t = \Sigma_{t-1} - x^T \Sigma_{t-1} \big(x^T \Sigma_{t-1} x + \Lambda\big)^{-1} \Sigma_{t-1} x \tag{7}$$

$$\mu_t = \mu_{t-1} + \Sigma_t \big(x^T \Sigma_{t-1} x + \Lambda\big)^{-1} x \left(y - \tanh(x^T a)\right) \tag{8}$$

For the calculation of a residual, in order to increase the stability of the results assuming that the outputs of the classifier are -1 and 1, we use *tanh* to squash the output into $[-1, 1]$.

In the end, we use the obtained posterior at time t as the prior at time $t+1$, and for attribution at time t we take the mean of the posterior (i.e. $a_t = \mu_t$ (8)). Thus, since the posterior also depends on the previous attribution, weighted Gaussian explainer can be interpreted as a hidden markov model.

IV. Experiments and Results

Approaches presented in the previous section were evaluated using *xEvaluator* environment for explanation of explainability methods in data streams. Implementation of the weighted Gaussian approach along with maskers of other explainers are included in xEvaluator and can be found on github[1]. Code used to produce the results and additional figures is available there. All of the experiments were done with defined random state variables on a machine with an i5 6200U CPU and 12 Gb RAM, running Ubuntu 18.04.

We evaluated the presented approaches using the prequential evaluation on data sets listed in Table I. Domains of all of the features were rescaled to $[0, 1]$. For manipulation and creation of data streams we used scikit-multiflow [9], a framework for experimentation with data streams in Python. Additionally, in order to get a better insight into explanations on data sets that could be encountered with deployed models, we also use two data sets from practice available in UCI repository [10]. These data sets were also converted into streams using scikit-multiflow.

Spambase data set contains a collection of mails labeled as 'spam' or 'not spam' with selected personalized features. Before using spambase as a data stream, we permute it to simulate a more realistic environment where spam and genuine mails are arriving independently of each other. Finally, card default [11] data set contains data points representing individuals with labels denoting whether a person defaulted in the following month.

With regard to classifiers that are explained, we use logistic regression, because, since the data is scaled to $[0, 1]$ range, weights of the linear model can be defined as local ground truth attributions. In addition, to simulate black-box models found in practice, we also use a feedforward neural network with ReLU activation functions in hidden layers and logistic function at the output layer. Local ground truth of the neural network classifier at every time point was created by computing a mean for each feature of the weights between input and the first hidden layer where activations at time t are not zero.

For each of the desired properties of explainability methods mentioned in introduction we define suitable evaluation metric. *Efficiency* of the explainers we evaluate by calculating the mean time needed for computation of explanations over all time steps. Furthermore, we evaluate cumulative *stability* of explanations in the following way (D is the number of features):

$$S_{t+1} = S_t + \left| \frac{a_{t+1}}{\sum_{i=1}^{D} |(a_{t+1})_i|} - \frac{a_t}{\sum_{i=1}^{D} |(a_t)_i|} \right| \tag{9}$$

That is, we accumulate changes over time in normalized attributions between two subsequent time steps. Moreover, as an evaluation metric for *local faithfulness* we use absolute

[1] https://github.com/delanorr/xevaluator

TABLE I
DATA SETS

	Samples	Features	Positive Class
Synthetic			
RTG	5000	100	1105
RBF	5000	100	2198
Application			
Spambase	4601	57	1812
Card Default	5000	23	1107

distance between the attribution output of the explainer and the actual weights of the linear classifier:

$$LF_{t+1} = LF_t + \left| \frac{a_{t+1}}{\sum_{i=1}^{D} |(a_{t+1})_i|} - \frac{g_{t+1}}{\sum_{i=1}^{D} |(g_{t+1})_i|} \right| \quad (10)$$

where $(g_t)_i$ represents the ground truth, e.g. a parameter associated with the feature i of the explained linear model at time step t.

Lastly, to evaluate *global faithfulness* we compare rankings of Top-10 cumulative feature attributions of evaluated explainer with Top-10 cumulative sums of the ground truth feature weights at the end of the data stream. We record a number of Top-10 cumulative feature attributions that are present in Top-10 ground truth feature weights. Similarity of the ranking we represent by a sum over the inverse rank distances weighted by a position of the feature in Top-10 ground truth:

$$R = \sum_{i \in P} \frac{(K+1) - r_i^{(g)}}{1 + |r_i^{(a)} - r_i^{(g)}|} \quad (11)$$

where K is the number of Top-K features (here $K = 10$), $r_i^{(a)}$ is the rank of feature i in Top-K global attributions, $r_i^{(g)}$ is the rank of feature i in Top-K global ground truth weights and P is a set of features shared between these Top-K rankings.

Now, regarding the hyperparameters of the presented approaches, we mention only important settings and those cases where we did not use the default parameters. In a combination of a local explainer with FIRES we gave the same weights to both explainers. Furthermore, for FIRES, to use the whole gradient, the learning rate was set to 1. Size of the baseline set for SHAP and LIME was set to 100, and for local evaluations in LIME and approximation of Shapley value in SHAP, $2 \cdot$ `num_features` samples was used. Before starting the generation of explanations, all of the models were pretrained for 100 time steps.

The results of the experiments are shown in Table II.

V. DISCUSSION AND CONCLUSION

Local explainers and in particular SHAP are significantly more unstable than global explainers. This difference is even more pronounced because of a small number of local evaluations on which local explainers rely. Overall LIME performed better in most areas apart from ranking accuracy where SHAP performs better. On the other hand, global explainers were much more stable and efficient than local explainers. Moreover, in cases where their inductive bias was compatible with the explained model they were also the most locally faithful approaches. Surprisingly, ranking accuracy of global explainers was not better and sometimes even worse than local explainers.

Using an ensemble of models sampled from model distribution improved both stability and local faithfulness measures for SHAP. Conversely, ensemble of LIME explainers performed worse than a single LIME explainer. Furthermore, the efficiency of ensembles of explainers is severely reduced by the need to create multiple explained models at every time step. An explicit weighting approach to combination of local and global explanations seems to improve the stability while keeping and sometimes surpassing local and global faithfulness of a sole local explainer.

When it comes to the limitations of presented approaches, the problem with surrogate approaches appears when explained classifiers can perfectly fit the data. In that case we get the same explanations for all of the classifiers because the surrogate model cannot differentiate between the classifiers and fits to seen data points and their labels. This is, however, in general a problem in model agnostic post-hoc explanations because explainers only get to see data points and corresponding labels. Local explainers, while sacrificing efficiency, partially circumvent this problem through additional local evaluations at an individual time step. Consequently, their resulting explanations depend less on the inductive bias of the surrogate model. Another limitation of the presented methods is that they are applicable only when features have meanings, i.e. they are in general not applicable to images.

Possible future work in the area of online explanations could be further investigation into the combinations of global and local explainers. For example, explanations from local explainers could be incorporated into the prior of the weighted Gaussian explainer. Further, using a (possibly time-weighted) sliding window could potentially provide better evaluation of global faithfulness through cumulative feature attributions because otherwise initial erroneous attributions could overly influence the results. Moreover, research of explanations through visualization by, for instance, using cumulative feature attributions could potentially identify concept drift. Finally, some of the existing surrogate rule extraction approaches might be relevant for data stream environment [2].

In this paper we presented and evaluated explainability methods in an online learning environment. We described desirable properties of an online explainer and defined corresponding evaluation metrics. Furthermore, we looked at local explainers (LIME and SHAP), global explainers (FIRES and weighted Gaussian) as well as two possible combinations of local and global explainers. Experiments on four data sets show that each of the approaches has its advantages. While explanations produced by a combination of local and global explainers offer improved stability without decrease in explanation quality, it seems to us that much higher efficiency and stability of global explainers makes them more suitable for online learning environment.

TABLE II
EXPERIMENT RESULTS

Classifier	Data set	Accuracy[a]	Explainer	Stability[l] (9)	Local faithfulness[l] (10)	Top-10: Presence[h]	Top-10: Similarity[h] (11)	Time[b]
Logistic Regression								
	RTG	0.76	SHAP	8,269.73	8,113.30	**9**	**52.00**	8.53
			SHAP-Ensemble	7,131.70	7,263.96	8	32.87	161.40
			SHAP+FIRES	4,999.38	7,160.12	8	34.00	9.29
			LIME	2,490.03	4,627.24	7	26.20	6.90
			LIME-Ensemble	4,110.04	5,718.96	8	27.33	140.98
			LIME+FIRES	1,556.06	5,483.81	6	30.00	8.21
			FIRES	169.57	7,625.89	7	31.00	0.94
			WeightedGaussian	**121.29**	**3,509.65**	8	34.50	**0.82**
	RBF	1.00	SHAP	8,400.35	7,727.88	**10**	16.20	10.78
			SHAP-Ensemble	6,999.86	6,817.20	7	12.14	197.72
			SHAP+FIRES	5,414.15	6,718.39	6	**17.25**	10.35
			LIME	1,463.99	3,590.53	6	10.13	6.99
			LIME-Ensemble	2,130.56	3,464.96	5	9.50	139.77
			LIME+FIRES	1,081.33	4,869.30	6	10.14	8.73
			FIRES	144.67	6,319.55	6	16.25	0.96
			WeightedGaussian	**34.44**	**1,830.01**	6	10.12	**0.94**
	Spambase	0.88	SHAP	7,168.42	8,148.04	**7**	14.06	3.41
			SHAP-Ensemble	3,278.98	6,726.97	3	8.00	72.93
			SHAP+FIRES	4,450.45	7,482.92	5	13.25	4.66
			LIME	2,644.31	4,006.69	6	18.37	6.08
			LIME-Ensemble	2,141.79	5,253.20	1	1.75	106.28
			LIME+FIRES	1,864.88	5,109.33	6	11.63	6.73
			FIRES	39.59	6,565.19	4	12.00	0.79
			WeightedGaussian	**29.14**	**2,262.38**	**7**	**23.00**	**0.52**
	Credit default	0.69	SHAP	6,546.01	8,161.52	**8**	**31.28**	1.65
			SHAP-Ensemble	7,190.42	8,028.11	**8**	10.48	32.30
			SHAP+FIRES	4,478.32	7,633.85	**8**	22.76	2.43
			LIME	3,125.35	5,192.01	6	16.78	4.34
			LIME-Ensemble	4,624.49	5,584.53	6	12.46	84.60
			LIME+FIRES	2,017.00	5,914.34	7	25.98	6.00
			FIRES	**164.13**	6,659.01	6	13.42	0.91
			WeightedGaussian	250.03	**4,738.61**	5	15.95	**0.41**
Feedforward Neural Network								
	RTG	0.81	SHAP	5,013.39	7,215.31	**5**	8.62	17.88
			SHAP-Ensemble	7,139.58	7,349.94	1	9.00	325.90
			SHAP+FIRES	3,087.10	6,415.79	3	10.17	17.85
			LIME	813.34	6,141.59	3	**11.83**	11.56
			LIME-Ensemble	1,418.90	7,178.89	1	4.00	130.22
			LIME+FIRES	566.21	6,163.52	3	4.57	12.59
			FIRES	**43.70**	**5,966.87**	3	10.07	**0.82**
			WeightedGaussian	121.29	8,582.27	**5**	9.37	5.71
	RBF	1.00	SHAP	8,544.08	8,394.33	0	0.00	28.46
			SHAP-Ensemble	7,242.76	7,835.08	2	4.33	484.76
			SHAP+FIRES	5,448.43	6,879.99	0	0.00	30.99
			LIME	1,508.16	7,647.68	0	0.00	12.34
			LIME-Ensemble	290.18	**5,043.64**	**4**	**5.33**	139.00
			LIME+FIRES	1,128.47	6,770.07	1	1.43	13.23
			FIRES	109.29	6,259.52	0	0.00	**0.93**
			WeightedGaussian	**34.44**	7,625.24	0	0.00	6.05
	Spambase	0.93	SHAP	7,283.82	7,968.65	**6**	**15.43**	9.10
			SHAP-Ensemble	5,820.94	7,711.02	2	1.72	83.28
			SHAP+FIRES	4,626.41	6,502.62	3	3.66	10.04
			LIME	1,536.41	5,739.14	3	3.75	12.41
			LIME-Ensemble	1,584.98	5,947.33	3	4.50	103.00
			LIME+FIRES	1,108.42	**5,389.49**	3	7.60	12.97
			FIRES	**28.58**	5,691.36	3	3.66	**0.89**
			WeightedGaussian	29.14	5,635.62	5	6.38	5.62
	Credit default	0.79	SHAP	4,838.49	7,456.88	5	8.89	8.69
			SHAP-Ensemble	1,146.68	5,506.02	3	4.83	58.85
			SHAP+FIRES	2,975.42	5,800.03	5	10.72	8.88
			LIME	411.57	**5,457.68**	3	**11.17**	10.01
			LIME-Ensemble [c]	/	/	/	/	/
			LIME+FIRES	298.27	5,662.37	5	8.42	10.30
			FIRES	**31.53**	5,541.34	5	9.25	**0.72**
			WeightedGaussian	250.03	6,778.04	**6**	8.43	5.49

[a] Explained classifier accuracy on last 250 samples.
[b] ms per explanation
[c] Did not produce any explanations
[l] Lower is better
[h] Higher is better

REFERENCES

[1] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell. "What do we need to build explainable AI systems for the medical domain?" In: *arXiv preprint arXiv:1712.09923* (2017).

[2] N. Burkart and M. F. Huber. "A Survey on the Explainability of Supervised Machine Learning". In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 245–317. DOI: 10.1613/jair.1.12228.

[3] A. Paleyes, R. Urma, and N. D. Lawrence. "Challenges in Deploying Machine Learning: a Survey of Case Studies". In: *CoRR* abs/2011.09926 (2020). arXiv: 2011.09926.

[4] S. Lundberg and S.-I. Lee. "A unified approach to interpreting model predictions". In: *arXiv preprint arXiv:1705.07874* (2017).

[5] M. Ribeiro, S. Singh, and C. Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. Association for Computational Linguistics, 2016. DOI: 10.18653/v1/n16-3020.

[6] G. Hinton and N. Frosst. "Distilling a Neural Network Into a Soft Decision Tree". In: 2017. URL: https://arxiv.org/pdf/1711.09784.pdf.

[7] J. Haug, M. Pawelczyk, K. Broelemann, and G. Kasneci. "Leveraging Model Inherent Variable Importance for Stable Online Feature Selection". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2020. DOI: 10.1145/3394486.3403200.

[8] C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006. URL: https://gaussianprocess.org/gpml/.

[9] J. Montiel, J. Read, A. Bifet, and T. Abdessalem. "Scikit-Multiflow: A Multi-output Streaming Framework". In: *Journal of Machine Learning Research* 19.72 (2018), pp. 1–5. URL: http://jmlr.org/papers/v19/18-251.html.

[10] D. Dua and C. Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[11] I.-C. Yeh and C. hui Lien. "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients". In: *Expert Systems with Applications* 36.2 (2009), pp. 2473–2480. DOI: 10.1016/j.eswa.2007.12.020.