

Overlay_dx - Automating forecasting evaluation

Long H. Ngo¹, Mohammed Amine Chamli¹, Jonathan Rivalan¹, and Thomas Jaillon²

¹ Smile, Asnières, France

² Paris, France

Abstract. Traditional evaluation metrics provides numerical values but often lack comprehensibility, hindering effective differentiation of model performances. Our work addresses this challenge by introducing *overlay_dx*, a novel evaluation metric measuring the performance of time series prediction models. *Overlay_dx* is a visual metric that represents the percentage of predictions falling within a confidence interval around actual values. Additionally, once evaluation results are plotted, *overlay_dx* computes the area under the overlay curve, providing a quantitative measure of alignment between predicted and actual values across different thresholds and predictions. Through extensive experiments, we demonstrate that our approach offers a unified evaluation framework that combines both visual and numerical assessments, enabling improved model comparison and providing valuable insights for further research and optimization efforts in time series prediction.

Keywords: Machine learning · Evaluation metric · Time series data · Forecasting · Optimization.

1 Introduction

Time series prediction models have become increasingly central in various domains, from financial forecasting to industrial monitoring. However, evaluating these models presents a challenge. Traditional evaluation metrics provide numerical scores that, while mathematically sound, often fail to capture the nuanced performance characteristics that practitioners need to make informed decisions. This limitation becomes particularly apparent when comparing multiple models or when communicating model performance to stakeholders with varying levels of technical expertise.

For example, when comparing two forecasting models using RMSE, a difference between scores of 0.15 and 0.17 provides little intuitive understanding of real-world performance implications. Traditional metrics also fail to capture important aspects like timing of predictions and consistency across different scales, which are crucial for domain-oriented applications such as energy forecasting or financial markets analysis.

Current approaches to time series evaluation typically fall into three categories: point-wise metrics (e.g. Root Mean Squared Error (RMSE), Mean Absolute Error (MAE)), distribution-based metrics (e.g. Kullback–Leibler (KL)

divergence), and shape-based metrics (e.g. Dynamic Time Warping (DTW)). While each category offers specific insights, they often fail to provide a unified framework that combines statistical rigor with intuitive interpretability. This fragmentation in evaluation approaches makes it difficult for practitioners to make holistic assessments of model performance.

In this work, we address this fundamental challenge through the introduction of *overlay_dx*, a novel evaluation metric that combines visual interpretability with quantitative rigor. This new understandable evaluation metric helps to compare the performance of multiple models, not only through its score but also through visualization of prediction performance. The key contributions of our work include: 1) Development of a new visual evaluation metric (*overlay_dx*) that represents prediction accuracy through an intuitive confidence interval approach; 2) Introduction of a quantitative scoring mechanism based on the area under the overlay curve; 3) Demonstration of the metric’s effectiveness across various time series prediction scenarios; 4) Provision of a framework that bridges the gap between technical evaluation and interpretability.

The *overlay_dx* metric offers several advantages over traditional evaluation approaches, including enhanced visual interpretability for effective communication with non-technical stakeholders, robustness to outliers and extreme values, ability to capture performance characteristics across varying prediction thresholds, and seamless integration with existing evaluation workflows.

Overlay_dx is implemented as an open source project to ensure reproducibility and community adoption. The codebase, along with documentation and usage examples, is publicly available on GitHub³. This implementation supports seamless integration with popular machine learning frameworks and includes utilities for visualizing overlay curves and computing Area Under Curve (AUC) scores [1].

The rest of the paper is organized as follows: Section 2 introduces the related works; Section 3 presents the design principles; Section 4 shows the experiments; Section 5 concludes the paper while listing the contributions and future works.

2 Related works

The quality of time series predictions is assessed using evaluation metrics. Evaluation metrics measure the accuracy of predictions by comparing predicted values with actual values. The most commonly used evaluation metrics for assessing predictions include MAE [2, 3] and its family, RMSE [3, 4] and its family, and Akaike’s entropy-based Information Criterion (AIC) [5, 6]. Below, we review the most commonly used metrics for forecasting models evaluation.

Mean Absolute Error (MAE) represents the average of the absolute differences between predicted and actual values. This measure shows us what level of error to expect in average forecasts. As MAE is an average, it does not identify proportionally very high or low errors. With MAE, lower values indicate better predictions.

$$MAE = \frac{\sum_{i=1}^n |Y_i - \hat{Y}_i|}{n}, \quad (1)$$

³ https://github.com/Smile-SA/overlay_dx

where Y_i denotes actual values, $\hat{Y}_i$ predicted values, and n the number of predicted values.

Mean Absolute Percentage Error (MAPE) [7] represents the proportion of the mean difference between actual and predicted values divided by the actual value. This measure works best with data without zeros and extreme values due to the denominator. The smaller the MAPE, the better the model.

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right|. \quad (2)$$

Weighted Mean Absolute Percentage Error (WMAPE) [8] is similar to MAPE, but the errors are weighted according to the absolute value of the target value. This can be useful in preventing large errors in target values from overly influencing the evaluation measure.

$$WMAPE = \frac{\sum_{i=1}^n |Y_i - \hat{Y}_i|}{\sum_{i=1}^n |Y_i|}. \quad (3)$$

Mean Squared Error (MSE) [9] is defined as the average of the squared errors. This measure integrates variance and bias, and solves the extreme value and zero problems of MAE and MAPE. The smaller the score, the better the prediction.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2. \quad (4)$$

Root Mean Squared Error (RMSE) is defined as the square root of the mean square error (MSE). The RMSE value is in the same unit as the projected value, and we aim to minimize it.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2}. \quad (5)$$

Normalized Root Mean Squared Error (NRMSE) [10] is a version of RMSE normalized by the mean or difference of the extremums of the actual values. NRMSE is used to compare models on several data sets with different scales.

$$NRMSE = \frac{RMSE}{\text{mean}(y)} \text{ or } NRMSE = \frac{RMSE}{y_{\max} - y_{\min}}, \quad (6)$$

where $Y_{\max}/Y_{\min}$ denote maximum/minimum actual values and Y actual values.

Additionally, Dynamic Time Warping (DTW) [11] is a traditional metric that measures similarity between temporal sequences by allowing elastic transformation of time series. Unlike point-wise metrics, DTW can capture phase shifts and temporal distortions. However, its computational complexity and lack of intuitive interpretation limit its practical application.

These approaches, while valuable, often focus on specific aspects of model performance rather than providing a comprehensive evaluation framework. Recent advances in time series evaluation have explored multi-objective metrics that combine multiple aspects of prediction quality. For instance, TIGER [12] incorporates domain-specific constraints into the evaluation framework. However,

these approaches often increase complexity without proportionally improving interpretability, highlighting the need for our proposed *overlay_dx* method.

While existing metrics have served the field well, they share common limitations: (1) difficulty in interpreting scores in practical terms, (2) sensitivity to outliers and noise, and (3) lack of visual interpretability. These limitations particularly affect practitioners who need to make quick, informed decisions about model selection and optimization. Our proposed *overlay_dx* metric directly addresses these gaps while maintaining mathematical rigor.

3 Methodology

The methodology employed in this study involves the implementation of highly visual metrics and measures aimed at enhancing understanding through visualization while exploring new possibilities. Subsequently, a new visual metric is developed and applied to assess the performance of predictive models in time series analysis.

For instance, the peak overlap rate (local extrema) was implemented to assess whether predictions were capable of anticipating incidents or extreme values in a time series. Subsequently, the overlay (overlap rate) is defined, representing the percentage of predictions falling within a confidence interval drawn around the actual values of the series. As illustrated in Figure 1a, the overlay indicates the percentage of forecasted values (in red) falling within the intervals (gray, orange, and green) around the target values (in blue). The overlay value varies depending on the size of the interval drawn and can only be equal to or lower than the previous measure when reducing the threshold interval.

By varying the size of the confidence interval, multiple overlay measures were conducted on a single prediction. These interval-based measures can be visualized in a single graph (see Figure 1b, depicting a local CPU usage forecast). In this graph, the interval size (in decreasing order) is plotted on the x-axis, while the overlay measure for that interval is plotted on the y-axis, resulting in the curve profile as depicted in Figure 1b. This curve inevitably decreases as the size

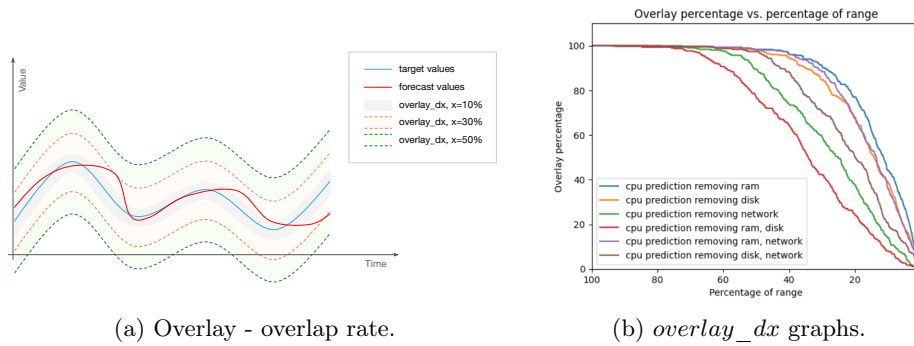


Fig. 1: Overview of *overlay_dx* visualisation.

of the confidence interval diminishes, allowing for the visual identification of the optimal curve profile, which remains the highest and longest. Thus, the overlay curve enables the visualization of a model’s performance through a curve profile. Consequently, within a single graph, it is possible to present multiple curve profiles and visually compare the performance of several models or configurations. As shown in Figure 1b, it is feasible to quantify and visually measure the performance of one approach compared to another.

Given that the visualization of the *overlay_dx* allows for the comparison of several curves, there must also be a numerical metric to differentiate between two very similar curve profiles and make the *overlay_dx* an evaluation metric. The *overlay_dx* score is obtained by evaluating the area under the overlay curve, which represents the cumulative *overlay_dx* measures for different thresholds. More specifically, *overlay_dx* consists of several measures of the overlay metric, which draws an interval around the target values and returns the percentage of forecasted values that fall within this interval. *Overlay_dx* calculates different measures of the overlay metric by reducing the size of its interval. Specifically, *overlay_dx* computes the percentage of values where the absolute difference between the forecast and actual values is less than or equal to a threshold x . A high score indicates better alignment between predicted and actual values, while a low score indicates a larger deviation from the ideal scenario where perfect alignment is achieved at threshold = 100. For instance, a score of 77% represents how well the forecasted values align with the actual values at different thresholds. It indicates that the achieved score is 77% of the maximum possible score, where perfect alignment would occur at 100% thresholds.

The score quantifies overall accuracy relative to the ideal scenario. The higher the score, the better the alignment between forecasted and actual values, while a lower score suggests larger deviations. The overlay curve intuitively evaluates model performance, providing insights into accuracy across different thresholds and highlighting areas for optimization.

The major advantage of this new metric lies in its visual representation through the overlay curve. Unlike traditional numerical measures, the overlay curve allows for a rapid understanding of the model’s performance without being significantly influenced by outliers. It provides an overview of the model’s accuracy across chosen thresholds. By examining the overlay curve, it is possible to assess the model’s performance for different threshold levels. The point where deviations become more significant can be identified, highlighting areas requiring potential improvement or specific attention. The overlay curve offers a more nuanced evaluation of model accuracy, considering performance at different thresholds and identifying specific thresholds requiring attention. It also provides a clear and understandable visual representation of model performance, facilitating communication and interpretation of results.

In summary, the use of the *overlay_dx* metric and overlay curve enables a more comprehensive evaluation of time series prediction accuracy. It provides a relative measure of alignment between predicted and actual values while offering a clear visualization of model performance. This approach offers more precise

Algorithm 1: Overlay()

Input : x , forecast, target

- 1 if $x == 0$: $x = 1$
- 2 Calculate the absolute difference between the forecast and actual values:
 $abs_diff = abs(target - forecast)$
- 3 Count the number of values where the absolute difference is less than or equal to x : $num_overlay = (abs_diff \leq x).sum()$

Output: Percentage of values that overlay:
 $pct_overlay = 100 * num_overlay / len(y)$

Algorithm 2: Overlay_dx Area Under Curve (AUC) Algorithm

Input : target, forecast, max_percentage, min_percentage, step

- 1 Compute the value range of the target:
 $value_range = \max(target) - \min(target)$
- 2 Generate a range of percentages:
 $percentages = np.arange(max_percentage, min_percentage, -step)$
- 3 Initialize an empty list: $overlay_percentages = []$
- 4 **for** $pct \in percentages$ **do**
- 5 Compute tolerance x : $x = \frac{pct}{100} \cdot \frac{value_range}{2}$
- 6 Compute the overlay percentage using Algorithm 1:
 $overlay_pct = Overlay(x, forecast, target)$
- 7 Append $overlay_pct$ to $overlay_percentages$
- 8 **end**

Output: $overlay_dx = AUC(overlay_percentages) / (max_percentage \cdot 100)$

perspectives for improvement and optimization by highlighting specific thresholds where deviations become more significant and facilitating the identification of areas requiring further research. Algorithm 2 summarizes the approach to calculate the *overlay_dx* metric with the help of Algorithm 1.

4 Experiments

To demonstrate the effectiveness and utility of the *overlay_dx* metric, we conducted extensive experiments using both synthetic and real-world time series data. These experiments aimed to compare *overlay_dx* with traditional metrics, assess its robustness to outliers, demonstrate its utility in model selection, and validate its applicability across different types of time series data.

4.1 Experimental Setup

To validate the performance of *overlay_dx*, we utilized three real-world datasets, namely Beijing Multi-Site Air-Quality (Pollution) [13], Electricity Transformer

Table 1: Generated groups of time series and their variations.

Group Name	Baseline Function	Added Variations
Constant	$f(x) = 100$	Noise, bias, delay, outliers
Linear Trend	$f(x) = x$	Misestimate, lag, bias, step changes
Seasonal	$50 + 30 \cdot \sin(x)$	Amplitude, frequency, asymmetry errors; missed peaks, phase shift
Random Walk	$\sum_{i=1}^n X_i$ - normal random vars	Smoothed, delay, noise, trend-bias, regime shifts
Multiple Seasonality	$50 + 20 \cdot \sin(x) + 10 \cdot \sin(7x)$	Missed short cycle, amplitude ratio error, noise
Trend Change	x if $x \in [0, 50]$ else $50 - x$	Missed reversal, late detection, overreaction
Cyclic	$x + \text{sawtooth}(x)$	Trend/cycle only, magnitude error
Exponential Growth	$\exp(x)$	Linear approximation, misestimate, delayed response, noise

Temperature (ETT) [14], and Electricity Load Diagrams 2011-2014 (ELD) [15] datasets and a simulated dataset.

The simulated dataset comprises diverse time series, each derived from a base curve modified by adding noise, introducing outliers, bias, delay, or other alterations (see Table 1). The base curve represents the target, while the modified versions simulate predictions made by a model. The objective is to showcase diverse scenarios and evaluate *overlay_dx*'s effectiveness across various situations.

The pollution dataset [13] includes hourly air pollutant measurements from 12 air quality monitoring stations in Beijing. It includes data on six major air pollutants along with six related meteorological variables. The data, covering 2013 to 2017, were sourced from the Beijing Municipal Environmental Monitoring Center and local weather stations operated by the China Meteorological Administration. Missing values are represented as NA.

The ETT dataset [14] contains two years of data from two counties in China, focusing on long-term electric power system deployment. It includes subsets at the 1-hour (ETTh1, ETTh2) and 15-minute (ETTm1) levels, with each data point featuring the target variable "oil temperature" and six power load features. It is split into training, validation, and test sets at a 12/4/4-month ratio.

The ELD dataset [15] contains electricity consumption of 370 points/ clients from 2011 to 2014.

We selected eight time series forecasting methods for comparison on the two real-world datasets, including Linear Regression [16], Random Forest Regressor [17, 18], XGBoost Regressor [19], LightGBM Regressor [20], K-nearest Neighbors Regressor [21, 22], Extra Trees Regressor [23], Bagging Regressor [24], and ARIMA [25].

In our experiments, we used TimeSeriesSplit⁴ as the cross-validator. Unlike traditional k-fold cross-validation, TimeSeriesSplit maintains the chronological

⁴ https://scikit-learn.org/1.6/modules/generated/sklearn.model_selection.TimeSeriesSplit.html

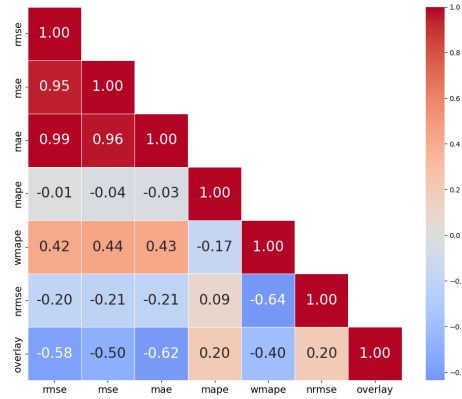


Fig. 2: Heat map of metrics correlation on the simulated dataset.

order of observations. In the k -th split, the first k folds serve as the training set, and the $(k+1)$ -th fold is used for testing, mimicking real-world scenarios where models are trained on historical data to predict future outcomes.

4.2 Experimental Results

To ensure robust evaluation, we conducted experiments across multiple dimensions. First, we calculated the correlation of *overlay_dx* and other metrics using the simulated dataset. The heat map in Figure 2 reveals a high negative correlation between the *overlay_dx* and absolute error metrics (MAE, RMSE, MSE), with values ranging from -0.62 to -0.5, implies that *overlay_dx* prioritizes absolute improvements in error magnitudes. It has weak positive correlations with percentage-based metrics (MAPE, NRMSE), suggesting that its interpretation may differ for datasets where relative errors are more critical. The weak correlation with WMAPE suggests *overlay_dx* is somewhat effective in capturing weighted errors but not as strongly as absolute error metrics. In conclusion, the varying correlations demonstrate that *overlay_dx* captures different performance aspects compared to traditional metrics, providing a complementary perspective on model accuracy.

We then compared *overlay_dx* with traditional metrics, including MAE, RMSE, MSE, MAE, MAPE, WMAPE, and NRSME, across all the models on the two real-world datasets.

Detailed error analysis revealed distinct performance patterns across different time scales. Figure 3 illustrates the visual prediction results of 6 forecasting methods on the ETT dataset. Visualization of predictions vs. actual values shows that all models sometimes struggle with sudden extreme events.

Table 2, which presents the *overlay_dx* scores and traditional metrics, and Figure 4, which shows the *overlay_dx* graphs for all models, demonstrate how *overlay_dx* metric offers a nuanced evaluation framework that complements traditional metrics by capturing performance across multiple confidence intervals. In

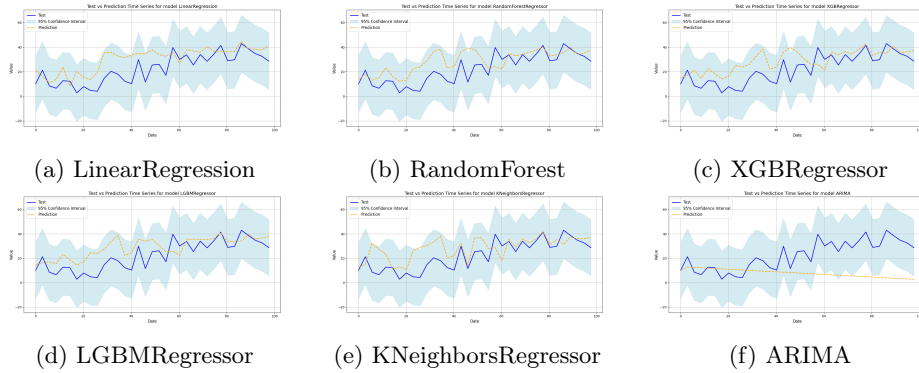


Fig. 3: The predictions of 6 forecasting methods on the ETT dataset. The orange/blue curves represent predictions/ ground truth values.

the pollution dataset, ExtraTreesRegressor demonstrates superior performance with the highest *overlay_dx* score of 0.952, accompanied by the lowest RMSE (36.018), MAE (25.9096), and MAPE (0.591). The WMAPE of 0.00 for ARIMA highlights a computational anomaly, underscoring the limitations of this metric. This suggests that the WMAPE calculation for ARIMA might have encountered a computational edge case problem. Similarly, in the ETT and ELD datasets, RandomForestRegressor and ARIMA lead with *overlay_dx* scores of 0.674 and 0.847, respectively, showcasing the metric's ability to provide comprehensive model assessment beyond single-point evaluations.

5 Conclusion and future works

Our introduction of *overlay_dx* represents a significant advancement in evaluating time series prediction models by combining visual interpretability and quantitative assessments into a comprehensive framework, addressing a critical gap in existing methods. The overlay curve provides an intuitive visual representation of model performance, allowing for rapid understanding and identification of areas for improvement, while *overlay_dx* offers a quantitative measure of alignment between predicted and actual values. This dual approach enhances the evaluation process, making it both interpretable and precise. Additionally, *overlay_dx* serves as a comprehensive evaluation framework, accommodating both technical and non-technical stakeholders to facilitate better communication and decision-making in model selection. Through extensive experimentation, we have demonstrated its reliability across various datasets, model types, and prediction scenarios.

While *overlay_dx* offers significant advantages, we acknowledge several limitations. While the computational complexity increases with the number of threshold levels evaluated, the choice of threshold ranges can influence the final

Table 2: Performance metrics of different forecasting models for three datasets.

Dataset	Model	rmse	mse	mae	mape	wmape	nrmse	overlay
Pollution	LinearRegression	47.410	2247.729	32.028	0.870	0.377	0.559	0.935
	RandomForestRegressor	40.325	1626.087	26.639	0.660	0.314	0.475	0.946
	XGBRegressor	39.195	1536.277	26.020	0.630	0.307	0.462	0.947
	LGBMRegressor	39.110	1529.555	25.910	0.620	0.305	0.461	0.947
	KNeighborsRegressor	43.781	1916.793	27.729	0.629	0.327	0.516	0.943
	ExtraTreesRegressor	36.018	1297.328	23.338	0.591	0.275	0.424	0.952
	BaggingRegressor	38.0439	1447.336	24.457	0.607	0.288	0.448	0.950
	ARIMA	84.4496	7131.738	62.472	2.814	0.00	0.995	0.874
ETT	LinearRegression	11.632	135.308	9.467	1.835	0.414	0.509	0.646
	RandomForestRegressor	11.219	125.862	8.723	2.347	0.381	0.490	0.674
	XGBRegressor	11.441	130.907	9.045	2.22	0.395	0.500	0.663
	LGBMRegressor	11.333	128.447	8.998	2.281	0.393	0.495	0.664
	KNeighborsRegressor	12.007	144.173	9.336	2.697	0.408	0.525	0.654
	ExtraTreesRegressor	11.523	132.769	9.205	2.649	0.402	0.504	0.656
	BaggingRegressor	11.508	132.424	9.113	2.568	0.398	0.503	0.660
	ARIMA	20.853	434.866	16.784	1.531	0.734	0.912	0.435
ELD	LinearRegression	8.875	78.766	6.423	2.2E15	1.583	2.187	0.683
	RandomForestRegressor	5.786	33.482	3.222	1.4E15	0.794	1.426	0.838
	XGBRegressor	5.747	33.024	3.514	1.5E15	0.866	1.416	0.823
	LGBMRegressor	5.813	33.791	3.256	1.3E15	0.803	1.433	0.837
	KNeighborsRegressor	9.101	82.820	7.688	5.1E15	1.895	2.243	0.610
	ExtraTreesRegressor	5.632	31.718	3.242	1.4E15	0.800	1.388	0.837
	BaggingRegressor	5.916	34.995	3.358	1.5E15	0.828	1.458	0.831
	ARIMA	6.079	36.957	3.059	0.8E15	0.754	1.498	0.847

score. Also, the visual interpretation may still require some training for optimal use.

This work opens several promising leads for future research. One direction involves the development of algorithms for automated threshold selection, enabling the determination of optimal threshold ranges for various types of time series data. Another is the extension of *overlay_dx* to accommodate multivariate time series prediction evaluation. Exploring real-time evaluation methods, such as streaming variants of *overlay_dx* for online assessment of prediction models, also presents valuable opportunities. Additionally, integrating *overlay_dx* into deep learning frameworks by developing *overlay_dx*-based loss functions for neural network training could enhance model performance. Finally, creating domain-specific adaptations of *overlay_dx* for specialized applications, such as financial forecasting or weather prediction, offers significant potential for targeted advancements.

We believe *overlay_dx* represents a significant step forward in time series model evaluation, providing a foundation for future research in the forecasting area. The metric’s ability to combine visual interpretability with quantitative rigor addresses a fundamental need in the field, and its extensibility provides numerous opportunities for future development and application.

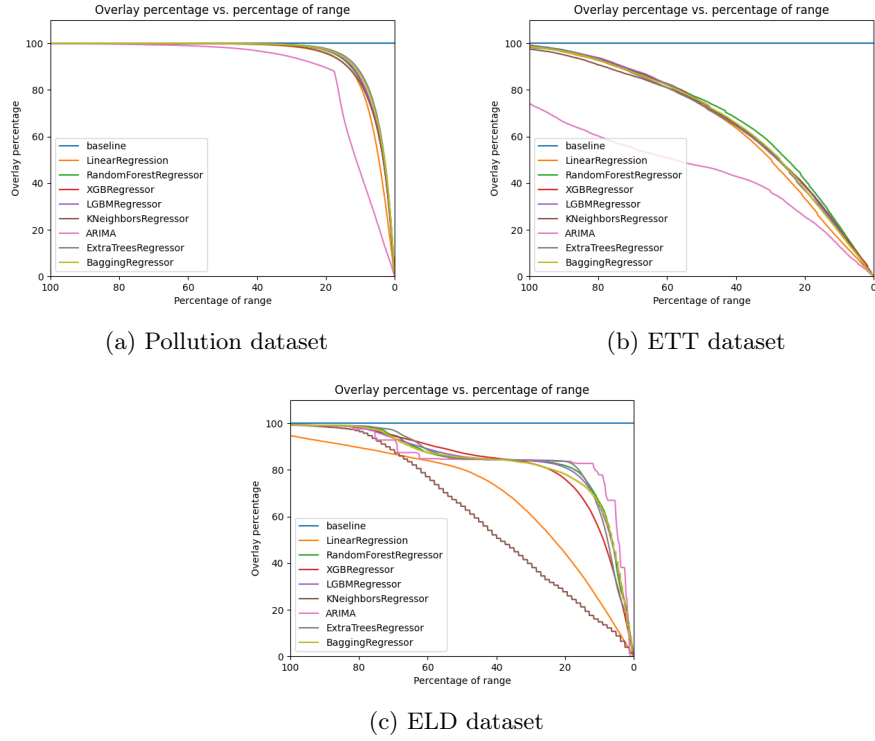


Fig. 4: *overlay_dx* graphs of various methods on 3 datasets.

Acknowledgments. This work has been funded by the European Union’s Horizon Europe research and innovation program under grant agreement No. 101070487 (NEPHELE).

References

1. Myerson, J., Green, L., Warusawitharana, M.: Area under the curve as a measure of discounting. *Journal of the experimental analysis of behavior* **76**, 235–243 (2001)
2. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research* **30**(1), 79–82 (2005)
3. Hodson, T.O.: Root mean square error (rmse) or mean absolute error (mae): When to use them or not. *Geoscientific Model Development Discussions* pp. 1–10 (2022)
4. Chai, T., Draxler, R.R., et al.: Root mean square error (rmse) or mean absolute error (mae). *Geoscientific model development discussions* **7**(1), 1525–1534 (2014)
5. Bozdogan, H.: Model selection and akaike’s information criterion (aic): The general theory and its analytical extensions. *Psychometrika* **52**(3), 345–370 (1987)
6. Cavanaugh, J.E., Neath, A.A.: The akaike information criterion: Background, derivation, properties, application, interpretation, and refinements. *Wiley Interdisciplinary Reviews: Computational Statistics* **11**(3), e1460 (2019)

7. De Myttenaere, A., Golden, B., Le Grand, B., Rossi, F.: Mean absolute percentage error for regression models. *Neurocomputing* **192**, 38–48 (2016)
8. Cleger-Tamayo, S., Fernández-Luna, J.M., Huete, J.F.: On the use of weighted mean absolute error in recommender systems. In: *RUE@ RecSys*. pp. 24–26 (2012)
9. Wang, Z., Bovik, A.C.: Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine* **26**(1), 98–117 (2009)
10. Shcherbakov, M.V., Brebels, A., Shcherbakova, N.L., Tyukov, A.P., Janovsky, T.A., Kamaev, V.A., et al.: A survey of forecast error measures. *World applied sciences journal* **24**(24), 171–176 (2013)
11. Müller, M.: Dynamic time warping. *Information retrieval for music and motion* pp. 69–84 (2007)
12. Cummins, C.A., McInerney, J.O.: A method for inferring the rate of evolution of homologous characters that can potentially improve phylogenetic inference, resolve deep divergence and correct systematic biases. *Systematic biology* **60**, 833–844 (2011)
13. Zhang, S., Guo, B., Dong, A., He, J., Xu, Z., Chen, S.X.: Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **473**(2205), 20170457 (2017)
14. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W.: Informer: Beyond efficient transformer for long sequence time-series forecasting. In: *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*. vol. 35, pp. 11106–11115. AAAI Press (2021)
15. Trindade, A.: *ElectricityLoadDiagrams20112014*. UCI Machine Learning Repository (2015), DOI: <https://doi.org/10.24432/C58C86>
16. Aalen, O.O.: A linear regression model for the analysis of life times. *Statistics in medicine* **8**(8), 907–925 (1989)
17. Segal, M.R.: *Machine learning benchmarks and random forest regression*. UCSF: Center for Bioinformatics and Molecular Biostatistics (2004)
18. Schonlau, M., Zou, R.Y.: The random forest algorithm for statistical learning. *The Stata Journal* **20**(1), 3–29 (2020)
19. Zhang, X., Yan, C., Gao, C., Malin, B.A., Chen, Y.: Predicting missing values in medical data via xgboost regression. *Journal of healthcare informatics research* **4**, 383–394 (2020)
20. Shehadeh, A., Alshboul, O., Al Mamlook, R.E., Hamedat, O.: Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, lightgbm, and xgboost regression. *Automation in Construction* **129**, 103827 (2021)
21. Peterson, L.E.: K-nearest neighbor. *Scholarpedia* **4**(2), 1883 (2009)
22. Song, Y., Liang, J., Lu, J., Zhao, X.: An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing* **251**, 26–34 (2017)
23. Ahmad, M.W., Reynolds, J., Rezgui, Y.: Predictive modelling for solar thermal energy systems: A comparison of support vector regression, random forest, extra trees and regression trees. *Journal of cleaner production* **203**, 810–821 (2018)
24. Aslam, F., Alyousef, R., Awan, H.H., Javed, M.F.: Forecasting the self-healing capacity of engineered cementitious composites using bagging regressor and stacking regressor. In: *Structures*. vol. 54, pp. 1717–1728. Elsevier (2023)
25. Ariyo, A.A., Adewumi, A.O., Ayo, C.K.: Stock price prediction using the arima model. In: *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*. pp. 106–112. IEEE (2014)