Mini-Project (ML for Time Series) - MVA 2024/2025

Abdessamad Badaoui abdessamad.badaoui@ens-paris-saclay.fr
Abdennacer Badaoui abdennacer.badaoui@student-cs.fr

January 7, 2025

1 Introduction and contributions

Anomaly detection in time series data has traditionally focused on identifying point-based anomalies—deviations occurring at a single moment in time. However, many real-world applications involve range-based anomalies, where anomalous behavior spans a continuous period. In this project, we studied a novel mathematical framework proposed by [2] for evaluating the performance of time series classification algorithms. This framework extends the classical Precision and Recall metrics to account for range-based anomalies, providing a more comprehensive evaluation methodology. Our objective was to implement this model from scratch, reproduce the results presented in the paper, and conduct novel experiments to evaluate the influence of different functions within this framework on the calculated metrics.

To conduct this project, each of us focused on implementing one specific metric (Precision or Recall), while regularly communicating with each other to ensure coherent work. We wrote the code ourselves, without using any external source code. Once the implementation was complete, we began by directly assessing the impact of different functions (overlap cardinality function, overlap size function, and positional bias function) on a synthetic dataset. Then, we reproduced nearly all the experiments presented in the paper. When it comes to the dataset used, since the goal of this project is to assess the new scoring model and not the anomaly detectors, we obtained the real and predicted anomaly ranges for each dataset from the official GitHub repository of the paper.

2 Method

Traditional scoring models (classical and Numenta[1]) lack support for partial detection and flexible time bias, making it challenging to accurately assess the performance of anomaly detection models and impossible to define domain-specific preferences. The new scoring model addresses these issues by considering the following aspects:

1. **Existence**: Detecting the presence of an anomaly, even with a single-point prediction, holds value for many applications.
2. **Size**: A larger correctly predicted portion of the anomaly increases the recall score.
3. **Position**: The relative position of the correctly predicted portion within the anomaly range may be important, depending on the application (e.g., front-end for cancer detection, back-end for robotic defense systems, etc.).

4. **Cardinality**: Identifying an anomaly with a single prediction is often more valuable than fragmented detections across multiple ranges.

To evaluate the performance of anomaly detection models, the scoring framework relies on a set of real anomaly ranges R and predicted anomaly ranges P. Each real anomaly range is denoted as R_i, while each predicted anomaly range is denoted as P_j. The number of real anomaly ranges and predicted anomaly ranges are represented by N_r and N_p, respectively. Additionally, the framework includes α, a parameter that adjusts the relative weight of the existence reward, and three core functions: $\gamma()$ (overlap cardinality function), $\omega()$ (overlap size function), and $\delta()$ (positional bias function).

The recall metric, $Recall_T(R, P)$, is calculated as the average recall across all real anomaly ranges:

$$Recall_T(R, P) = \frac{\sum_{i=1}^{N_r} Recall_T(R_i, P)}{N_r}.$$

For an individual real anomaly range R_i, the recall score is defined as:

$$Recall_T(R_i, P) = \alpha \times ExistenceReward(R_i, P) + (1 - \alpha) \times OverlapReward(R_i, P),$$

where the existence reward measures whether R_i overlaps with any predicted range:

$$ExistenceReward(R_i, P) = \begin{cases} 1, & \text{if } \sum_{j=1}^{N_p} |R_i \cap P_j| \geq 1, \\ 0, & \text{otherwise.} \end{cases}$$

The overlap reward quantifies the alignment between R_i and P using the $\omega()$ function, with more weight to a specific position of the prediction (front, back, middle, or flat) as determined by the selected $\delta()$ function (see Appendix A for the definition of these functions):

$$OverlapReward(R_i, P) = CardinalityFactor(R_i, P) \times \sum_{j=1}^{N_p} \omega(R_i, R_i \cap P_j, \delta).$$

Here, the cardinality factor determines whether R_i overlaps with a single predicted range or multiple ranges:

$$CardinalityFactor(R_i, P) = \begin{cases} 1, & \text{if } R_i \text{ overlaps with at most one } P_j \in P, \\ \gamma(R_i, P), & \text{otherwise.} \end{cases}$$

with $\gamma()$ is equal to 1 or an inversely proportional function to the number of distinct ranges that a given anomaly range overlaps.

Similarly, the precision metric, $Precision_T(R, P)$, is defined as the average precision across all predicted anomaly ranges:

$$Precision_T(R, P) = \frac{\sum_{i=1}^{N_p} Precision_T(R, P_i)}{N_p}.$$

For an individual predicted anomaly range P_i, the precision score is computed as:

$$Precision_T(R, P_i) = CardinalityFactor(P_i, R) \times \sum_{j=1}^{N_r} \omega(P_i, P_i \cap R_j, \delta).$$

There is no existence reward for $Precision_T$ because precision focuses on evaluating the accuracy of positive predictions, rather than measuring how many real anomalies are detected. When it comes to the positional bias $\delta()$ for $Precision_T$, we will focus only on the flat bias, as false alarms (FP) are generally considered uniformly undesirable regardless of when they occur.

3 Data

This study utilized a combination of real and synthetic datasets, all comprising time-ordered, univariate numeric time series with known anomalies (ground truth). The details of the datasets are summarized in the Appendix B.1 and B.2.

As we already have the real and predicted anomalies from three anomaly detectors—LSTM-AD, Greenhouse, and Luminol—for each of the aforementioned datasets, we will use these signals directly to evaluate our new scoring model.

To process the real and predicted anomaly signals, originally represented as binary sequences where 1 indicates an anomaly and 0 indicates no anomaly, two distinct formats were derived: **point-based** and **range-based**. In the point-based format, each 1 in the binary sequence is treated as an isolated anomaly, represented as a single point. For example, the binary sequence $[0, 1, 1, 1, 0]$ translates to the point-based format $[(1, 1), (2, 2), (3, 3)]$, where each tuple represents the position of an anomaly. Conversely, the range-based format captures contiguous runs of 1s as a single range, highlighting intervals of anomalies. For instance, the sequence $[0, 1, 1, 0, 1, 1, 1]$ translates to $[(1, 2), (4, 6)]$ in the range-based format, where each tuple denotes the start and end of an anomaly interval. The binary format will be used to calculate classical point-based precision and recall, illustrating the limitations of traditional metrics. The point-based format will be used to show that the new scoring model subsumes the classical methods effectively, while the range-based format will enable generalized precision and recall calculations, accounting for range-based anomalies in a more comprehensive manner.

The complete data analysis for each model on each dataset is presented in the Appendix B.3. These results present point-based precision, recall, and average cardinality across different datasets for three anomaly detection methods: LSTM-AD, Greenhouse, and Luminol. While these values provide an initial insight into the detection quality, they do not fully capture the nature of anomaly detection, where anomalies often occur as ranges rather than isolated points. The average cardinality, reflecting the number of predicted points associated with each true anomaly, highlights differences in the prediction behavior of models. For instance, high cardinality in datasets like NYC_Taxi suggests that the model tends to predict multiple points for the same anomaly, potentially leading to over-segmentation. This behavior, while influencing precision and recall, further underscores the challenge of interpreting point-based metrics in the context of range-based anomalies. These observations motivate a deeper consideration of evaluation methods that align better with the actual structure of anomalies.

4 Results

4.1 Analyzing Positional Bias and Cardinality Functions

To assess the impact of the positional bias and cardinality functions on the recall score (similar experiments can be conducted for precision), we created simple datasets with different characteristics, including real anomalies and various prediction types. These included predictions at the front, back, and center, as well as fragmented predictions to evaluate the effect of the overlap cardinality function. The results show that the positional bias function significantly influences recall, where aligning the bias to the prediction position (front, middle, or back) maximizes the recall, as shown in Figure 8. Additionally, using a reciprocal cardinality function, as seen in Figure 9, is more effective for penalizing fragmented predictions, especially when aiming for a single prediction per real anomaly. This highlights the importance of adapting both the bias and cardinality functions based on the nature of the predictions and domain-specific needs.

4.2 Comparison to the classical point-based model

The dataset used in all experiments is from LSTM-AD, unless specified otherwise.

In this experiment, we compared the new scoring model to the classical one. To do so, we computed recall, precision, and F1-score using both the classical model and different variants of the new model by changing the positional bias function, as presented in Appendix D. All figures show that the first two bars are equal across all datasets, indicating that the new model effectively subsumes the classical point-based model when all ranges are represented as unit-sized ranges, using the arguments $\alpha = 0$, $\gamma() = 1$, and $\delta() = \text{Flat}$. In Figure 10, except for the Time-Guided and Machine-Temp datasets, $Recall_T$ is smaller than $Recall$ in all other datasets. This is because the overlap reward is not fully earned, and anomalies are often captured by more than one predicted anomaly, resulting in a penalized score due to the cardinality factor. For the Machine-Temp dataset, the recall is perfect, as all anomaly ranges were entirely detected by a single range ($x = 1$ in the cardinality factor). Regarding the Time-Guided dataset, 12 ranges were detected—some fully and others partially—each by a single predicted range. Partially predicted ranges were located near the back-ends, which explains why $Recall_{T_{Back}}$ has a higher value than others. This behavior is significant in cases where detecting delayed anomalies, such as robotic defense systems with delayed responses, is more critical—a scenario not effectively captured by the classical model.

When it comes to precision, we focus only on the flat bias, as false alarms (FP) are generally considered uniformly bad regardless of when they occur. In Figure 11, except for the Time-Guided and NYC-Taxi datasets, $Precision_T$ is smaller than the point-based score. For the Time-Guided dataset, precision values are similar due to narrow real and predicted anomaly ranges, minimizing the distinction between point- and range-based metrics. In NYC-Taxi, narrow predictions against wide real anomalies result in many false positives. However, $Precision_T$ slightly exceeds traditional Precision because it accounts for overlap rewards more effectively (the predicted anomaly ranges N_p are fewer in number compared to the total points contained within those N_p ranges).

Figure 12 shows that $F1$-score exhibits a similar behavior regarding positional bias as Recall, meaning that this combined metric (the harmonic mean of recall and precision) is as expressive as the other metrics for range-based anomaly detection evaluation.

4.3 Comparison to the Numenta Anomaly Benchmark (NAB) scoring model

The three graphs in Figure 13 show similar behavior, with both models decreasing from left to right, except for the Space-Shuttle dataset. This indicates that our model can mimic the behavior of the NAB scoring system. Analyzing the difference between the two scores for the Space-Shuttle dataset, we see that this dataset has a few real anomaly ranges and a large number of predicted anomalies, all of medium size. This resulted in a large number of both false positives and true positives. NAB severely penalizes false positives, favoring precision over recall, which is why we observe a very small value. This suggests that NAB did not capture the relatively large recall (due to the low precision), whereas the new model captured it and produced a relatively larger value.

4.4 Evaluating multiple anomaly detectors

From Figure 15, we have:

- **Sine Dataset:** Luminol is the most accurate anomaly detector, showing the highest score, while Greenhouse is the least accurate, with the smallest score. All scoring models agree on this ranking.
- **ECG Dataset:** All scoring models agree that Greenhouse performs poorly, mainly due to a high number of false positives. LSTM-AD is favored by our model due to fewer, more accurate predictions, while Luminol's excessive predictions lead to score degradation because of the cardinality factor, and this was not detected by the Numenta score as it ranks Luminol first.
- **NYC-Taxi Dataset:** LSTM-AD and Greenhouse perform similarly, with our model favoring Greenhouse due to higher recall. Numenta strongly favors Luminol, despite it missing two anomaly ranges. Our model demonstrates effectiveness in evaluating anomaly detectors, capturing application requirements, and addressing data subtleties.

4.5 Cost analysis

The naive Approach in the $Recall_T$ and $Precision_T$ equations compares each $R_i \in R$ with all $P_j \in P$, leading to a computational complexity of $O(N_r \times N_p)$. Optimization 1 improves this by leveraging sequential relationships between ranges: by ordering ranges as timestamp pairs, we can iterate over R and P simultaneously, reducing the complexity to $O(\max N_r, N_p)$. Optimization 2 further enhances efficiency by applying positional bias functions (e.g., flat) in closed form, performing a single computation per range instead of for each point. Figure 16 shows that computing precision and recall in a naive manner requires significantly more time compared to the classical model, with the computation time growing quadratically with the number of real and predicted anomaly ranges (the classical model bars are not clearly visible as they are too small). However, after optimization, we observe a significant improvement in performance, with a linear trend relative to the number of ranges. This shows that the new range-based metrics are computationally efficient, with only a minimal overhead compared to the classical metrics.

References

[1] Alexander Lavin and Subutai Ahmad. “Evaluating Real-time Anomaly Detection Algorithms - the Numenta Anomaly Benchmark”. In: *CoRR* abs/1510.03336 (2015). arXiv: `1510.03336`. URL: `http://arxiv.org/abs/1510.03336`.

[2] N Tatbul. “Precision and Recall for Time Series”. In: *arXiv preprint arXiv:1803.03639* (2018).

Appendix

A Definitions of $\omega()$ and $\delta()$ functions

```
function $\omega$(AnomalyRange, OverlapSet, $\delta$)
MyValue $\leftarrow 0$
MaxValue $\leftarrow 0$
AnomalyLength $\leftarrow$ length(AnomalyRange)
for $i \leftarrow 1$, AnomalyLength do
    Bias $\leftarrow \delta(i, \text{AnomalyLength})$
    MaxValue $\leftarrow$ MaxValue + Bias
    if AnomalyRange[$i$] $\in$ OverlapSet then
        MyValue $\leftarrow$ MyValue + Bias
return MyValue / MaxValue
```

```
function $\delta(i, \text{AnomalyLength})$
return 1                                  ▷ Flat bias
function $\delta(i, \text{AnomalyLength})$
return AnomalyLength $- i + 1$            ▷ Front-end bias
function $\delta(i, \text{AnomalyLength})$
return $i$                                ▷ Back-end bias
function $\delta(i, \text{AnomalyLength})$
if $i \leq \text{AnomalyLength}/2$ then
    return $i$
else
    return AnomalyLength $- i + 1$
```

Figure 1: Definitions of $\omega()$ and $\delta()$ functions.

B Datasets Analysis

B.1 Datasets description

Dataset	Type	Description and Anomalies
NYC-Taxi	Real	Passenger counts recorded in 30-minute intervals. Anomalies include events like the NYC Marathon, Thanksgiving, Christmas, New Year's Day, and a snowstorm.
Twitter-AAPL	Real	Mentions of Apple's ticker symbol (AAPL) in tweets, aggregated every 5 minutes.
Machine-Temp	Real	Temperature sensor readings from an industrial machine. Anomalies include a planned shutdown, an unidentified error, and a catastrophic failure.
ECG	Synthetic	Based on real electrocardiogram data, augmented with additional synthetic anomalies.
Space-Shuttle	Synthetic	Sensor data from NASA valves, augmented with synthetic anomalies.
Sine	Synthetic	A sine wave oscillating between 0.2 and 0.5 over 360 timestamps. Stochastic anomalies span 50–100 time intervals.
Time-Guided	Synthetic	Monotonically increasing values with stochastic range-based anomalies exhibiting inverted negative values.

Table 1: A summary of used datasets

B.2 Datasets visualization

The different real and predicted anomalies identified by the anomaly prediction models for each of the datasets used.

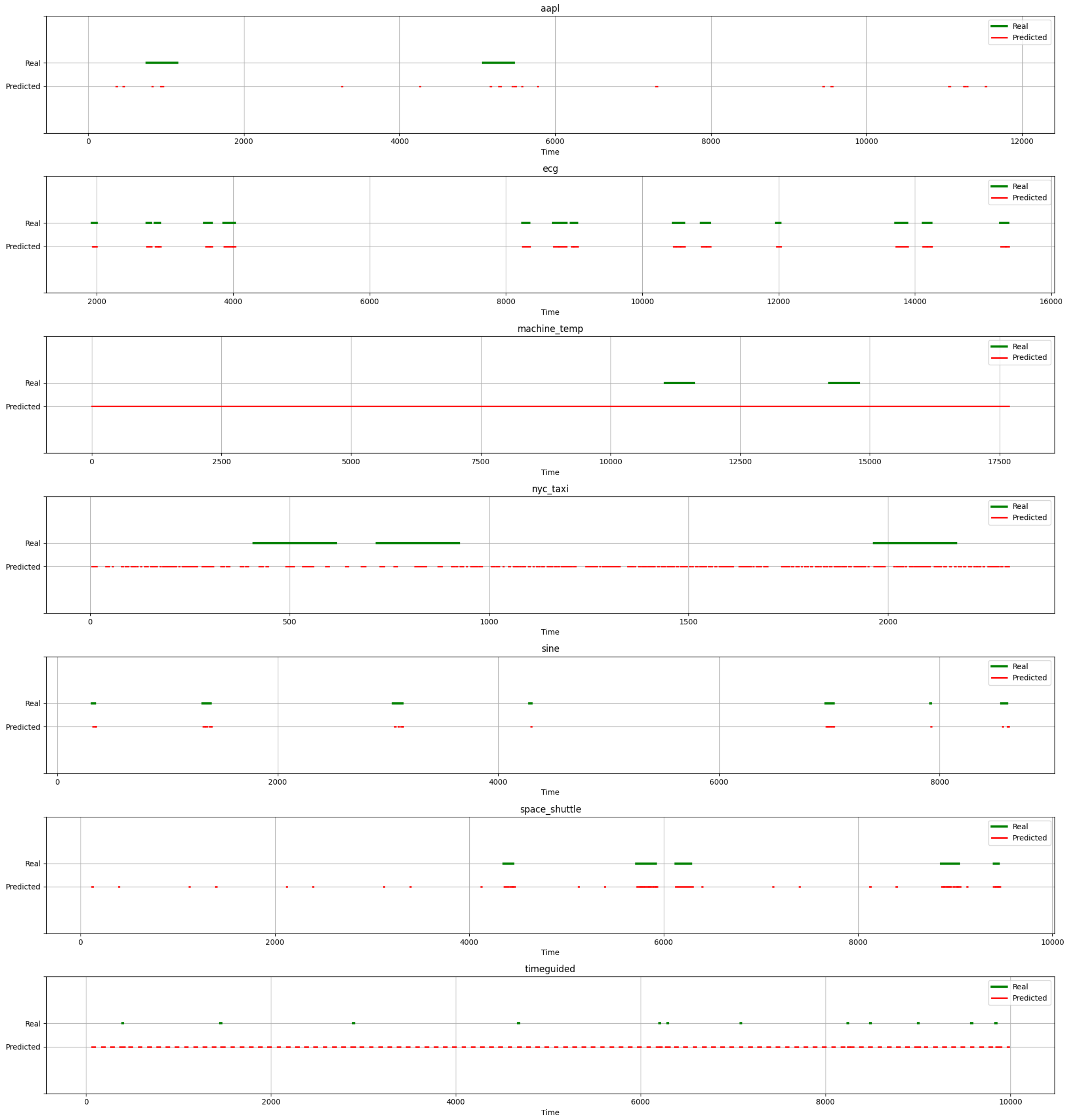

Figure 2: Real and Predicted anomalies by LSTM-AD for the different datasets

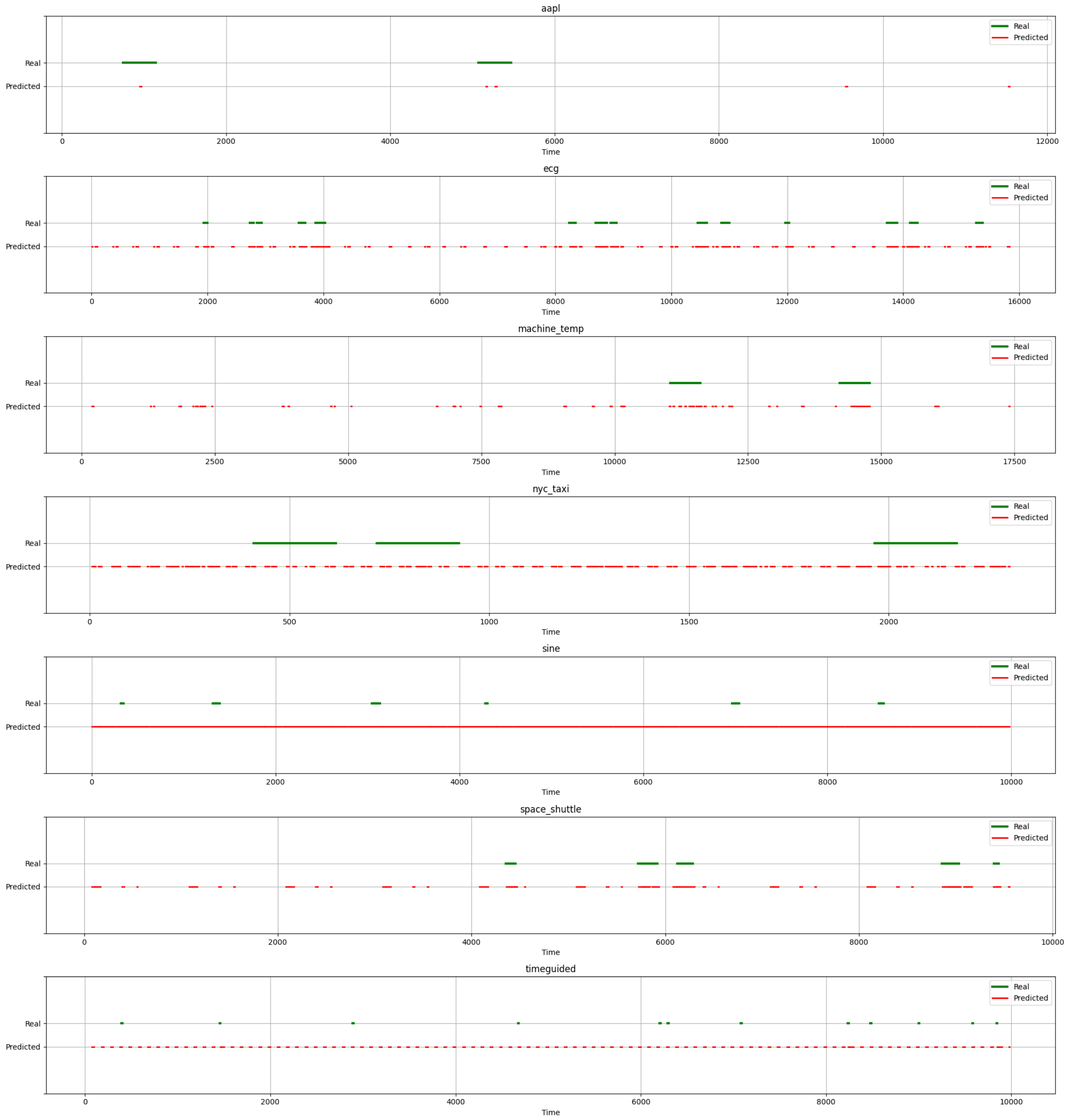

Figure 3: Real and Predicted anomalies by Greenhouse for the different datasets

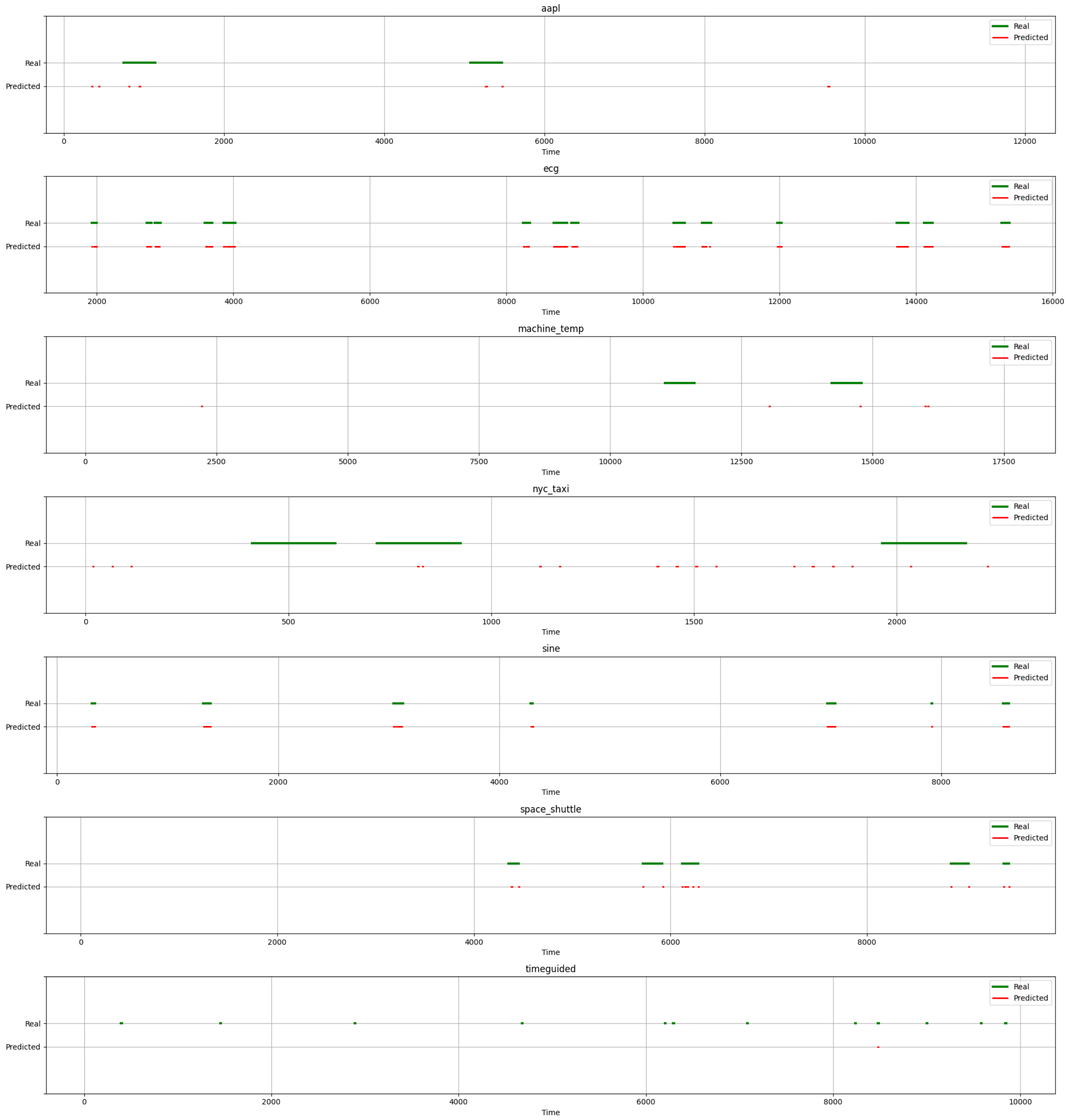

Figure 4: Real and Predicted anomalies by Luminol for the different datasets

B.3 Datasets Statistics

Here are the statistics of our three anomaly detectors on the different datasets. R stands for Real and P for Predicted. Prec and Rec are the classical Precision and Recall. The Avg Card stands for the average cardinality.

Table 2: Statistics of Various Datasets for LSTM-AD

Dataset	Num R	Num P	R Anomalies P	R Avg Len	P Avg Len	Prec	Rec	Avg Card
aapl	2	138	2	397	3.25	0.24	0.13	7
ecg	14	29	14	121.14	57.55	0.94	0.93	2
m_temp	2	1	2	567	17673	0.06	1	1
nyc_taxi	3	193	3	207	6.48	0.24	0.49	17.33
sine	8	45	6	43.38	5.18	0.85	0.57	4.62
s_shuttle	5	80	5	139.2	9.5	0.72	0.79	9.8
time-g	12	102	12	9.67	35.41	0.03	0.83	1

Table 3: Statistics of Various Datasets for Greenhouse

Dataset	Num R	Num P	R Anomalies P	R Avg Len	P Avg Len	Prec	Rec	Avg Card
aapl	2	19	2	397	5.32	0.5	0.06	2.5
ecg	14	193	14	121.14	19.89	0.42	0.96	1.14
m_temp	2	164	2	567	8.84	0.33	0.42	12.5
nyc_taxi	3	133	3	207	8.67	0.23	0.43	10.67
sine	8	55	8	43.38	170.38	0.04	1	1
s_shuttle	5	99	5	139.2	17.69	0.34	0.85	11.4
time-g	12	109	11	9.67	22.29	0.03	0.69	0.92

Table 4: Statistics of Various Datasets for Luminol

Dataset	Num R	Num P	R Anomalies P	R Avg Len	P Avg Len	Prec	Rec	Avg Card
aapl	2	55	2	397	2.56	0.37	0.07	6.5
ecg	14	269	14	121.14	4.96	0.99	0.78	18.86
ma_temp	2	371	2	567	1.17	0.1	0.04	15
nyc_taxi	3	63	2	207	1.59	0.15	0.02	3
sine	8	72	8	43.38	3.74	0.93	0.72	8.12
s_shuttle	5	68	5	139.2	2.38	0.67	0.16	8.8
time-g	12	111	7	9.67	1.17	0.19	0.22	0.83

C Analyzing Positional Bias and Cardinality Functions

C.1 Datasets

The datasets used for evaluating the impact of the positional bias function and the cardinality function are as follows:

$$\begin{aligned}
\text{Real Anomalies} &= [(4,15),(24,35),(43,56),(63,82),(91,105)]\\
\text{Front Predicted} &= [(4,7),(24,27),(43,47),(63,69),(91,95)]\\
\text{Back Predicted} &= [(12,15),(32,35),(52,56),(76,82),(101,105)]\\
\text{Centered Predicted} &= [(7,12),(27,32),(47,52),(69,76),(95,101)]\\
\text{Fragmented Predicted} &= [(4,7),(8,9),(24,27),(28,29),(30,31),(43,52),(63,82),(91,99),(101,105)]
\end{aligned}$$

C.2 Results

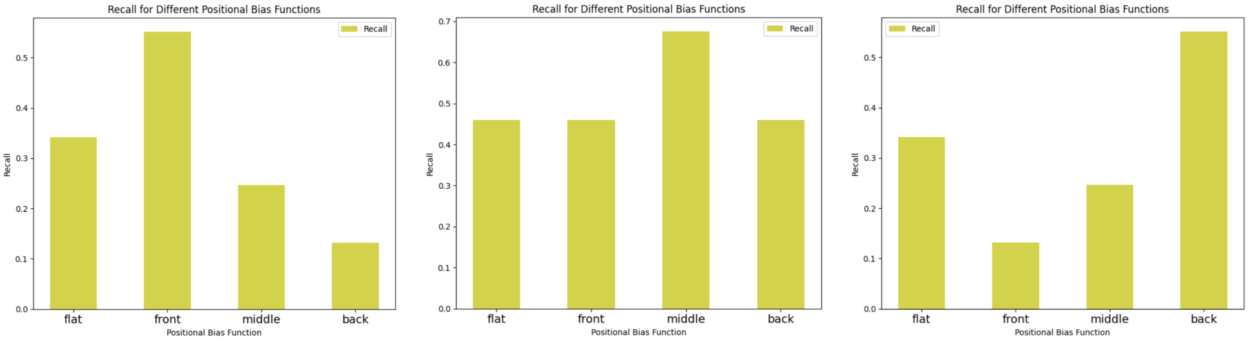

Figure 5: Front Predicted

Figure 6: Centered Predicted

Figure 7: Back Predicted

Figure 8: Recall score for different versions of predictions (front, center, back) with respect to the positional bias function

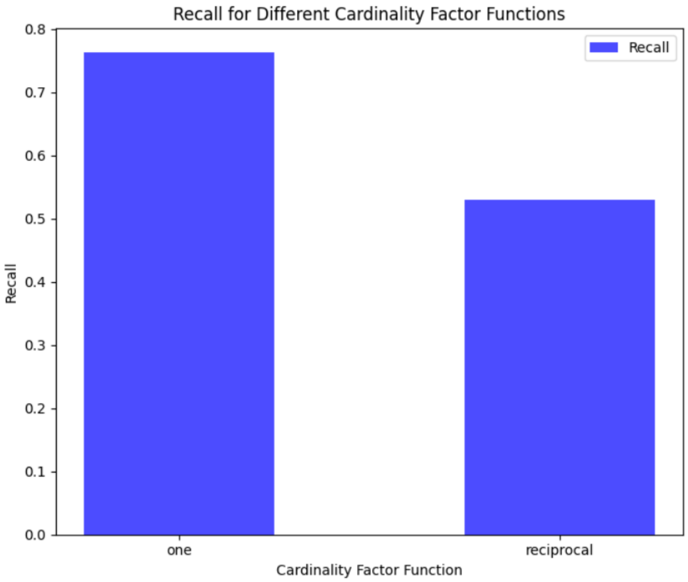

Figure 9: Recall for two cardinality functions: one and the reciprocal, for the fragmented predictions dataset

D Comparison to the classical point-based model

The goal of this first comparison is twofold: first, to show that the new scoring model subsumes the old one, and second, to highlight the additional variations in anomaly ranges that the new model can capture. `Recall_Classical`, `Precision_Classical`, and `F1_Classical` represent the classical scoring metrics (point-based model).

The parameters used for range-based metrics are:

- For the `[Recall/Precision/F1]T_Classical` metrics: $\alpha = 0$, $\gamma() = 1$, $\delta =$ Flat, with unit-sized ranges.
- For others: $\alpha = 0$, $\gamma() = \frac{1}{x}$, and $\delta =$ Flat.

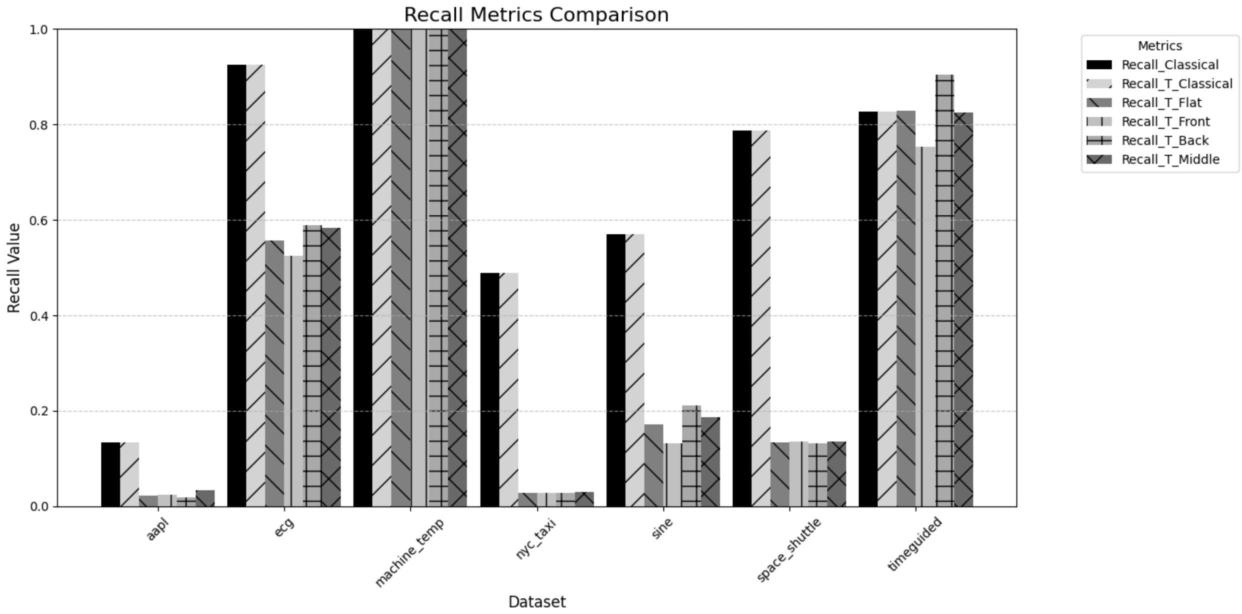

Figure 10: Recall Metrics Comparison

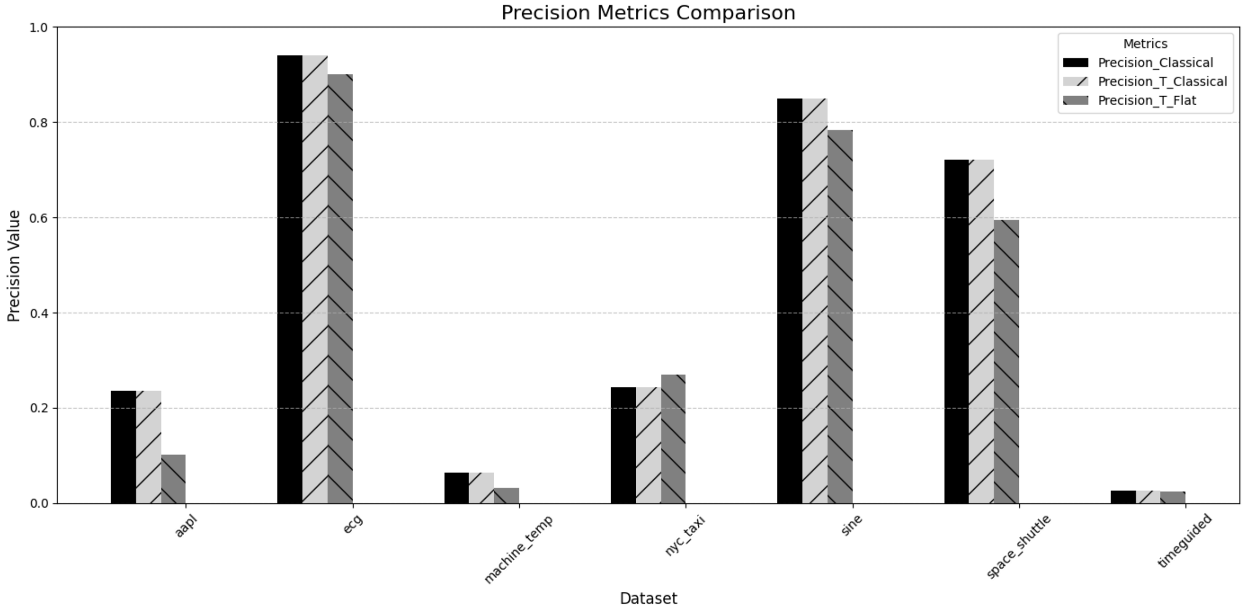

Figure 11: Precision Metrics Comparison

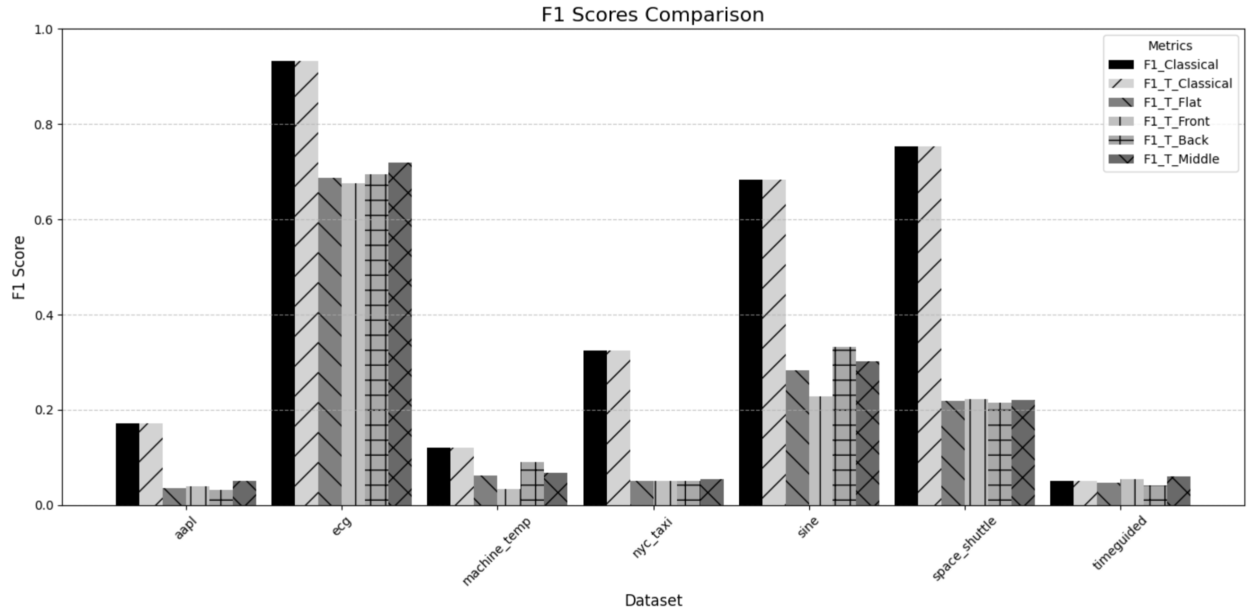

Figure 12: F1-Score Metrics Comparison

E Comparison to the Numenta Anomaly Benchmark (NAB) scoring model

In this section, we will compare the model to the Numenta Anomaly Benchmark (NAB) scoring model, with the goal of determining whether the new model can mimic the NAB model and extend it.

To obtain metrics that are similar to NAB, we will be using:

- $\alpha = 0$, $\gamma() = 1$ and $\delta() =$ Front for $Recall_T$ and Flat for $Precision_T$.
- P_i will be represented as points instead of ranges.
- `F1_T` to compare it with `Numenta_Standard`.
- `F0.5_T` to compare it with `Numenta_Reward_Low_FP`.
- `F2_T` to compare it with `Numenta_Reward_Low_FN`.

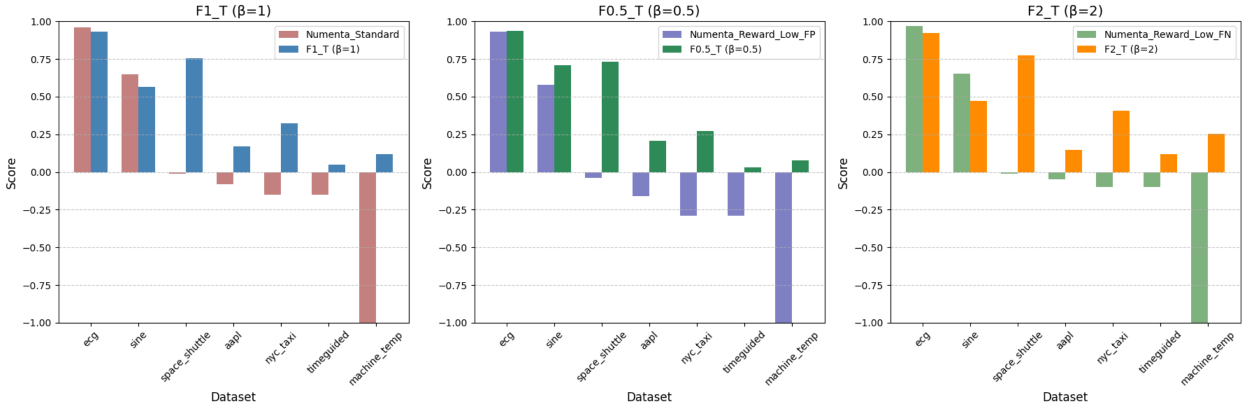

Figure 13: Comparison of the model to the Numenta model

F Evaluating multiple anomaly detectors

In this section, we will be evaluating and comparing the new model using different anomaly detectors: **LSTM-AD**, **Greenhouse**, and **Luminol**.

In this experiment, we consider an application that requires early detection of anomaly ranges in a non-fragmented manner, with equal importance placed on both precision and recall.

Model Settings:

- $\alpha = 0$
- $\gamma() = \frac{1}{x}$ for both $Precision_T$ and $Recall_T$
- $\delta()$ is front bias for $Recall_T$ and flat bias for $Precision_T$
- $\beta = 1$ for F score

In comparison, Numenta's closest application profile is the **Standard** model. In the classical point-based model, the only tunable parameter is $\beta = 1$ for the F_β score.

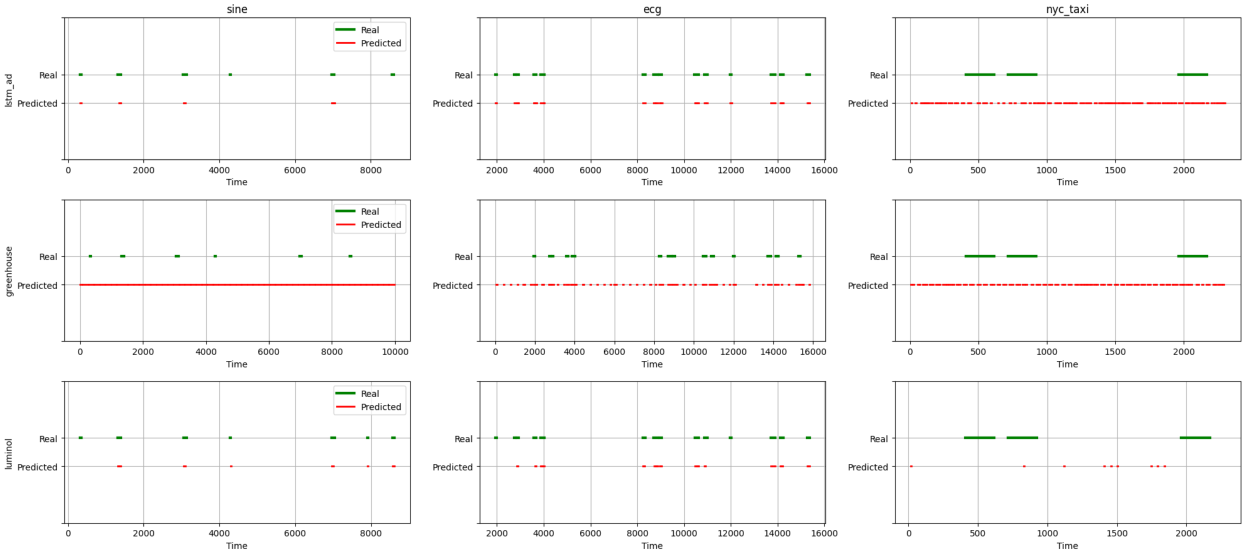

Figure 14: Real and predicted anomalies by the three anomaly detectors on the Sine, ECG, and NYC-Taxi datasets.

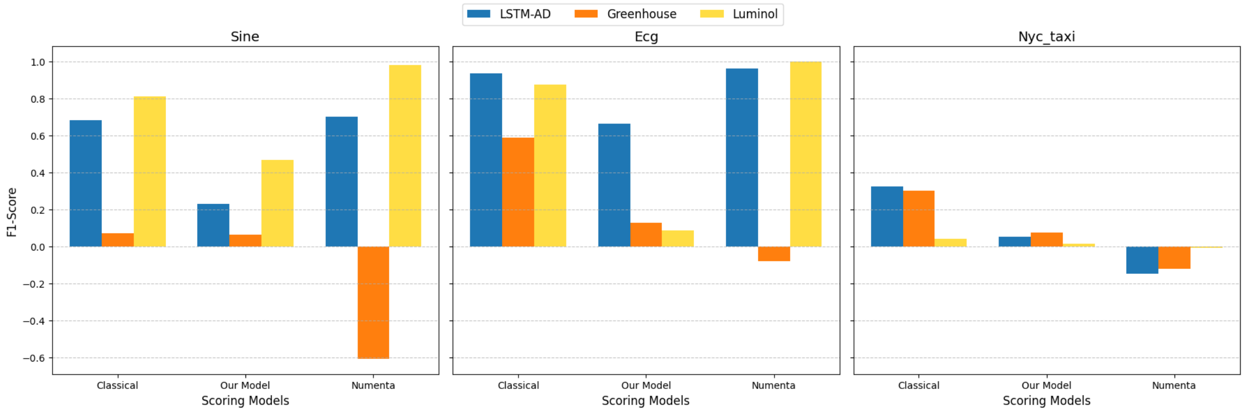

Figure 15: Evaluation of multiple anomaly detectors.

G Cost analysis

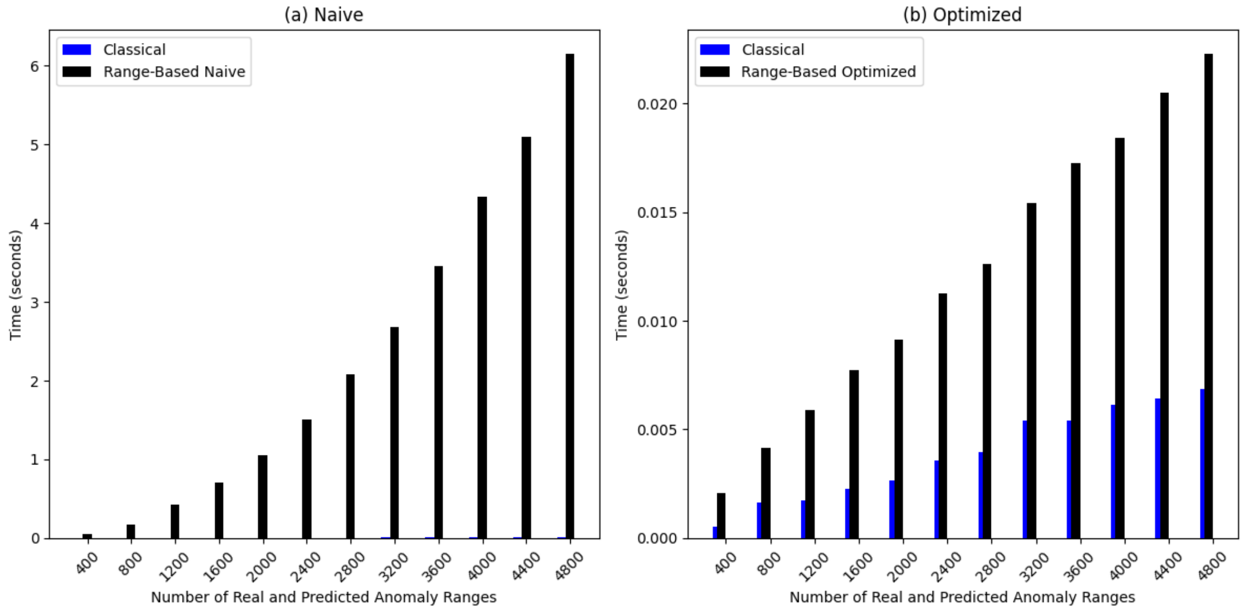

Figure 16: Naive vs. Optimized range-based metrics calculation.