Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Reject Options for Incremental Regression Scenarios

Jonathan Jakob, Martina Hasenjäger, Barbara Hammer

2022

Preprint:

This is a post-peer-review, pre-copyedit version of an article published in International Conference on Artificial Neural Networks (ICANN) 2022. The final authenticated version is available online at:
https://doi.org/10.1007/978-3-031-15937-4_21

Reject Options for Incremental Regression Scenarios

Jonathan Jakob[1(✉)], Martina Hasenjäger[2], and Barbara Hammer[1]

[1] Bielefeld University, Bielefeld, Germany
jjakob@techfak.uni-bielefeld.de
[2] Honda Research Institute, Offenbach, Germany

Abstract. Machine learning with a reject option is the empowerment of an algorithm to abstain from prediction when the outcome is likely to be inaccurate. Although the topic has been investigated in the literature already some time ago, it has not lost any of its relevance as machine learning models are increasingly delivered to the market. At present, most work on reject strategies addresses classification tasks. Moreover, the majority of approaches deals with classical batch learning scenarios. In this publication, in contrast, we study the important problem of reject options for incremental online regression tasks. We propose different strategies to model this problem and evaluate different approaches, both in a theoretical and a real world setting from the domain of human motion prediction; from the methods which we evaluate, a clear winner emerges as regards accuracy and efficiency.

Keywords: Reject option · Incremental learning · Online regression

1 Introduction

In machine learning a reject option is the ability of an algorithm to abstain from prediction when the outcome is likely to be inaccurate [8]. Classification with reject option has already been studied many decades ago [3], and it has been proven that certainty-based rejection offers an optimal strategy, provided the underlying probability distributions are known. In practice, this strategy is usually approximated based on suitable surrogates, since the true probability distribution is not known [2,6,9]. However, most reject option applications from the literature have been proposed for classification tasks.

In this contribution, we are interested in reject options for regression tasks such as arise in online time series prediction. Only a few reject option systems for regression tasks have been published recently. In the work [12] the authors propose a new uncertainty function for regression called *Blend-Var*. The approach tackles the rejection problem from a risk-coverage point of view and measures the variance of multiple predictions on an input image that was rotated, reflected or shifted. [7] introduces *SelectiveNet*, an approach where a deep neural network with two separated heads (one for prediction and one for rejection) is trained

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
E. Pimenidis et al. (Eds.): ICANN 2022, LNCS 13532, pp. 248–259, 2022.
https://doi.org/10.1007/978-3-031-15937-4_21

end-to-end in a single model. The work [1] presents a neural framework, that is based on a generalized meta-loss function. It revolves around the simultaneous training of two neural networks, one for prediction and the other for rejection. Finally, [4] proposes a reject scheme which targets general prescription methods based on an extended cost functions, incorporating costs for rejection, and controls the reject strategy by an explicit adaptive output value, indicating whether the current input should be rejected. Here, regression or classification prescriptions are trained simultaneously to the reject indicator output.

However, all of these approaches deal with reject options in an offline setting; i.e., training data are given prior to training, and the regression model together with the reject strategy can be trained based on these batch training data. We, on the other hand, are interested in reject options for incremental regression tasks. Incremental learning tasks learn from a stream of data which arrive continuously over time rather than a priorly available batch of training data [16]. This is an important scenario e.g. in the context of lifelong learning, where models need to be continuously adapted to a possibly changing environment, or product personalization, where a smart device needs to be adapted to the specific demands of a user to enable full functionality.

In the specific setup which we consider in this contribution, the motivation stems from an application of learning schemes for an optimization of control of exoskeletons. Modern exoskeleton robots utilize machine learning to facilitate the prediction of upcoming movements in order to provide adequate support for the user. Hereby, incremental algorithms can be of great help because they can automatically adapt to new movement patterns [17,23]. However, the adaption usually takes some time. In such settings, no support in a smooth movement is better than applying the wrong support; hence it can be beneficial to realize incremental learning with reject option in places that outcasts those samples of a novel movement pattern that are highly afflicted by errors in the prediction forecast. Therefore, in this contribution, we investigate how reject options can be facilitated that work next to an incremental algorithm and ideally reject only the initiate samples of new concepts in the data stream so that the underlying regressor has time to adapt itself. We propose different approaches for this task and evaluate the different methods in a scenario which incorporates movement patterns measured in a realistic environment and with different individuals. We will demonstrate that one modeling in particular shows very promising results.

This paper is structured as follows: The next section explains the problem setting. Afterwards, we introduce the different rejection approaches that we compare. Then, the design of our experiments is revealed and subsequently the results are presented. Finally, the last section concludes the paper.

2 Problem Setting

We use incremental regression to predict a data stream one instance after another. Hereby, we define a data stream $S = \{s_1, s_2, s_3, ..., s_t\}$ as a potentially infinite set of data points $s_i \in \mathbb{R}^n$. The task is predicting the vector s_{t+1} from previous instances of the stream. An incremental model is an algorithm that receives a data

stream instance after instance and instantaneously generates a sequence of models $h_1, h_2, h_3, ..., h_t$ which are used for next step prediction, i.e. $h_{i-1}(s_i)$ should approximate the value s_{i+1} based on the function h_{i-1} that acts on the current instance and predicts the value of the next instance of the data stream. After that, the true value s_{i+1} is revealed and a new model h_i is learned. To evaluate this regression task, the *Interleaved train test error (ITTE)* is applied:

$$E(S) = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(h_{i-1}(s_i) - s_{i+1})^2}$$

ITTE measures the *Root Mean Squared Error (RMSE)* over every model h_i up to time t. By $E(s_t)$ we refer to the sequence of errors rather than the average.

Furthermore, a *Reject Option* in an incremental learning scheme is a function $r(s_i, E(s_i)) \rightarrow \{0, 1\}$, that acts on the current input data or the local error of the model h_{i-1}, i.e. a subpart or summary statistics of $E(s_i)$, and produces an indicator that governs whether the current sample is rejected or not. To evaluate the rejection function, we can refer to the ITTE over all non rejected samples. This should be compared to the percentage of points which are rejected, since these two quantities typically form a Pareto front: good reject strategies should result in an improved remaining ITTE the more samples are rejected.

3 Rejection Models

We evaluate four different approaches for a Reject Option in incremental regression. All of these approaches are agnostic to the underlying regression algorithm and can be used in combination with any online regression model. Based on the findings in [11], we use a simple kNN regressor to conduct our experiments, since it displayed competitive and particularly robust results in that work.

3.1 Drift Rejection

The first rejection is based on drift detection. It monitors the local errors of the underlying regression model on the incoming data stream and applies a standard drift detection algorithm to detect changes in those error values, using any drift detection technology [10]. The reasoning behind this is, that when changes in the error occur, e.g. the error increases, the algorithm does not perform well in this area of the data stream and therefore samples should be rejected until the algorithm has learned to deal with the current concept. Here, any drift detection method from the literature can be used. We opted for the Page-Hinkley drift detector [21] and its implementation from the online library River [18]. To determine, how long samples should be rejected once drift has been detected, we use a simple strategy, that monitors the overall mean error on the data stream, and compares it to the rolling mean of the last n samples. When the rolling mean falls under the long term mean, we stop the rejection and the drift detector is activated again. The hyperparameters that can be adjusted in this approach are the threshold for drift detection of the Page-Hinkley algorithm and a value α, that scales the long term mean used to determine the end of a rejection phase.

3.2 Local Outlier Probabilities Rejector

The next two approaches are based on Local Outlier Probabilities (LoOP) [13]. LoOP is a local density based outlier detection method that provides outlier scores in the range [0,1]. We use LoOP for two different rejection approaches.

LoOP Data. In the first approach we apply LoOP to the input data. Unlike drift-based rejection, which targets critical time steps, this approach targets points in space which are unknown by the current model. When a new concept manifests itself in the stream, LoOP should assing high outlier probabilities to these input samples. Since, the underlying incremental regressor has not learned the new concept yet, these samples should be rejected until enoug of them have been processed, rendering them not outliers anymore.

LoOP Error. The second approach, like the drift rejector, applies LoOP to the error stream generated by the underlying regressor. The idea is that unknown error profiles indicate regions of high insecurity of the process where prediction should be rejected. If the error values increase, they become outliers and should be labeled as such by higher outlier probabilities. So, in theory, samples with high error values are rejected until the error decreases into normal ranges again.

In both variants, the hyperparameter that governs the rejection rate is a cutoff probability β that defines the threshold from which to reject samples.

3.3 Baseline Rejection

The last rejection strategy is a simple baseline approach. It monitors the long term mean error over the whole data stream and rejects all samples that exhibit a local error higher than the mean. Hence rejection takes place based on the recent observed error. Here, the hyperparameter is a value γ that scales the long term mean in order to facilitate various rejection rates.

4 Experiments

As stated previously, we use a kNN regression model which incrementally acts on time windows as underlying model. We use two distinct types of data sets. Theoretical benchmark data with known ground truth and data from a real world scenario from the domain of online human motion prediction.

4.1 Chaotic Time Series Data

For the theoretical part, we create special benchmark data sets from chaotic systems. We use the Lorenz system [14] along with the Roessler system [22] as well as the Tinkerbell map [20] along with the Duffing map [5] to create four new benchmark data sets. Hereby, the Lorenz and Roessler systems are three

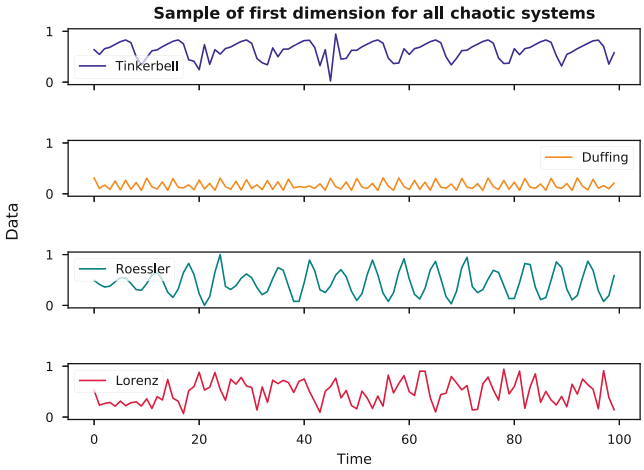

Fig. 1. Example plot of the first 100 data points in the first dimension for all chaotic systems. The first row shows the Tinkerbell map, the second row shows the Duffing map, the third row shows the Roessler system, the last row shows the Lorenz system.

dimensional, while the Tinkerbell and Duffing map are two dimensional data streams. Figure 1 shows the expansion of the first 100 data points in a stream created from each of the chaotic systems. Our benchmark data sets are created from the raw streams in the following way. Data streams holding 2000 instances of the three dimensional Lorenz and Roessler systems are glued together in alternate fashion, to form a data set with two sudden change points of which the latter initiates the reoccurring of the first system for a second time. The same is done for the two dimensional Tinkerbell and Duffing maps, leading to four data sets of 6000 data points each. Figure 2 shows a sample plot of the Roessler-Lorenz data set.

4.2 Real World Data

For the real world part, we use data from the NEWBEE database [15]. This data was created at the Honda Research Institute Europe and comprises human gait recordings from 20 participants that completed three different walk courses with various terrains, such as stairs, slopes and level walk. The data was recorded using full body xsens suits as well as insole trackers. We use a subset of this database comprising of four different persons on the course-A track, where we only use the lower body features of acceleration and angular momentum in three directions. This leads to an input space of 42 features.

4.3 RMSE-Reject Curves

We evaluate and compare the different rejection approaches by means of Root Mean Squared Error (RMSE) - reject curves [19]. This means, that we compute

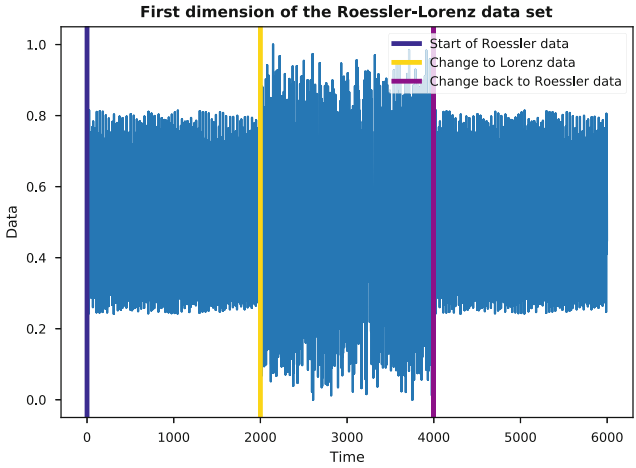

Fig. 2. Example plot of the first dimension of the Roessler-Lorenz data set. The first 2000 samples are taken from the Roessler system, followed by another 2000 points from the Lorenz system. In the end another change point back to the Roessler system is induced.

the rejected samples for a wide range of possible hyperparameters in order to acquire runs with different rates of rejection. Then, we plot the RMSE of the non rejected samples per rejection rate and evaluate the algorithms by comparing the curves. Different RMSE-reject curves for a given data set all start at the same point in the plot. This point is given by the RMSE of the underlying regressor on the total data set, i.e. on a run with 0% rejection. As the rejection rate increases, the RMSE on the non rejected data samples drops, until it reaches 0, for a 100% rejection. This means, that algorithms with lower curves outperform those with higher curves because they exhibit a lower RMSE for the same rejection rates.

4.4 Chaotic Data Experiment

In the first experiment we evaluate the four rejection approaches on the chaotic data sets. It is important to note, that all sets have different internal complexities, i.e. they exhibit different levels of difficulty for the underlying regressor. For the two dimensional systems, the Tinkerbell map is easier than the Duffing map, while for the three dimensional systems, the Roessler expansion is easier than the Lorenz expansion. Overall, the Lorenz system is the hardest to predict.

All chaotic data sets consist of two different chaotic expansions, one sandwiched in between the other. This means, that the underlying incremental regressor first learns one system and then suddenly has to switch to learning the next system. This clear and sudden switch creates a situation that is very suitable for the deployment of a reject option. Ideally, a rejection approach should reject the initial samples of the new system because the underlying regressor needs time

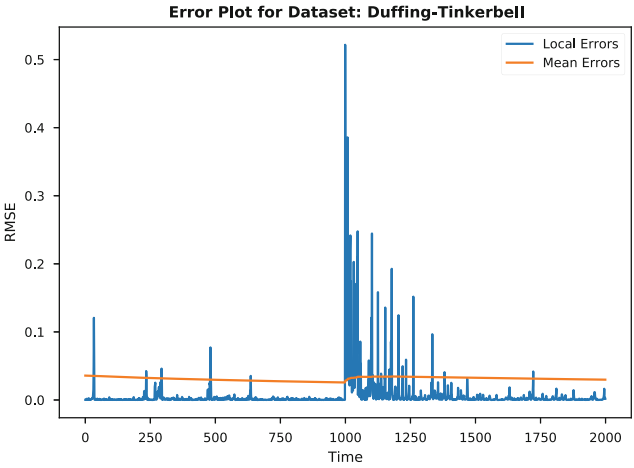

Fig. 3. Error plot for the chaotic experiment on the Duffing-Tinkerbell data set.

to adequately learn it and return accurate predictions. This, then should lead to a reduction in RMSE compared to a setting without rejection.

An example of the error progression of the underlying regressor for the Duffing-Tinkerbell data set is given in Fig. 3. We pre-train the regressor on 1000 samples and plot the local errors and accumulating mean errors on the subsequent 2000 samples. Hence the onset of the Tinkerbell expansion is located at time step $t = 1000$. One can observe, that the local errors increase drastically here and then gradually decrease as the regressor adapts to the new system.

4.5 Real World Data Experiment

In the second experiment we evaluate the different approaches in a real world setting. Here, data is complex and exhibits a high degree of noise. This leads to a scenario, where the local errors fluctuate strongly over the whole series.

An example is shown in Fig. 4. It shows the error progression for the NEW-BEE CourseA Person 1 data set in the same fashion as Fig. 3. One can observe, that changes in the underlying pattern, given for example at time steps $t = 300$ and $t = 1300$, are not leading to dramatic error spikes as in the chaotic data. Instead, the behaviour of the error values is a lot more coherent, which renders it harder to work on with error based rejection approaches.

5 Results

In this section, we first report the results for both experiments by means of RMSE-reject curves. Afterwards, we evaluate the effect of different rejection rates as the relative reduction of the RMSE compared to a scenario without rejection in tabular form.

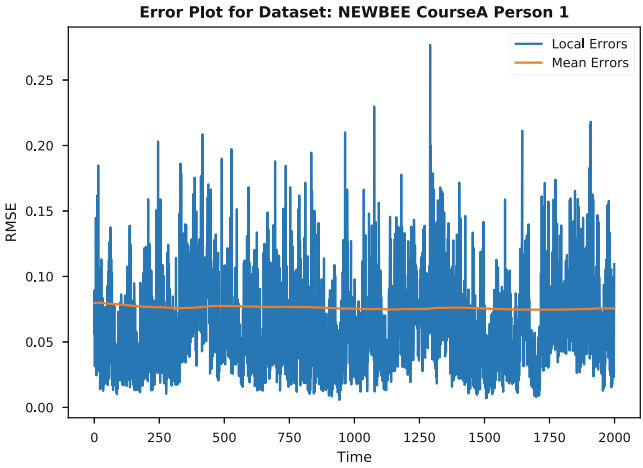

Fig. 4. Error plot for the real world experiment on the NEWBEE CourseA Person 1 data set.

5.1 Chaotic Data Results

The results for the chaotic data experiment are visualized in Fig. 5. The plots show the RMSE-reject curves of all rejection approaches for all chaotic data sets. The first thing to note is, that both LoOP approaches do not seem to work very well, meaning, they do not reduce the error substantially even when the rejection rate increases dramatically. In the case of LoOP Error this can be explained by the fact, that the error values do not spread out over a large space. Therefore, only the first few high errors are classified as outliers while the then slowly diminishing error values form their own group, and thus are not perceived as outliers anymore. The LoOP Data approach on the other hand does not work well because the different chaotic expansions do not seem to be very far apart in input space.

The Page-Hinkley rejector is the best performing approach on three out of the four data sets and tied for first place on the remaining one.

The baseline works rather well, but only on the Tinkerbell-Duffing and the Roessler-Lorenz data sets. These are those sets, where the easier chaotic expansion is intercepted by the harder one. Instead, when the harder expansion comes first, the baseline does not work well because it tracks the mean error over the whole sequence and therefore is faced with a gradually reducing error in the interception part, leading to none adequate rejections.

Another observation to note is the behaviour of the approaches with regard to granularity. Hereby, the baseline shows the most agile behaviour, meaning that it is possible to tweak its hyperparameters in such a way as to enable the realization of a wide array of rejection rates. The Page-Hinkley approach on the other hand, does not come with such a fine granularity. Due to its reliance on active drift detection it can only realize larger spaced rejection rates. Similarly,

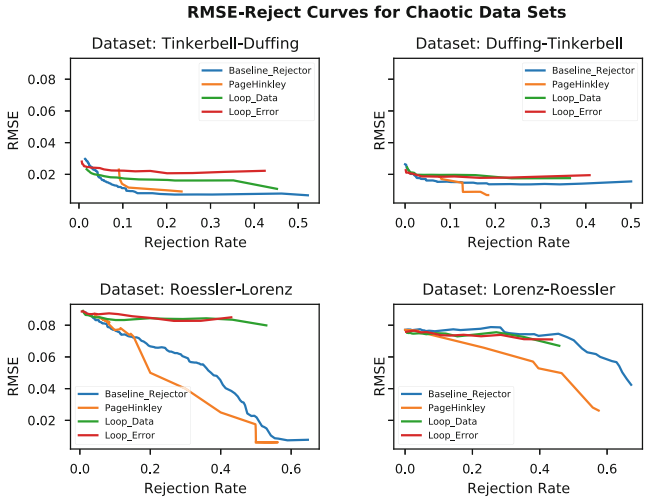

Fig. 5. RMSE-Rejection Curves for all rejection methods on the four chaotic data sets.

the LoOP approaches can not match the baseline in terms of agility with regard to the space of rejection rates.

5.2 Real World Data Results

The RMSE-reject curves of the real world data experiment are visualized in Fig. 6. Again, and for the same reason as in the previous experiment, the LoOP error approach does not yield good results. Interestingly, the LoOP Data rejector now becomes the second best performing system, tying for first place on two out of the four data sets and coming in second on the remaining ones. This can be explained by the much more complex input space, where clear differences of the samples can now manifest themselves.

Same as in the previous experiment, the Page-Hinkley rejector is the best performing approach, although it is tied for first place with the LoOP Data system on two out of four data sets.

The baseline shows more ambiguous results. It is the worst performer on one data set but matches the performance of LoOP Data in another one. For the remaining ones it is clear, that it does not match LoOP Data and Page-Hinkley.

5.3 Tabular Evaluation

Here, we evaluate the previous experiments by means of the reduction in RMSE for different rejection rates compared to a scenario without rejection. The results are listed in Table 1. This table shows the mean reduction of RMSE on the chaotic and real world data sets for four distinct rejection rates. As observed previously, the Page-Hinkley approach wins outright in all categories. On the

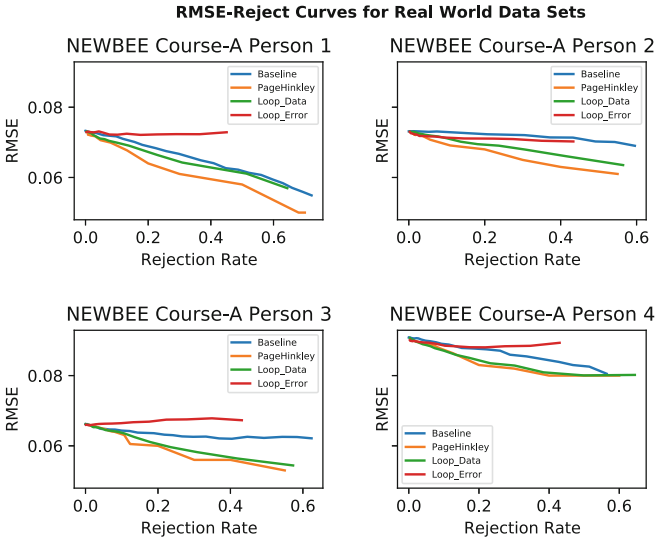

Fig. 6. RMSE-Rejection Curves for all rejection methods on the four real world data sets.

chaotic data sets, it is followed by the baseline, while the LoOP systems come in last. However, on the real world data, the results of the baseline diminish and it gets overtaken by the LoOP Data approach.

Furthermore, one can observe, that the rejection works much better on the chaotic data sets than on the real world setting. This is probably due to the fact, that those sets are a lot less complex and less noisy than the real world data sets. However, the Page-Hinkley approach still manages a reduction between 10% and 20% on various rejection rates. In our opinion, this is a very promising result with high relevance in practice.

Table 1. Relative reduction in RMSE (in %) for different rejection rates. All values have been averaged over all chaotic and real world data sets respectively.

Rejectors	Data sets and rejection rates							
	Chaotic data				Real world data			
	20%	30%	40%	50%	20%	30%	40%	50%
LoOP Error	22.23	20.53	20.28	20.28	1.78	2.12	2.12	2.12
LoOP Data	24.56	25.94	28.52	33.66	7.55	10.22	12.89	14.65
Page-Hinkley	52.24	58.08	64.85	73.38	10.39	14.75	16.92	19.79
Baseline	40.00	42.31	47.47	54.70	4.21	5.83	7.51	9.05

6 Conclusion

In this contribution we investigated the problem of reject options for online regression tasks. Out of the four models that we compared one clear winner emerged. The Page-Hinkley approach works very well on easy data with clear rejection conditions but it also delivers adequate results in a more messy real world environment. The baseline is the cheapest version to apply but it can only deliver good results when the circumstances are appropriate. LoOP Data works better when the input space is more complex and a clear discrimination of input samples is possible. Finally, LoOP Error is an approach that is not suitable for the rejection problem because it does not yield better RMSEs for rising rejection rates, meaning its rejections are sub par.

References

1. Asif, A.: Generalized neural framework for learning with rejection. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2020). https://doi.org/10.1109/IJCNN48605.2020.9206612
2. Bartlett, P.L., Wegkamp, M.H.: Classification with a reject option using a hinge loss. J. Mach. Learn. Res. **9**, 1823–1840 (2008). https://dl.acm.org/citation.cfm?id=1442792
3. Chow, C.: On optimum error and reject trade-off. IEEE Trans. Inf. Theory **16**, 41–46 (1970)
4. Denis, C., Hebiri, M., Zaoui, A.: Regression with reject option and application to KNN (2021)
5. Duffing, G., Emde, F.: Erzwungene schwingungen bei veränderlicher eigenfrequenz und ihre technische bedeutung
6. Fischer, L., Hammer, B., Wersing, H.: Optimal local rejection for classifiers. Neurocomputing **214**, 445–457 (2016). https://doi.org/10.1016/j.neucom.2016.06.038
7. Geifman, Y., El-Yaniv, R.: SelectiveNet: a deep neural network with an integrated reject option (2019)
8. Hendrickx, K., Perini, L., der Plas, D.V., Meert, W., Davis, J.: Machine learning with a reject option: a survey. CoRR abs/2107.11277 (2021). https://arxiv.org/abs/2107.11277
9. Herbei, R., Wegkamp, M.H.: Classification with reject option. Can. J. Stat./La Revue Canadienne de Statistique **34**(4), 709–721 (2006). https://www.jstor.org/stable/20445230
10. Hinder, F., Artelt, A., Hammer, B.: Towards non-parametric drift detection via dynamic adapting window independence drift detection (DAWIDD). In: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13–18 July 2020, Virtual Event. Proceedings of Machine Learning Research, vol. 119, pp. 4249–4259. PMLR (2020). https://proceedings.mlr.press/v119/hinder20a.html
11. Jakob, J., Hasenjäger, M., Hammer, B.: On the suitability of incremental learning for regression tasks in exoskeleton control. In: IEEE Symposium on Computational Intelligence in Data Mining (CIDM). IEEE, December 2021
12. Jiang, W., Zhao, Y., Wang, Z.: Risk-controlled selective prediction for regression deep neural network models. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2020). https://doi.org/10.1109/IJCNN48605.2020.9207676

13. Kriegel, H., Kröger, P., Schubert, E., Zimek, A.: Loop: local outlier probabilities (2009)
14. Lorenz, E.: Deterministic nonperiodic flow. Journal of the atmospheric sciences 20, 130–41.1. Prog. Phys. Geogr. Earth Environ. **32**(4), 475–480 (2008). https://doi.org/10.1177/0309133308091948
15. Losing, V., Hasenjaeger, M.: NEWBEE: a multi-modal gait database of natural everyday-walk in an urban environment. Sci. Data (2022, submitted)
16. Losing, V., Hammer, B., Wersing, H.: Incremental on-line learning: a review and comparison of state of the art algorithms. Neurocomputing **275**, 1261–1274 (2018). https://doi.org/10.1016/j.neucom.2017.06.084
17. Losing, V., Yoshikawa, T., Hasenjäger, M., Hammer, B., Wersing, H.: Personalized online learning of whole-body motion classes using multiple inertial measurement units. In: International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, 20–24 May 2019, pp. 9530–9536. IEEE (2019). https://doi.org/10.1109/ICRA.2019.8794251
18. Montiel, J., et al.: River: machine learning for streaming data in python. J. Mach. Learn. Res. **22**(110), 1–8 (2021). https://jmlr.org/papers/v22/20-1380.html
19. Nadeem, M.S.A., Zucker, J.D., Hanczar, B.: Accuracy-rejection curves (ARCS) for comparing classification methods with a reject option. In: Machine Learning in Systems Biology, pp. 65–81. PMLR (2009)
20. Nusse, H., Yorke, J.: Dynamics: Numerical Explorations. Springer, New York (1997)
21. Page, E.S.: Continuous inspection schemes. Biometrika **41**(1–2), 100–115 (1954). https://doi.org/10.1093/biomet/41.1-2.100
22. Rössler, O.: An equation for continuous chaos. Phys. Lett. A **57**(5), 397–398 (1976). https://doi.org/10.1016/0375-9601(76)90101-8. https://www.sciencedirect.com/science/article/pii/0375960176901018
23. Tijjani, I., Kumar, S., Boukheddimi, M.: A survey on design and control of lower extremity exoskeletons for bipedal walking. Appl. Sci. **12**(5) (2022). https://doi.org/10.3390/app12052395. https://www.mdpi.com/2076-3417/12/5/2395