Adaptive Approaches towards Prediction Interval for Data Stream Regression *

*Yibin Sun[1][0000−0002−8325−1889], Bernhard Pfahringer[1][0000−0002−3732−5787], Heitor Murilo Gomes[1,2][0000−0002−5276−637X], and Albert Bifet[1,3][0000−0002−8339−7773]

[1] AI Institute, The University of Waikato, Hamilton, New Zealand
[2] School of Engineering and Computer Science, Victoria University of Wellington, Wellington, New Zealand
[3] LTCI, Télécom Paris, IP Paris, France
*ys388@students.waikato.ac.nz, {bernhard, abifet}@waikato.ac.nz, heitor.gomes@vuw.ac.nz

Abstract. Prediction Interval (PI) is a powerful technique for quantifying the uncertainty of regression tasks. However, research on PI for data streams has not received much attention. Moreover, traditional PI-generating approaches are not directly applicable due to the dynamic and evolving nature of data streams. This paper presents `AdaPI` (ADAptive Prediction Interval), a novel method that can automatically adjust the interval width by an appropriate amount according to historical information to converge the coverage to a user-defined percentage. `AdaPI` can be applied to any streaming PI technique as a postprocessing step. This paper develops an incremental variant of the pervasive Mean and Variance Estimation (MVE) method for use with `AdaPI`. An empirical evaluation over a set of standard streaming regression tasks demonstrates `AdaPI`'s ability to generate compact prediction intervals with a coverage close to the desired level, outperforming alternative methods.

Keywords: Data streams · Regression · Prediction Intervals.

1 Introduction

The machine learning literature has thoroughly investigated learning algorithms for regression tasks [7]. However, a single-valued prediction is insufficient for many real-world applications [10]. A prediction interval (PI) [9] provides more informativeness and uncertainty measurements [16] to a regression model since it generates intervals that encompass the expected range of true values with a desired confidence level. Learning data streams is different from conventional machine learning tasks. Due to the potentially infinite amount of data, stream learning algorithms can neither store all the previous information nor iterate multiple times through the datasets. The basic assumption is that data points

* I would like to acknowledge the support from TAIAO project.

from a stream can only be inspected once times [2]. Several techniques for establishing prediction intervals were introduced based on different mathematical theories since the seminal technique in [9]. Zhao et al in [20] summarised the most commonly used ones, such as bootstrapping techniques and delta method.

Current research rarely focuses on prediction intervals on data streams. The few PI methods available for data streams (or time series) rely on windowed versions of existing techniques, which do not agree with current state of the art methods for data stream regression [17], which are fully incremental (as discussed in Section 2). The lack of a fully incremental PI methods motivated our work. Our main contribution is the application of the Mean and Variance Estimation (MVE) prediction interval method in a streaming fashion and the ADAptive Prediction Interval (`AdaPI`) methodology, which can be applied to any PI method as a post-calibration step. `AdaPI` incrementally adjusts the generated interval widths according to the current coverage, ensuring that the coverage converges towards the desired confidence level. This new approach also adapts automatically to concept drifts, a critical issue in the streaming scenario [2].

2 Related Work

Xu and Xie [19] introduced a Bootstrap-based PI methodology for dynamic time series. This methodology establishes several base learners $\mathcal{A}$ and an error-based aggregation function ϵ. $\mathcal{A}$ and ϵ are updated by re-sampling the time series with replacement in batches. Subsequently, quantiles of $\mathcal{A}$ and ϵ are utilised for providing PIs, i.e. $[\mathcal{A}^{1-\alpha} - \epsilon^{1-\alpha}, \mathcal{A}^{1-\alpha} + \epsilon^{1-\alpha}]$.

IT2FGNNDEnsemble (Interval Type-2 Fuzzy Granular Neural Network Dynamic Ensemble) [13] comprises a seven-layer neural network, including a granularity layer, a normalisation layer, a layer that contains an ensemble of the basic algorithm – IT2FGNN, and so forth. In the ensemble, each IT2FGNN provides a prediction interval, which is fused into the final PI after an elimination process that removes "weak" ensemble members.

Jackknife+ [1] is a popular approach to prediction intervals, derived from Jackknife resampling. Jackknife obtains residuals (R^{LOO}) using a Leave-one-out strategy, and Jackknife+ modifies it by ignoring the current instance. Prediction intervals are produced by the α quantile of the given series, i.e. $[q^{-}_{n,\alpha}(\mathcal{A}_{-i} - R^{LOO}_i), q^{+}_{n,\alpha}(\mathcal{A}_{-i} + R^{LOO}_i)]$.

Inductive Conformal Prediction (ICP) is the common regressor within the Conformal Prediction framework [15]. ICP trains an ML model on a training set and computes a nonconformity score as a deviation measurement of an instance's behaviour. Prediction intervals for further examples are generated based on these scores and then evaluated on a calibration set. Split Conformal Prediction (SCP), an extension of ICP, divides the dataset into "splits", each of which provides an individual conformal score. The final PI is a combination of all "splits".

Hadjicharalambous et al. [8] introduced a neural network (NN)-based online prediction interval method BLM (Bootstrap-LUBE Method), which combines the basic Bootstrap technique and LUBE (Lower-Upper Bound Estima-

tion) [12] PI methods. Specifically, a number of NNs are trained to provide pseudo-measurements (dummy data points) in data-sparse regions to ensure sufficient information for the LUBE method to yield reasonable PIs. This method is adapted for data streams using windowing approaches. However, training NNs in a sliding window manner can be computationally expensive.

Both low efficiency and the need to process instances multiple times pose challenges for all the aforementioned PI methods. Recently, Zhao et al. [21] proposed a tree-based algorithm and an ensemble of tree-learners for constructing PI on time series and streaming data. They selected Conditional Inference Tree (ctree) as the base learner in their work, although all tree-based regressors are suitable. In the single tree approach, prediction intervals are determined by the quantile values stored in the leaves. The PIs are expanded by a predefined parameter α during the initialisation. If the current coverage falls outside the desired range, the PIs are adjusted using both the current RMSE and a β parameter (i.e.,$\beta\times$ RMSE). The ensemble approach also begins with a single tree. New trees are periodically appended until the ensemble size reaches its maximum. The n_{th} PI is a weighted combination of the n_{th} and $\mathrm{n\text{-}1}_{th}$ tree, i.e. $\bar{P}_n = \omega_1 \times P_{n-1} + \omega_2 \times P_n$. Generally, ω_2 is greater than ω_1, giving newer trees a larger impact.

3 Background

Mean and Variance Estimation (MVE) [14] is one of the most straightforward and widely applied approaches for PI. MVE assumes the predictive errors follow a Gaussian distribution (Normal Assumption) which results in a generalised linear model, and therefore leads to a prediction interval illustrated as in Equation 1:

$$\mathrm{PI} \ \in \left(\mathcal{Y} - G^{-1}(0,\gamma) \times \sigma_\epsilon, \mathcal{Y} + G^{-1}(0,\gamma) \times \sigma_\epsilon\right) \tag{1}$$

where $G^{-1}(0,\gamma)$ is the inverse Gaussian distribution function with 0 mean and γ probability, and $\mathcal{Y}$ is the prediction output. According to the normal assumption, $\mathcal{E} \sim \mathcal{N}(0,\sigma_\epsilon)$. Hence, when $\gamma = 95\%$, $G^{-1}(0,0.95) \approx 1.96$.

Objectively measuring the quality of PI is challenging as we need to consider both the width of the interval and its coverage. To achieve a fair and comprehensive evaluation, we consider two metrics used in previous studies [8], the Coverage and Normalised Mean Prediction Interval Width (NMPIW). Coverage represents the ratio of the ground truth falling within the predicted PI, i.e.:

$$\mathrm{Coverage} = \frac{1}{N}\sum_{i=1}^{N} I_i \tag{2}$$

where $I_i = 1$ if $y \in [\mathcal{P}_l, \mathcal{P}_u]$ and $I_i = 0$ otherwise; N is the total observation number. NMPIW is expressed in Equation 3:

$$\mathrm{NMPIW} = \frac{\frac{1}{N}\sum_{i=1}^{N}(\mathcal{P}_{u_i} - \mathcal{P}_{l_i})}{R} \tag{3}$$

where $\mathcal{P}_{u_i}$ and $\mathcal{P}_{l_i}$ are the i_{th} upper and lower bounds for the i_{th} observation, and R is the range of the target values observed. As implied in the equation, NMPIW represents the ratio of the average PI width to the range of the true values and therefore reflects the effectiveness of the PI.

Very wide intervals naturally achieve high coverage, therefore the goal for a well-performing PI method is to produce intervals covering more or less the desired percentage of predictions, while also being as narrow as possible, indicated by small NMPIW values.

4 Adaptive Prediction Interval(`AdaPI`)

`AdaPI` uses a coefficient for scaling the generated interval width. Precisely, expanding the MVE Equation 1, the modified upper and lower bound in `AdaPI` are defined as in Equation 4:

$$\texttt{AdaPI} \ = \left(\mathcal{Y} - \mathcal{S} \times G^{-1}(0,\gamma) \times \sigma_\epsilon, \mathcal{Y} + \mathcal{S} \times G^{-1}(0,\gamma) \times \sigma_\epsilon\right) \tag{4}$$

where the scalar $\mathcal{S}$ is specified by Equation 5:

$$\mathcal{S} = \begin{cases} 100 - \mathcal{C} & \text{if } \mathcal{C} < 2\mathcal{L} - 100 \\ \log_{\mathcal{L}} \frac{100-\mathcal{C}}{100-\mathcal{L}} + 1 & \text{if } \mathcal{C} \geq \mathcal{L} \\ (\mathcal{L} - 100) \log_{\mathcal{L}} \frac{100+\mathcal{C}-2\mathcal{L}}{100-\mathcal{L}} + 1 & \text{otherwise} \end{cases} \tag{5}$$

where $\mathcal{C}$ is the current coverage, and $\mathcal{L}$ represents the confidence level. This curve varies with different confidence levels $\mathcal{L}$. Figure 1 illustrates the curve graph of the scalar with a fixed confidence level at 95%, i.e. $\mathcal{L} = 95$.

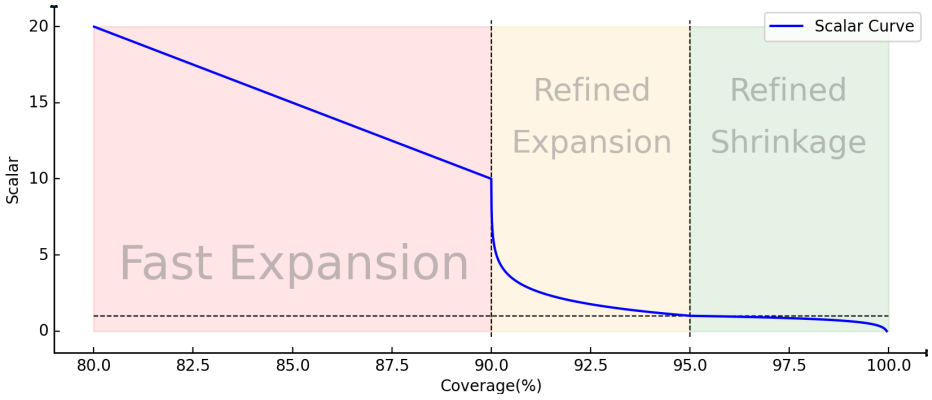


Fig. 1: Scalar Curve with a 95% Confidence Level ($\mathcal{C} \in [80, 100]$). Dotted lines denote significant values in the figure. The horizontal line is 1.0, the right vertical line is the desired confidence level (95%). The refined expansion and the refined shrinkage areas have same width.

Three segments can be observed in Figure 1. The red part (Fast expansion) denotes a fast expansion phase where the scalars are determined linearly according to the current coverage. The purpose of this segment is to rapidly increase the coverage towards the confidence level. The orange part (Refined expansion) also expands the PI by producing scalars larger than one. However, the scalars from this part are generated more subtly by a logarithmic function. As the coverage nears the confidence level, the rate of change in the scalar will decrease. The green part (Refined shrinkage), on the other hand, aims to shrink the PI when the coverage exceeds the confidence level. Similar to the orange part, the scalar will be moderately updated when the coverage is close to the confidence level. This mechanism prevents the scalar from being too radical.

`AdaPI` adjusts dynamically, varying based on requirements rather than following a set pattern. This approach offers several benefits: 1. Radical adjustments occur for substantial coverage-confidence disparities; 2. Modifications are moderate when coverage approaches the confidence level, preventing overreactions; and 3. Auto cessation at the desired confidence level is achieved without adding extra hyperparameters.

`AdaPI` introduces a scalar-based solution alongside MVE, aiming to address the normal assumption inherent in MVE. In practical scenarios, it's often inaccurate to presume that predictive errors conform to a Gaussian distribution. Consequently, predictive intervals relying on statistical Gaussian measures might fail to adequately encompass actual observations. This disparity results in the coverage deviating from the confidence level established in the MVE approach. When regression models demonstrate exceptional (or poor) accuracy in forecasting specific data streams, MVE tends to produce overly narrow (or wide) predictive intervals. `AdaPI` counters this by generating scalar adjustments to calibrate these intervals accordingly.

Moreover, `AdaPI` naturally adapts to concept drift as it tends to be accompanied by increases in RMSE, which leads to shifts in coverage. Consequently, these coverage shifts trigger the interval scaling process.

In our experiments we observed that the **expansion** phase continues until the instability caused by a concept drift ends. `AdaPI` remains in a widening state during the unstable phases to prevent a significant drop in Coverage. After the regression model detects and adapts to the new concept, `AdaPI` also switches back to the refined expansion state. For a detailed analysis, please refer to Section 5.2.

All our experiments include a warm-up phase of 100 examples, to allow for a reasonably robust MVE estimation before starting any adaptation of the interval width by `AdaPI`.

5 Experiments and Results

The experimental datasets are briefly illustrated in Table 1. Ailerons, Elevators, and House8L datasets can be found at OpenML. The MetroTraffic dataset can be found at UCI Machine Learning Repository, and HyperA is generated

Table 1: Datasets Overview

Synthetic			Real		
Datasets	$N_{Features}$	$N_{Instances}$	Datasets	$N_{Features}$	$N_{Instances}$
Ailerons	40	13750	Abalone	8	4977
Elevators	18	16599	Bike	12	17379
Fried	10	40768	House8L	8	22784
HyperA	10	500000	MetroTraffic	7	48204

by a stream generator in MOA [2]. The Abalone, Bike, and Fried datasets are presented in [18], [4], and [3], respectively.

5.1 Comparison to Interval Forecast

Interval Forecast (IF) and its ensemble version (EnsembleIF) are introduced in [21] (refer to Section 2). In this work, a fully incremental tree called Fast Incremental Model Tree with Drift Detection (FIMT-DD) [11] is used as a base-learner substitute in both IF and EnsembleIF. The maximum ensemble number is set to 100 in EnsembleIF, and the window length for updating the tree is set to 1000. The values of $\alpha = 2$ and $\beta = 2.5$ are selected, as they are the medians of the recommended value ranges in the original paper. Multiple attempts have been made on the weighted-sum strategy in EnsembleIF, and the combination that performed the best — $\omega_1 = \omega_2 = 0.5$ — is presented in the results.

We present the Coverage and NMPIW results for single IF, EnsembleIF, MVE, and `AdaPI` in Table 2. For IF and EnsembleIF, the adjustment mechanism is triggered only when the model's coverage is not within the range of 94% to 96% (i.e., Coverage $\notin [0.94, 0.96]$). Additionally, MVE and `AdaPI` share the ARF-Reg algorithm [6] as their base learner and 95% as their confidence level.

Table 2: Coverage(%) and NMPIW(%) for 95% Confidence Level

	IF		**EnsembleIF**		**MVE**		`AdaPI`	
	Coverage	NMPIW	Coverage	NMPIW	Coverage	NMPIW	Coverage	NMPIW
Bike	98.35	46.11	99.52	67.47	89.51	**29.31**	**94.00**	37.97
Ailerons	98.79	44.75	99.05	54.16	94.73	**29.32**	**95.12**	30.12
HyperA	94.00	26.85	96.55	35.94	94.25	**20.22**	**94.70**	20.72
Abalone	98.39	59.67	98.92	69.74	**95.06**	35.15	95.10	**34.56**
Elevators	97.79	38.24	98.89	57.18	95.31	31.46	**95.24**	**30.7**2
House8L	97.87	29.14	98.97	50.88	96.68	30.44	**96.29**	**28.46**
Fried	96.58	40.24	99.47	76.90	97.43	33.80	**96.26**	**31.34**
MetroTraffic	98.15	101.99	99.76	143.88	98.15	98.34	**96.48**	**90.14**

In the "Coverage" columns in Table 2, values closer to 95% coverage are highlighted in bold and in the "NMPIW" columns the smaller values are bold.

It is evident that the MVE-based methods generally outperform the IF-based methods. Both MVE and `AdaPI` consistently provide smaller NMPIW values and coverage closer to the confidence level across almost all datasets. For instance, on the HyperA dataset, IF achieved a coverage of 94.00% but had an NMPIW of 26.85%, whereas `AdaPI` achieved higher coverage with a more compact NMPIW.

Pair-wise Friedman tests [5] are applied to the results in Table 2 in two ways: 1. Differences $\mathcal{D}$ from the coverage to the confidence level, i.e., $\mathcal{D} = |\mathcal{C} - \mathcal{L}|$; and 2. NMPIW. MVE, as well as `AdaPI`, versus IF and EnsembleIF, all produce p-values smaller than 0.05. Thus, the null hypothesis – there are no differences between the two candidates in each comparison – is rejected, demonstrating the significance of the MVE and `AdaPI` when compared to IF-based methods.

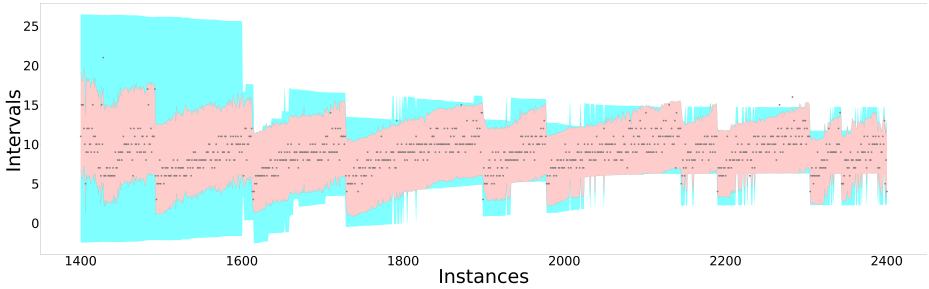


Fig. 2: `AdaPI`(95%) PI Area (Red, Narrow) and IF PI Area (Sky-Blue, Wide) on Abalone Dataset, and Ground Truths (Grey Dots).

Figure 2 plots the area covered by both `AdaPI` (in red) and IF (in sky-blue) on a subset of the Abalone dataset. It can be observed that `AdaPI` produces much narrower intervals yet covers almost the same amount of the data points as IF. Meanwhile, Figure 2 also underscores one of the advantages of a predictive model-based PI method over a quantile-based one in streaming scenarios: quantiles can be insensitive to changes. A streaming model learns from the data incrementally, while the quantiles in the leaves (or other structures) might remain relatively similar. `AdaPI` demonstrates a stronger ability to adapt to drifts (sudden shifts of the grey dots in Figure 2) faster than IF in Figure 2.

5.2 Comparison Between MVE and `AdaPI`

This section focuses on MVE and `AdaPI` results and analyses. Three streaming regression algorithms perform as background regression model in the experiments: Adaptive Random Forest for Regression (ARF-Reg) [6], Sliding Window K Nearest Neighbours (KNN), and Self-Optimising K Nearest Leaves (SOKNL) [17]. All the algorithms above are available in Massive Online Analysis (MOA) [2], a well-known open-source framework software for data streams. Our experiments use a "TestThenTrain" regime with default parameter settings in MOA.

Table 3: MVE vs. AdaPI for 95% Confidence Level (Closer Coverage in Bold)

	Algorithms	**ARF-Reg**		**KNN**		**SOKNL**	
	Metrics	Coverage	ΔNMPIW	Coverage	ΔNMPIW	Coverage	ΔNMPIW
Bike	AdaPI	**93.91**	28.6%↑	**93.70**	36.9%↑	**93.93**	27.7%↑
	MVE	89.50		88.16		89.15	
Ailerons	AdaPI	**94.99**	2.2%↑	**94.56**	2.1%↑	**95.06**	0.5%↑
	MVE	94.72		94.42		95.13	
HyperA	AdaPI	**94.70**	2.5%↑	**94.64**	1.0%↑	**94.75**	0.4%↑
	MVE	94.25		93.70		94.35	
Abalone	AdaPI	**94.98**	0.3%↓	95.24	0.8%↑	**95.06**	3.3%↓
	MVE	95.06		**94.98**		95.32	
Elevators	AdaPI	**95.08**	2.7%↓	94.86	1.9%↓	**95.28**	2.9%↓
	MVE	95.31		**95.03**		95.61	
House8L	AdaPI	**96.27**	6.6%↓	**95.35**	2.0%↓	**95.96**	5.9%↓
	MVE	96.67		95.47		96.45	
Fried	AdaPI	**96.24**	7.2%↓	**94.91**	1.0%↓	**96.17**	7.1%↓
	MVE	97.43		95.10		97.41	
MetroTraffic	AdaPI	**96.44**	8.4%↓	**95.72**	4.3%↓	**95.23**	0.5%↓
	MVE	98.14		96.88		95.39	

Table 3 shows the results regarding Coverage and NMPIW. Notably, the "ΔNMPIW" columns illustrate the rate of increase (or decrease) of AdaPI's average interval widths compared to MVE's, i.e. Rate $= \frac{|\text{AdaPI NMPIW - MVE NMPIW}|}{\text{MVE NMPIW}}\%$.

The discussion will primarily focus on ARF-Reg from this point onward since all three algirithms share similar tendency. It can be observed in the Bike dataset that MVE only achieves 89.5% coverage and AdaPI improves it to 93.91% at the cost of widening the interval width by almost 30%. Similar trends are observed in the Ailerons and HyperA datasets, albeit on a smaller scale. The last five datasets in Table 3 follow similar patterns where the coverage of MVE surpasses the confidence level and the AdaPI decreases them while shrinking the interval width. The coverage value of Abalone exceeds the desired confidence level rather than only approaching it. This circumstance can be explained by the basic (MVE) coverage being too close to the confidence level. We can reasonably assume that AdaPI's coverage fluctuates around 95% constantly during the process, hence the system switches between **shrink** and **expand** modes. Still, both methods perform more or less equally well for both the Ailerons and Abalone datasets, achieving very close to 95% coverage with very similar average interval width.

Figure 3 visualises the coverage results. It illustrates a clear picture that AdaPI is providing a closer-to-95% value than the regular MVE in almost all cases since the green area covers most markers. The exceptions can only be found around 95% coverage values, which are caused by the repeatedly switching between shrinking and expanding for AdaPI. Noticeably, most of the markers are reasonably far from the green area's edges, which indicates the significant improvements of AdaPI with respect to coverage.

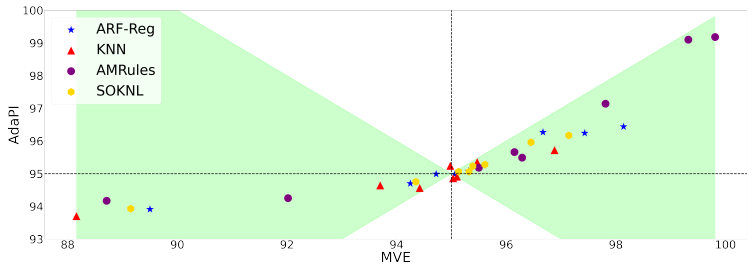


Fig. 3: AdaPI versus MVE coverage. Points inside the green area are closer to the 95% target for AdaPI than for MVE, i.e. AdaPI outperforms MVE in these cases.

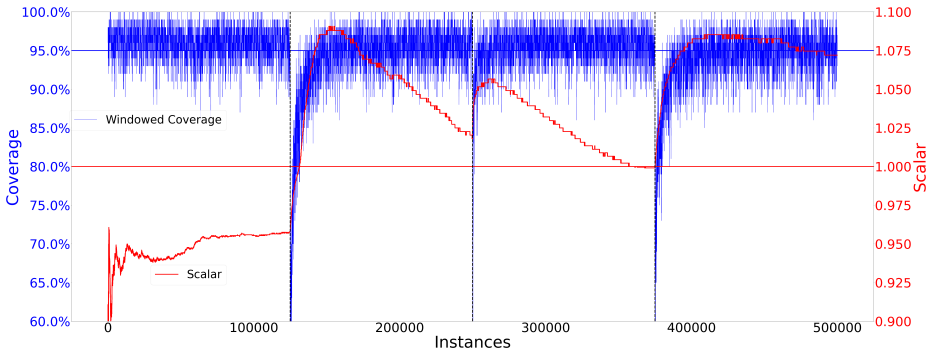
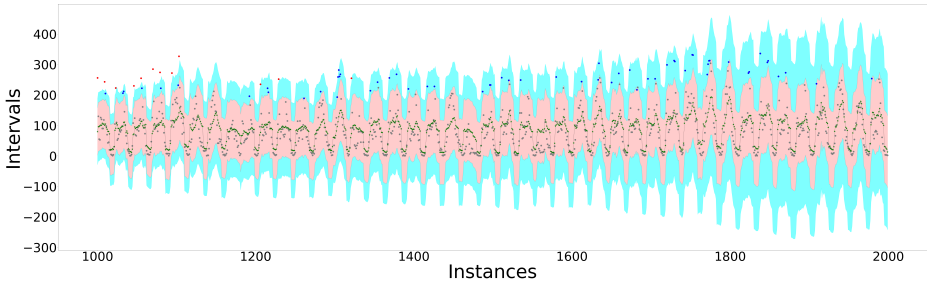


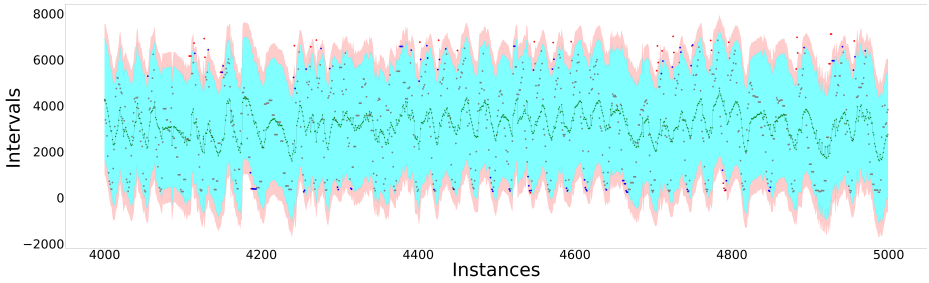
Fig. 4: Scalar (Red) and the Coverage (Blue) for ARF-Reg on HyperA Over Time. The black vertical lines highlight positions of concept drifts; the red horizontal line marks the value 1.0; and the blue horizontal line emphasises 95% confidence level.

Figure 4 demonstrates how the scalar (red line) and windowed coverage (blue line) behave over time for the ARF-Reg algorithm on the HyperA dataset. Windowed coverage is chosen as it responds faster to change, presenting a clearer picture of the dynamics of adaptation. Figure 4 demonstrates AdaPI's robustness in handling concept drifts. The HyperA dataset involves predicting the distance from a random point to a high-dimensional hyperplane. It comprises 500k instances and experiences three abrupt concept drifts at 125k, 250k, and 375k instances, aligning with the sudden shifts in the figure. The graph illustrates how ARF-Reg initially establishes a suitable model for HyperA, resulting in desirable coverage and moderate scalar. Then, the model encounters the first drift and no longer aligns with the new concept, leading to a decline in coverage. AdaPI responds by increasing the scalars, as indicated by the sharp rise in the red line. The scalars eventually stabilise closer to one once the model adapts to the new concept. Similar trends occur at the other two drift points.

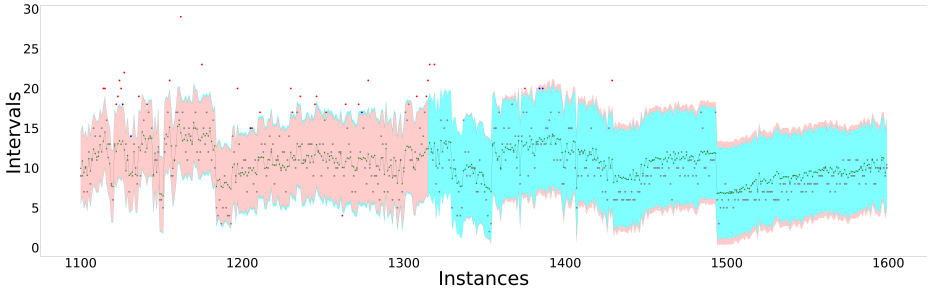
In Figure 5, three images depict behavioral visualizations for MVE and AdaPI. The MVE-covered area is highlighted in red, while AdaPI's area is filled in blue. Grey dots represent the ground truths covered by both methods, blue



(a) Bike data: `AdaPI` constantly larger than MVE



(b) MetroTraffic data: `AdaPI` constantly smaller than MVE



(c) Abalone data: `AdaPI` first larger, then smaller than MVE

Fig. 5: MVE Area (Red) and `AdaPI` Area (Sky-Blue), Narrower Area Covered Ground Truths (Grey Dots), Wider Area Covered Ground Truths (Blue Dots) and the Ground Truths outside Both Area (Red Dots).

dots indicate ground truths missed by the narrower approach but covered by the wider approach, and red dots are outside of both areas. The thin line in the middle represents the predictions made by ARF-Reg.

Figure 5a is an explicit annotation of the Bike dataset in Table 3. Apparently, numerous blue dots are included within `AdaPI`'s boundaries, even though they are excluded by MVE. This is the main reason why `AdaPI` achieves coverage much closer to 95%. It is worth noting that this enhancement is solely attributed to the scalar system as the MVE and `AdaPI` share the same predictive model.

Figure 5b presents a similar visualisation to Figure 5a but on the MetroTraffic dataset. MVE coverage in Table 3 is beyond 95% on MetroTraffic. Ergo, the chosen scalars are usually under 1, resulting in a more narrow `AdaPI` area. We can also conclude that by eliminating all blue dots out of the PI range, `AdaPI` shortens the interval width from a general perspective while the coverage value move toward the confidence level.

Figure 5c illustrates very well the adaptability of the `AdaPI` approach. Similar to Figures 5a and 5b, it showcases a subset of the Abalone dataset. As analysed in Table 3, the results of the Abalone dataset display erratic behaviour, which we attribute to fluctuations in adaptation. In Figure 5c, we pinpoint a moment when the coverage surpasses the confidence level. Prior to this, the coverage is below 95% and the scalar is greater than 1, resulting in a larger area for the `AdaPI` (Sky-Blue). Conversely, after this point, the `AdaPI` area diminishes, indicating a scalar smaller than one. Focusing on the latter half of the figure, almost all ground truths are covered by the algorithm, increasing the coverage. If the errors persist at a similar level for a period—meaning the predictions (black line) closely align with the ground truths (grey dots)—as demonstrated in Figure 5c, the `AdaPI` area should continue to shrink. This trend is evident in the figure, affirming the proper scalar-selection mechanism of `AdaPI`.

Finally, the efficiency of `AdaPI` needs to be assessed. `AdaPI` require minimal computational resources. Due to the page limit, complete results cannot be provided here. In summary, the total runtime only increase by around 1% on average, affirming `AdaPI`'s efficiency.

6 Conclusions

In this paper, we introduce the `AdaPI` algorithm for computing prediction intervals in streaming regression tasks, inheriting all the advantages of the MVE approach. Additionally, `AdaPI` dynamically adjusts the interval width, enabling the coverage to approach the desired confidence level. The results for coverage and interval width indicate that `AdaPI` achieves a closer match to the desired level compared to the static MVE approach. Furthermore, in many instances, `AdaPI` produces narrower prediction intervals. The analysis of scalar values underscores the robustness and stability of `AdaPI`. Moreover, our approach demonstrates the ability to adapt to concept drifts.

Looking ahead, there are several promising directions for future research. Firstly, the same adaptation principles could be applied to other static prediction interval algorithms, although the practical impact of these modifications would require thorough evaluation. Secondly, the scalar curve presented in this paper appears relatively moderate and conservative; a more aggressive and radical scaling strategy might yield even better results. Lastly, the evaluation of prediction intervals along both coverage and width dimensions introduces complexity to the analysis. A unified evaluation metric would significantly simplify the selection of a prediction interval method in practical applications.

References

1. R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani. Predictive inference with the jackknife+. 2021.
2. A. Bifet, G. Holmes, B. Pfahringer, P. Kranen, H. Kremer, T. Jansen, and T. Seidl. Moa: Massive online analysis, a framework for stream classification and clustering. PMLR, 2010.
3. L. Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.
4. H. Fanaee-T and J. Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, 2:113–127, 2014.
5. M. Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.
6. H. M. Gomes, J. P. Barddal, L. E. B. Ferreira, and A. Bifet. Adaptive random forests for data stream regression. In *ESANN*, 2018.
7. H. M. Gomes, J. Montiel, S. M. Mastelini, B. Pfahringer, and A. Bifet. On ensemble techniques for data stream regression. In *2020 IJCNN*. IEEE.
8. M. Hadjicharalambous, M. M. Polycarpou, and C. G. Panayiotou. Neural network-based construction of online prediction intervals. *Neural Computing and Applications*, 32(11):6715–6733, 2020.
9. G. J. Hahn. Factors for calculating two-sided prediction intervals for samples from a normal distribution. *Journal of the American Statistical Association*, 1969.
10. G. J. Hahn and W. Nelson. A survey of prediction intervals and their applications. *Journal of Quality Technology*, 5(4):178–188, 1973.
11. E. Ikonomovska, J. Gama, and S. Džeroski. Learning model trees from evolving data streams. *Data mining and knowledge discovery*, 23:128–168, 2011.
12. A. Khosravi, S. Nahavandi, D. Creighton, and A. F. Atiya. Lower upper bound estimation method for construction of neural network-based prediction intervals. *IEEE transactions on neural networks*, 22(3):337–346, 2010.
13. Y. Liu, J. Zhao, W. Wang, and W. Pedrycz. Prediction intervals for granular data streams based on evolving type-2 fuzzy granular neural network dynamic ensemble. *IEEE Transactions on Fuzzy Systems*, 29(4):874–888, 2020.
14. D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. In *ICNN'94*. IEEE, 1994.
15. G. Shafer and V. Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.
16. D. L. Shrestha and D. P. Solomatine. Machine learning approaches for estimation of prediction interval for the model output. *Neural networks*, 19(2):225–235, 2006.
17. Y. Sun, B. Pfahringer, H. M. Gomes, and A. Bifet. SOKNL: A novel way of integrating k-nearest neighbours with adaptive random forest regression for data streams. *Data Mining and Knowledge Discovery*, 2022.
18. S. G. Waugh. *Extending and benchmarking Cascade-Correlation: extensions to the Cascade-Correlation architecture and benchmarking of feed-forward supervised artificial neural networks*. PhD thesis, University of Tasmania, 1995.
19. C. Xu and Y. Xie. Conformal prediction interval for dynamic time-series. In *International Conference on Machine Learning*, pages 11559–11569. PMLR, 2021.
20. J. Zhao, W. Wang, C. Sheng, J. Zhao, W. Wang, and C. Sheng. Industrial prediction intervals with data uncertainty. *Data-Driven Prediction for Industrial Processes and Their Applications*, pages 159–222, 2018.
21. X. Zhao, S. Barber, C. C. Taylor, and Z. Milan. Interval forecasts based on regression trees for streaming data. *Advances in Data Analysis and Classification*, 15:5–36, 2021.