UNSUPERVISED MODEL SELECTION FOR TIME-SERIES ANOMALY DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Anomaly detection in time-series has a wide range of practical applications. While numerous anomaly detection methods have been proposed in the literature, a recent survey concluded that no single method is the most accurate across various datasets. To make matters worse, anomaly labels are scarce and rarely available in practice. The practical problem of selecting the most accurate model for a given dataset without labels has received little attention in the literature. This paper answers this question *i.e.* Given an unlabeled dataset and a set of candidate anomaly detectors, how can we select the most accurate model? To this end, we identify three classes of surrogate (unsupervised) metrics, namely, *prediction error*, *model centrality*, and *performance on injected synthetic anomalies*, and show that some metrics are highly correlated with standard supervised anomaly detection performance metrics such as the F_1 score, but to varying degrees. We formulate metric combination with multiple imperfect surrogate metrics as a robust rank aggregation problem. We then provide theoretical justification behind the proposed approach. Large-scale experiments on multiple real-world datasets demonstrate that our proposed unsupervised approach is as effective as selecting the most accurate model based on partially labeled data.

1 INTRODUCTION

Anomaly detection in time-series data has gained considerable attention from the academic and industrial research communities due to the explosion in the amount of data produced and the number of automated system requiring some form of monitoring. A large number of anomaly detection methods have been developed to solve this task (Schmidl et al., 2022; Blázquez-García et al., 2021), ranging from simple algorithms (Keogh et al., 2005; Ramaswamy et al., 2000) to complex deep-learning models (Xu et al., 2018; Challu et al., 2022). These models have significant variance in performance across datasets (Schmidl et al., 2022), and evaluating their actual performance on real-world anomaly detection tasks is non-trivial, even when labeled datasets are available (Wu & Keogh, 2021).

Labels are seldom available for many, if not most, anomaly detection tasks. Labels are indications of which time points in a time-series are anomalous. The definition of an anomaly varies with the use case, but these definitions have in common that anomalies are rare events. Hence, accumulating a sizable number of labeled anomalies typically requires reviewing a large portion of a dataset by a domain expert. This is an expensive, time-consuming, subjective, and thereby error-prone task which is a considerable hurdle for labeling even a subset of data. Unsurprisingly, a large number of time-series anomaly detection methods are unsupervised or semi-supervised – *i.e.* they do not require any anomaly labels during training and inference.

There is no single universally best method (Schmidl et al., 2022). Therefore it is important to select the most accurate method for a given dataset without access to anomaly labels. The problem of unsupervised anomaly detection model selection has been overlooked in the literature, even though it is a key problem in practical applications. Thus, we offer an answer to the question: **Given an unlabeled dataset and a set of candidate models, how can we select the most accurate model?**

Our approach is based on computing "surrogate" metrics that correlate with model performance yet do not require anomaly labels, followed by aggregating model ranks induced by these metrics (Fig. 1). Empirical evaluation on 10 real-world datasets spanning diverse domains such as medicine,

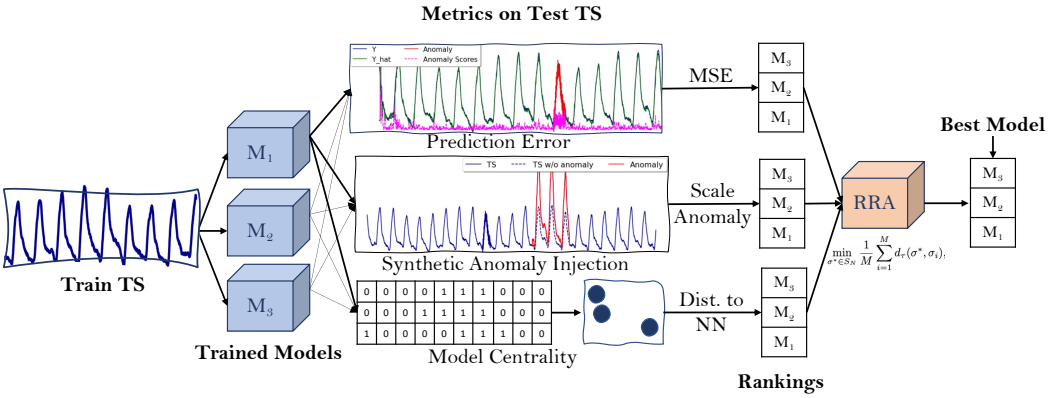

Figure 1: *The Model Selection Workflow*. We identify three classes of surrogate metrics of model quality (Sec. 3), and propose a novel robust rank aggregation framework to combine multiple rankings from metrics (Sec. 4).

sports, etc. shows that **our approach can perform unsupervised model selection as efficiently as selection based on labeling a subset of data**.

In summary, our contributions are as follows:

- To the best of our knowledge, we propose the first method for unsupervised selection of anomaly detection models on time-series data. To this end, we identify intuitive and effective unsupervised metrics for model performance.
- We propose a novel robust rank aggregation method for combining multiple surrogate metrics into a single model selection criterion. We show that our approach performs on par with selection based on labeling a subset of data.
- We conduct large-scale experiments on over 275 diverse time-series, spanning a gamut of domains such as medicine, entomology, etc. with 5 popular and widely-used anomaly detection models, each with 1 to 4 hyper-parameter combinations, resulting in over 5,000 trained models. Upon acceptance, we will make our code publicly available.

In the next section, we formalize the model selection problem. We then describe the surrogate metric classes in Sec. 3, and present our rank aggregation method in Sec. 4. Our empirical evaluation is described in Sec. 5. Finally, we summarize related work in Sec. 6 and conclude in Sec. 7.

2 Preliminaries & The Model Selection Problem

Let $\{\boldsymbol{x}_t, y_t\}_{t=1}^{T}$ denote a multivariate time-series (TS) with observations $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$, $\boldsymbol{x}_t \in \mathbb{R}^d$ and anomaly labels $(y_1, \ldots, y_T)$, $y_t \in \{0, 1\}$, where $y_t = 1$ indicates that the observation $\boldsymbol{x}_t$ is an anomaly. The labels are only used for evaluating our selection approaches, but not for the model selection procedure itself.

Next, let $\mathcal{M} = \{A_i\}_{i=1}^{N}$ denote a set of N candidate anomaly detection models. Each model A_i is a tuple (`detector`, `hyper-parameters`), i.e., a combination of an anomaly detection method (e.g. LSTM-VAE (Park et al., 2018)) and a fully specified hyper-parameter configuration (e.g. for LSTM-VAE, `hidden_layer_size=128, num_layers=2, ···`).

Here, we only consider models that do not require anomaly labels for training. Some models still require unlabeled time-series as training data. We therefore, consider a train/test split $\{\boldsymbol{x}_t\}_{t=1}^{t_{\text{test}}-1}$, $\{\boldsymbol{x}_t\}_{t=t_{\text{test}}}^{T}$, where the former is used if a model requires training (without labels). In our notation, A_i denotes a trained model.

We assume that a trained model A_i, when applied to observations $\{\boldsymbol{x}_t\}_{t=t_{\text{test}}}^{T}$, produces anomaly scores $\{s_t^i\}_{t=t_{\text{test}}}^{T}$, $s_t^i \in \mathbb{R}_{\geq 0}$. We assume that a higher anomaly score indicates that the observation is

more likely to be an anomaly. However, we do not assume that the scores correspond to likelihoods of any particular statistical model or that the scores are comparable across models.

Model performance or, equivalently, quality of the scores can be measured using a supervised metric $\mathcal{Q}(\{s_t\}_{t=1}^T, \{y_t\}_{t=1}^T)$, such as the Area under Precision Recall curve or best F_1 score, commonly used in literature. We discuss the choice of the quality metric in the next section.

In general, rather than considering a single time-series, we will be performing model selection for a set of L time-series with observations $\mathcal{X} = \{\{\boldsymbol{x}_t^j\}_{t=1}^T\}_{j=1}^L$ and labels $\mathcal{Y} = \{\{y_t\}_{t=1}^T\}_{j=1}^L$, where j indexes time-series. Let $\mathcal{X}_{\text{train}}$, $\mathcal{X}_{\text{test}}$, and $\mathcal{Y}_{\text{test}}$ denote the train and test portions of observations, and the test portion of labels, respectively. We are now ready to introduce the following problem.

Problem 1. ***Unsupervised Time-series Anomaly Detection Model Selection.*** *Given observations $\mathcal{X}_{test}$ and a set of models $\mathcal{M} = \{A_i\}_{i=1}^N$ trained using $\mathcal{X}_{train}$, select a model that maximizes the anomaly detection quality metric $\mathcal{Q}(A_i(\mathcal{X}_{test}), \mathcal{Y}_{test})$. The selection procedure cannot use labels.*

2.1 Measuring Anomaly Detection Model Performance

Anomaly Detection can be viewed as a binary classification problem where each time point is classified as an anomaly or a normal observation. Hence, the performance of a model A_i can be measured using standard precision and recall metrics. However, these metrics ignore the sequential nature of time-series; thus, time-series anomaly detection is usually evaluated using adjusted versions of precision and recall. We adopt previously adjusted versions of precision and recall (Xu et al., 2018; Challu et al., 2022). These metrics treat time points as independent samples, except when an anomaly lasts for several consecutive time points. In this case, detecting any of these points is treated as if all points inside the anomalous segment were detected.

Adjusted precision and recall can be summarized in a metric called adjusted F_1 score, which is the harmonic mean of the two. The adjusted F_1 score depends on the choice of a decision threshold on anomaly scores. In line with several recent studies, we consider threshold selection as a problem orthogonal to our problem (Schmidl et al., 2022; Laptev et al., 2015; Blázquez-García et al., 2021; Rebjock et al., 2021) and therefore, consider metrics that summarize model performance across all possible thresholds. Common metrics include area under the precision-recall curve and best F_1 (maximum F_1 over all possible thresholds). Like Xu et al. (2018), we found a strong correlation between the two. Thus, in the remainder of the paper, we restrict our analysis to identifying models with the highest *adjusted best* F_1 score (*i.e.* $\mathcal{Q}$ is *adjusted best* F_1).

3 Surrogate Metrics of Model Performance

The problem of unsupervised model selection is often viewed as finding an unsupervised metric which *correlates* well with supervised metrics of interest (Ma et al., 2021; Goix, 2016; Lin et al., 2020; Duan et al., 2019). Each unsupervised metric serves as a noisy measure of "model goodness" and reduces the problem of picking the best performing model according to the metric. We identified three classes of *imperfect* metrics that closely align with expert intuition in predicting the performance of time-series anomaly detection models. Our metrics are unsupervised because *they do not require anomaly labels*. However, some metrics such as F_1 score on synthetic anomalies are typically used for supervised evaluation. To avoid confusion we use term *surrogate* for our metrics. Below we elaborate on each class of surrogate metrics, starting with an intuition behind each class.

Prediction Error *If a model can forecast or reconstruct time-series well it must also be a good anomaly detector.* A large number of anomaly detection methods are based on forecasting or reconstruction (Schmidl et al., 2022), and for such models, we can compute forecasting or reconstruction error without anomaly labels. We collectively call these *prediction error* metrics and consider a common set of statistics, such as mean absolute error, mean squared error, mean absolute percentage error, symmetric mean absolute percentage error, and the likelihood of observations. For multivariate time-series, we average each metric across all the variables. For example, given a series of observations $\{x_t\}_{t=1}^T$ and their predictions $\{\hat{x}_t\}_{t=1}^T$, mean squared error is defined as $\text{MSE} = \frac{1}{T}\sum_{t=1}^T (x_t - \hat{x}_t)^2$. In the interest of space, we refer the reader to a textbook (e.g., Hyndman & Athanasopoulos (2018)) for definitions of other metrics. Prediction metrics are not perfect

for model selection because the time-series can contain anomalies. Low prediction errors are desirable for all points except anomalies, but we do not know where anomalies are without labels. While prediction metrics can only be computed for anomaly detection methods based on time-series forecasting or reconstruction, the next two classes of metrics apply to any anomaly detection method.

Synthetic Anomaly Injection *A good anomaly detection model will perform well on data with synthetically injected anomalies.* Some studies have previously explored using synthetic anomaly injection to train anomaly detection models (Carmona et al., 2021). Here, we extend this line of research by systematically evaluating model selection capability after injecting different kinds of anomalies. Given an input time-series without anomaly labels, we randomly inject an anomaly of a particular type. The location of the injected anomaly is then treated as a (pseudo-)positive label, while the rest of the time points are treated as a (pseudo-)negative label. We then select a model that achieves the *best adjusted* F_1 score with respect to pseudo-labels. Instead of relying on complex deep generative models (Wen et al., 2020) we devise simple and efficient yet effective procedures to inject anomalies of previously described types (Schmidl et al., 2022; Wu & Keogh, 2021) as shown in Fig. 2. We defer the details of anomaly injection approaches to Appendix A.4. Model selection based on anomaly injection might not be perfect because (i) real anomalies might be of a different type compared to injected ones, and (ii) there may already be anomalies for which our pseudo-labels will be negative.

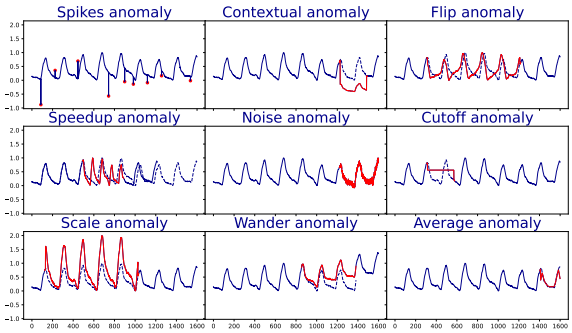

Figure 2: *Different kinds of synthetic anomalies*. We inject 9 different types of synthetic anomalies randomly, one at a time, across multiple copies of the original time-series. The - - is original time-series before anomalies are injected, – is the injected anomaly and the solid – is the time-series after anomaly injection.

Model Centrality *There is only one ground truth, thus models close to ground truth are close to each other. Hence, the most "central" model is the best one.* Centrality-based metrics have seen recent success in unsupervised model selection for disentangled representation learning (Lin et al., 2020; Duan et al., 2019) and anomaly detection (Ma et al., 2021). To adapt this approach to time-series, we leverage the scores from the anomaly detection models to define a notion of model proximity. Anomaly scores of different models are not directly comparable. However, since anomaly detection is performed by thresholding the scores, we can consider ranking of time points by their anomaly scores. Let $\sigma_k(i)$ denote the rank of time point i according to the scores from model k. We define the distance between models A_k and A_l as the Kendall's τ distance, which is the number of pairwise disagreements:

$$d_\tau(\sigma_k, \sigma_l) = \sum_{i<j} \mathbb{I}\{(\sigma_k(i) - \sigma_k(j))(\sigma_l(i) - \sigma_l(j)) < 0\} \tag{1}$$

Next, we measure model centrality as the average distance to its K nearest neighbors, where K is a parameter. This metric favors models which are close to their nearest neighbors. The centrality-based approach is imperfect, because "bad" models might produce similar results and form a cluster.

Each metric, while imperfect, is predictive of anomaly detection performance but to a varying extent as we show in Sec. 5. Access to multiple noisy metrics raises a natural question: *How can we combine multiple imperfect metrics to improve model selection performance?*

4 Robust Rank Aggregation

Many studies have explored the benefits of combining multiple sources of noisy information to reduce error in different areas of machine learning such as crowd-sourcing (Dawid & Skene, 1979) and programmatic weak supervision (Ratner et al., 2016; Goswami et al., 2021; Dey et al., 2022;

Gao et al., 2022; Zhang et al., 2022). Each of our surrogate metrics induces a (noisy) model ranking. Thus two natural questions arise in the context of our work: (1) *How do we reliably combine multiple model rankings?* (2) *Does aggregating multiple rankings help in model selection?*

4.1 APPROACHING MODEL SELECTION AS A RANK AGGREGATION PROBLEM

Let $[N] = \{1, 2, \cdots, N\}$ denote the universe of elements and S_N be the set of permutations on $[N]$. For $\sigma \in S_N$, let $\sigma(i)$ denote the rank of element i and $\sigma^{-1}(j)$ denote the element at rank j. Then the rank aggregation problem is as follows.

Problem 2. ***Rank Aggregation.*** *Given a collection of M permutations $\sigma_1, \cdots, \sigma_M \in S_N$ of N items, find $\sigma^* \in S_N$ that best summarizes these permutations according to some objective $C(\sigma^*, \sigma_1, \cdots, \sigma_M)$.*

In our context, the N items correspond to the N candidate anomaly detection models, M corresponds to the number of surrogate metrics, σ^* is the best summary ranking of models, and $\sigma^{*-1}(1)$ is the predicted best model.[1]

The problem of rank aggregation has been extensively studied in social choice, bio-informatics and machine learning literature. Here, we consider *Kemeny rank aggregation* which involves choosing a distance on the set of rankings and finding a barycentric or median ranking (Korba et al., 2017). Specifically, the rank aggregation problem is known as the Kemeny-Young problem if the aggregation objective is defined as $C = \frac{1}{M}\sum_{i=1}^{M} d_\tau(\sigma^*, \sigma_i)$, where d_τ is the Kendall's τ distance (Eq. 1).

Kemeny aggregation has been shown to satisfy many desirable properties, but it is NP-hard even with as few as 4 rankings (Dwork et al., 2001). To this end, we consider an efficient approximate solution using the *Borda method* (Borda, 1784) in which each item (model) receives points from each ranking (surrogate metric). For example, if a model is ranked r-th by a surrogate metric, it receives $(N - r)$ points from that metric. The models are then ranked in descending order of total points received from all metrics.

Suppose that we have several surrogate metrics for ranking anomaly detection models. Why would it be beneficial to aggregate the rankings induced by these metrics? The following theorem provides an insight into the benefit of Borda rank aggregation.

Consider two arbitrary models i and j with (unknown) ground truth rankings $\sigma^*(i)$ and $\sigma^*(j)$. Without the loss of generality, assume $\sigma^*(i) < \sigma^*(j)$, i.e., model i is better. Next, let's view each surrogate metric $k = 1, \ldots, M$ as a random variable Ω_k taking values in permutations S_N, such that $\Omega_k(i) \in [N]$ denotes the rank of item i. We apply Borda rank aggregation on realizations of these random variables. Theorem 1 states that, under certain assumptions, the probability of Borda ranking making a mistake (i.e., placing model i lower than j) is upper bounded in a way such that increasing M decreases this bound.

Theorem 1. *Assume that Ω_k's are pairwise independent, and that $\mathbb{P}(\Omega_k(i) < \Omega_k(j)) > 0.5$ for any arbitrary models i and j, i.e., each surrogate metric is better than random. Then the probability that Borda aggregation makes a mistake in ranking items i and j is at most $2\exp\left(-\frac{M\epsilon^2}{2}\right)$, for some fixed ϵ.*

The proof of the theorem is given in Appendix A.7. We do not assume that Ω_k are identically distributed, but only that Ω_k's are pairwise independent. We emphasize that the notion of independence of Ω_k as random variables is different from rank correlation between realizations of these variables (i.e., permutations). For example, two permutations drawn independently from a Mallow's distribution can have a high Kendall's τ coefficient (rank correlation). Therefore, our assumption of independence, commonly adopted in theoretical analysis (Ratner et al., 2016), does not contradict our observation of high rank correlation between surrogate metrics.

The Borda method weighs all surrogate metrics equally. However, our experience and theoretical analysis suggest that focusing only on "good" metrics can improve performance after aggregation (see Sec. 5 and Corollary 1). But how do we identify "good" ranks without access to anomaly labels?

[1] We use terms "ranking" and "permutation" as synonyms. The subtle difference is that in a ranking multiple models can receive the same rank. Where necessary, we break ties uniformly at random.

4.2 EMPIRICAL INFLUENCE AND ROBUST RANK AGGREGATION

Since we do not have access to anomaly labels, we introduce an intuitive unsupervised way to identify "good" or "reliable" rankings based on the notion of empirical influence. Empirical influence functions have been previously used to detect *influential* cases in regression (Cook & Weisberg, 1980). Given a collection of M rankings $\mathcal{S} = \{\sigma_1, \cdots, \sigma_M\}$, let Borda($\mathcal{S}$) denote the aggregate ranking from the Borda method. Then for some $\sigma_i \in \mathcal{S}$, empirical influence EI$(\sigma_i, \mathcal{S})$ measures the impact of σ_i on the Borda solution.

$$\text{EI}(\sigma_i, \mathcal{S}) = f(\mathcal{S}) - f(\mathcal{S} \setminus \sigma_i), \quad \text{where } f(\mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{i=1}^{|\mathcal{A}|} d_\tau(\text{Borda}(\mathcal{A}), \sigma_i) \tag{2}$$

Under the assumption that a majority of rankings are good, EI is likely to identify bad rankings as ones with high positive EI. This is intuitive since removing a bad ranking results in larger decrease in the objective $f(\mathcal{S} \setminus \{\sigma_i\})$.

We cluster all rankings based on their EI using a two-component single-linkage agglomerative clustering (Murtagh & Contreras, 2012), discard all rankings from the "bad" cluster, and apply Borda aggregation to the remaining ones. To further improve ranking performance of aggregated rank, especially at the top, we only consider models at the top-k ranks, setting the positions of the rest of the models to N (Fagin et al., 2003). See Appendix A.6.1 for details. We collectively refer to these variations as Robust Borda rank aggregation.

5 EVALUATION RESULTS

5.1 DATASETS

We carry out experiments on two popular and widely used real-world collections with diverse time-series and anomalies:

UCR Anomaly Archive (`UCR`) (Wu & Keogh, 2021) Wu and Keogh recently found flaws in many commonly used benchmarks for anomaly detection tasks. They introduced the UCR time-series Anomaly Archive, a collection of 250 diverse univariate time-series of varying lengths spanning domains such as medicine, sports, entomology, and space science, with natural as well as realistic synthetic anomalies overcoming the pitfalls of previous benchmarks. Given the heterogeneity of included time-series, we split UCR into 9 subsets by domain, (1) Acceleration, (2) Air Temperature, (3) Arterial Blood Pressure, ABP, (4) Electrical Penetration Graph, EPG, (5) Electrocardiogram, ECG, (6) Gait, (7) NASA, (8) Power Demand, and (9) Respiration, RESP. We will refer to each of these subsets as a separate *dataset*.

Server Machine Dataset (`SMD`) (Su et al., 2019) SMD comprises of multivariate time-series with 38 features from 28 server machines collected from large internet company over a period of 5 weeks. Each entity includes common metrics such as CPU load, memory and network usage etc. for a server machine. Both train and test sets contain around 50,000 timestamps with 5% anomalous cases. While there has been some criticism of this dataset (Wu & Keogh, 2021), we use SMD due to its multivariate nature and the variety it brings to our evaluation along with UCR. We emphasize that on visually inspecting the SMD, we found most of the anomaly labels to be accurate, much like (Challu et al., 2022).

We refer to SMD as a single dataset, and thus in our evaluation consider 10 datasets (9 from UCR + SMD).

5.2 MODEL SET

There are over a hundred unique time-series anomaly detection algorithms developed using a wide variety of approaches. For our experiments, we implemented a representative subset of 5 popular methods (Table 1), across two different learning types (unsupervised and semi-supervised) and three different method families (forecasting, reconstruction and distance-based) (Schmidl et al., 2022). We defer brief descriptions of models to Appendix A.5.

We created a pool of **3** k-NN (Ramaswamy et al., 2000), **4** moving average, **4** DGHL (Challu et al., 2022), **4** LSTM-VAE (Park et al., 2018) and **4** RNN (Chang et al., 2017) models by varying hyper-parameters for each model, for a total **19** combinations. Finally, to efficiently train our models, we sub-sampled all time-series with $T > 2560$ by a factor of 10.

Model	Area	Learning Type	Method Family
`Moving Average`	Statistics	Unsupervised	Forecasting
k-`NN` (Ramaswamy et al., 2000)	Classical ML	Semi-supervised	Distance
`Dilated RNN` (Chang et al., 2017)	Deep Learning	Semi-supervised	Forecasting
`DGHL` (Challu et al., 2022)	Deep Learning	Semi-supervised	Reconstruction
`LSTM-VAE` (Park et al., 2018)	Deep Learning	Semi-supervised	Reconstruction

Table 1: *Models set and their properties*. Detailed descriptions of model architectures and hyper-parameters can be found in Appendix A.5.

5.3 Experimental Setup

Evaluating model selection performance. We evaluate all model selection strategies based on the adjusted best F_1 (Section 2) of the top-1 selected model. While there exist other popular measures to evaluate ranking algorithms such as Normalized Discounted Cumulative Gain (Wang et al., 2013), Mean Reciprocal Rank, etc., our evaluation metric is grounded in practice, since in our setting only one model will be selected and used.

Our approach can be used to select the best model for an individual time-series. Since anomalies are rare, a single time-series may contain only a few anomalies (e.g., one point anomaly in the case of some UCR time-series). This makes the *evaluation* of a selection strategy very noisy. To combat this issue, we perform model selection at the dataset level, i.e., we select a single method for a given dataset.

Baselines. As a baseline, we consider an approach in which we label a part of a dataset and select the best method based on these labels. We call this baseline "*supervised selection*" or `SS` in short. This represents current practice, where a subset of data is labeled, and the best performing model is used in the rest of the dataset. Specifically, we randomly partition each dataset into *selection* (20%) and *evaluation* (80%) sets. Each time-series in the dataset is assigned to one of these sets. The baseline method selects a model based on anomaly labels in the selection set. Our surrogate metrics do not use the selection set. Instead, the surrogate metrics are computed over the evaluation set without using the labels. For reference, we also report the expected performance when selecting a model from the model set at random ("`random`") and the performance of the actual best model ("`oracle`").

Adjusted best F_1 scores of the selected model (either by the baseline, surrogate metric or rank aggregation) are then reported. The entire experiment is repeated 5 times with random selection/evaluation splits.

Model selection strategies. We compare **17** individual surrogate metrics, including **5** prediction error metrics, **9** synthetic anomaly injection metrics, **3** metrics of centrality (average distance to 1, 3 and 5 nearest neighbors) (Section3); and **4** proposed robust variations of Borda rank aggregation (see Appendix A.6.2).

In the preliminary round of evaluations we found 5 of the anomaly injection metrics (*scale*, *noise*, *cutoff*, *contextual* and *speedup*) to generally outperform other surrogate metrics across all datasets (Appendix A.2). We apply rank aggregation only to these 5 metrics. While this subset of surrogate metrics was identified using labels, it does not appear to depend on the dataset. Thus, given a new dataset without labels, we still propose to use rank aggregation over the same 5 surrogate metrics.

Pairwise-statistical tests. We carry out one-sided paired Wilcoxon signed rank tests at a significance level of $\alpha = 0.05$ to identify significant performance differences between *all pairs* of model selection strategies and baselines on each dataset.

Experimentation environment. All models and model selection algorithms were trained and built using `scikit-learn` 1.1.1 (Pedregosa et al., 2011), PyTorch 1.11.0 Paszke et al. (2019) along

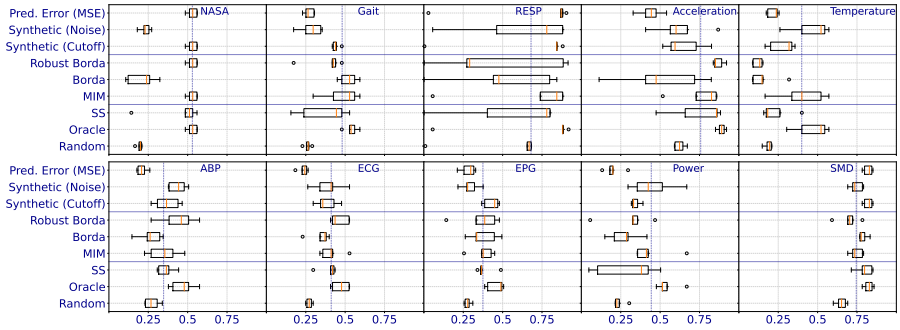

Figure 3: *Performance of the top-1 selected model.* Box plots of adjusted best F_1 of the model *selected* by each metric or baseline across 5 unique combinations of the selection and evaluation sets. Orange (|) and blue (|) vertical lines represent the median and average performance of each metric or baseline, and the minimum influence metric (MIM), respectively. Each box plot is organized into 3 sections: performance of individual metrics, Borda and its robust variations, and three baselines.

with Python 3.9.13. All our experiments were carried out on an AWS `g3.4xlarge` EC2 instance with with 16 Intel(R) Xeon(R) CPUs, 122 GiB RAM and a 8 GiB GPU.

5.4 Results and Discussion

A subset of our results are shown in Figure 3 and Table 2. Here we provide results for model selection using some of the individual surrogate metrics, as well as three variations of rank aggregation: Borda, robust Borda and a special case of robust Borda where we select only one metric, the one that minimizes empirical influence, "*minimum influence metric*" or MIM (Section 4). The complete set of results with all ranking metrics can be found in Appendix A.2.

No single best surrogate metric. Our results show that there is no surrogate metric which consistently selects the best model on all datasets. For instance, as can be seen in Figure 3, model selection based on prediction error (MSE) performs almost on par with the oracle on NASA, respiration (RESP) and server machine dataset (SMD), but worse than random model selection on the electrocardiogram (ECG), Arterial Blood Pressure (ABP) and acceleration datasets.

Ranking Methods/Baselines	Significant Wins		Significant Losses		
	SS	Random	Oracle	SS	Random
Pred. Error (MSE)	1	3	8	4	4
Syn. (Noise)	**2**	5	6	0	0
Syn. (Cutoff)	1	**8**	8	0	0
Syn. (Scale)	1	**8**	7	0	0
Syn. (Contextual)	1	5	7	0	0
Syn. (Average)	0	6	7	1	0
Robust Borda	0	4	**6**	2	0
Borda	0	4	9	3	0
MIM	1	**7**	**6**	0	0
SS	n/a	4	7	n/a	0
Oracle	7	10	n/a	0	0
Random	0	n/a	10	4	n/a

Table 2: *Results of pairwise-statistical tests.* On a total of $n = 10$ datasets, minimum influence metric (MIM) and robust Borda have the fewest significant losses to oracle, and are considerably better than random model selection and Borda.

Minimum Influence Metric perform on par with Supervised Selection. MIM is never significantly worse than SS and is in fact better on the air temperature (Temperature) dataset. Robust Borda on the other hand, is on par with SS on **8** datasets and significantly worse only on Temperature and SMD.

Robust variations of Borda outperform Borda. Aggregating via robust Borda and MIM are significantly better than Borda aggregation on **4** and **3** datasets, respectively. While MIM is never significantly worse than Borda, Robust Borda is inferior to Borda on SMD.

Robust variations of Borda minimizes losses to Oracle. From Table 2 it is evident that MIM and Robust borda have fewer significant losses to the oracle in comparison to the 5 surrogate metrics used as their input, and compared to SS. Thus, in general, robust aggregation is better than any randomly chosen individual surrogate metric. In terms of significant improvements over supervised and random selection, we found MIM to be a better alternative than Robust Borda.

Synthetic anomaly injection metrics outperform other classes of metrics. We found synthetic anomaly injection to perform better than prediction error and centrality metrics (Appendix A.2). Among the three surrogate classes, centrality metrics performed the worst, most likely due to highly correlated bad rankings.

Prior knowledge of anomalies might help identify good anomaly injection metrics. For instance, synthetic metrics such as *speedup* and *cutoff* perform well on respiration data since these datasets are likely to have such anomalies. The slowing down of respiration rate is a common anomaly and indicative of sleep apnea[2]. Similarly, synthetic noise performs well on air temperature dataset, likely because time-series have noise anomalies to simulate the sudden failure of temperature sensors. We also hypothesize that the superior performance of synthetic noise might be due to the intrinsic noise and distortion added to 100 of the 250 UCR datasets (Wu & Keogh, 2021), thereby allowing the metric to choose models which are better suited to handle the noise. In practice, domain experts are aware of the type of anomalies that might exist in the data and using our anomaly injection methods; this prior knowledge can be effectively leveraged for model selection.

6 Related Work

Time-series Anomaly Detection. The field has a long and rich history with well over a hundred proposed algorithms, differing in scope (e.g., multivariate vs univariate, unsupervised vs supervised), detection approach (e.g., forecasting, reconstruction, tree-based, etc.), etc. However, recent surveys have found that there is no single universally best method and the choice of the best method depends on the dataset (Schmidl et al., 2022; Blázquez-García et al., 2021).

Meta Learning and Model Selection. These approaches aim to identify the best model for a given dataset based on its characteristics such as the number of classes, attributes, instances etc. Recently, Zhao et al. (2021) explored the use of meta-learning to identify the best outlier detection algorithm on tabular datasets. For time-series datasets, both Ying et al. (2020) and Zhang et al. (2021) also approached model selection as meta-learning problem. All these studies rely on historical performance of models on a subset of "meta-train" datasets to essentially learn a mapping from the characteristics of a dataset ("meta-features") to model performance. These methods fail if anomaly labels are unavailable for meta-train datasets, or if the meta-train datasets are not representative of test datasets. Both these assumptions are frequently violated in practice, thus limiting the utility of such methods. In our approach, we do not assume access to similar datasets with anomaly labels.

Weak Supervision and Rank Aggregation. Our work draws on intuition from a large body of work on learning from weak or noisy labels since predictions of each surrogate metric have inherent noise and biases. We refer the reader to recent surveys on programmatic weak supervision, e.g. (Zhang et al., 2022). Since we are interested in ranking anomaly detection methods rather than inferring their exact performance numbers, rank aggregation is a much more relevant paradigm for our problem setting.

An extended version of Related Work is provided in Appendix A.1.

7 Conclusion

We consider the problem of unsupervised model selection for anomaly detection in time-series. Our model selection approach is based on surrogate metrics which correlate with model performance to varying degrees but do not require anomaly labels. We identify three classes of surrogate metrics namely, *prediction error*, *synthetic anomaly injection*, and *model centrality*. We devise effective procedures to inject different kinds of synthetic anomalies and extend the idea of model centrality to time-series. Next, we propose to combine rankings from different metrics using a robust rank aggregation approach and provide theoretical justification for rank aggregation. Our evaluation using over 5000 trained models and 17 surrogate metrics shows that our proposed approach performs on par with selection based on partial data labeling. Future work might focus on other definitions of model centrality and include more synthetic anomalies types.

[2]Sleep apnea is a common and potentially serious health condition in which breathing repeatedly stops and starts (`https://www.nhlbi.nih.gov/health/sleep-apnea`).

REFERENCES

Kalousis Alexandros and Hilario Melanie. Model selection via meta-learning: a comparative study. *International Journal on Artificial Intelligence Tools*, 10(04):525–554, 2001.

Alnur Ali and Marina Meilă. Experiments with kemeny ranking: What works when? *Mathematical Social Sciences*, 64(1):28–40, 2012.

Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.

JC de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)*, 1784.

Chris U Carmona, François-Xavier Aubet, Valentin Flunkert, and Jan Gasthaus. Neural contextual anomaly detection for time series. *arXiv preprint arXiv:2107.07702*, 2021.

Cristian I Challu, Peihong Jiang, Ying Nian Wu, and Laurent Callot. Deep generative model with hierarchical latent factors for time series anomaly detection. In *International Conference on Artificial Intelligence and Statistics*, pp. 1643–1654. PMLR, 2022.

Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. *Advances in neural information processing systems*, 30, 2017.

Fabien Collas and Ekhine Irurozki. Concentric mixtures of mallows models for top-k rankings: sampling and identifiability. In *International Conference on Machine Learning*, pp. 2079–2088. PMLR, 2021.

Vincent Conitzer, Andrew Davenport, and Jayant Kalagnanam. Improved bounds for computing kemeny rankings. In *AAAI*, volume 6, pp. 620–626, 2006.

R Dennis Cook and Sanford Weisberg. Characterizations of an empirical influence function for detecting influential cases in regression. *Technometrics*, 22(4):495–508, 1980.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28 (1):20–28, 1979.

Arnab Dey, Mononito Goswami, Joo Heung Yoon, Gilles Clermont, Michael Pinsky, Marilyn Hravnak, and Artur Dubrawski. Weakly supervised classification of vital sign alerts as real or artifact. *arXiv preprint arXiv:2206.09074*, 2022.

Sunny Duan, Loic Matthey, Andre Saraiva, Nicholas Watters, Christopher P Burgess, Alexander Lerchner, and Irina Higgins. Unsupervised model selection for variational disentangled representation learning. *arXiv preprint arXiv:1905.12614*, 2019.

Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pp. 613–622, 2001.

Ronald Fagin, Ravi Kumar, and Dakshinamurthi Sivakumar. Comparing top k lists. *SIAM Journal on discrete mathematics*, 17(1):134–160, 2003.

Michael A Fligner and Joseph S Verducci. Distance based ranking models. *Journal of the Royal Statistical Society: Series B (Methodological)*, 48(3):359–369, 1986.

Michael A Fligner and Joseph S Verducci. Multistage ranking models. *Journal of the American Statistical association*, 83(403):892–901, 1988.

Chufan Gao, Mononito Goswami, Jieshi Chen, and Artur Dubrawski. Classifying unstructured clinical notes via automatic weak supervision. *arXiv preprint arXiv:2206.12088*, 2022.

Nicolas Goix. How to evaluate the quality of unsupervised anomaly detection algorithms? *arXiv preprint arXiv:1607.01152*, 2016.

Mononito Goswami, Benedikt Boecking, and Artur Dubrawski. Weak supervision for affordable modeling of electrocardiogram data. In *AMIA Annual Symposium Proceedings*, volume 2021, pp. 536. American Medical Informatics Association, 2021.

Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In *International conference on machine learning*, pp. 2712–2721. PMLR, 2016.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

Eamonn Keogh, Jessica Lin, and Ada Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 8–pp. Ieee, 2005.

Anna Korba, Stephan Clémençon, and Eric Sibony. A learning theory of ranking aggregation. In *Artificial Intelligence and Statistics*, pp. 1001–1010. PMLR, 2017.

Nikolay Laptev, Saeed Amizadeh, and Ian Flint. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1939–1947, 2015.

Zinan Lin, Kiran Thekumparampil, Giulia Fanti, and Sewoong Oh. Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *international conference on machine learning*, pp. 6127–6139. PMLR, 2020.

Martin Q Ma, Yue Zhao, Xiaorong Zhang, and Leman Akoglu. A large-scale study on unsupervised outlier model selection: Do internal strategies suffice? *arXiv preprint arXiv:2104.01422*, 2021.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. Long short term memory networks for anomaly detection in time series. In *Proceedings*, volume 89, pp. 89–94, 2015.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438, 2000.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. *Advances in neural information processing systems*, 29, 2016.

Quentin Rebjock, Baris Kurt, Tim Januschowski, and Laurent Callot. Online false discovery rate control for anomaly detection in time series. *Advances in Neural Information Processing Systems*, 34:26487–26498, 2021.

Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 2828–2837, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330672. URL https://doi.org/10.1145/3292500.3330672.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pp. 25–54. PMLR, 2013.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

Renjie Wu and Eamonn Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*, pp. 187–196, 2018.

Yuanxiang Ying, Juanyong Duan, Chunlei Wang, Yujing Wang, Congrui Huang, and Bixiong Xu. Automated model selection for time-series anomaly detection. *arXiv preprint arXiv:2009.04395*, 2020.

Jieyu Zhang, Cheng-Yu Hsieh, Yue Yu, Chao Zhang, and Alexander Ratner. A survey on programmatic weak supervision. *arXiv preprint arXiv:2202.05433*, 2022.

Peiyi Zhang, Xiaodong Jiang, Ginger M Holt, Nikolay Pavlovich Laptev, Caner Komurlu, Peng Gao, and Yang Yu. Self-supervised learning for fast and scalable time series hyper-parameter tuning. *arXiv preprint arXiv:2102.05740*, 2021.

Yue Zhao, Ryan Rossi, and Leman Akoglu. Automatic unsupervised outlier model selection. *Advances in Neural Information Processing Systems*, 34:4489–4502, 2021.

A APPENDIX

A.1 RELATED WORK

Time-series Anomaly Detection. This field has a long and rich history with well over a hundred proposed algorithms (Schmidl et al. (2022); Blázquez-García et al. (2021)). The methods vary in their scope (e.g., multivariate vs univariate, unsupervised vs supervised), and in the anomaly detection approach (e.g., forecasting, reconstruction, tree-based, etc.). At a high level many of these methods learn a model for the "normal" time-series regime, and identify points unlikely under the given model as anomalies. For example, progression of a time-series can be described using a forecasting model, and points deviating from the forecast can be flagged as outliers (Malhotra et al. (2015)). In another example, for tree-based methods, a data model is an ensemble of decision trees that recursively partition the points (sub-sequences of the time-series), until each sample is isolated. Abnormal points are easier to separate from the rest of the data, and therefore these will tend to have shorter distances from the root (Guha et al. (2016)). For a comprehensive overview of the field, we refer the reader to one of the recent surveys on anomaly detection methods and their evaluation (Schmidl et al. (2022); Blázquez-García et al. (2021); Wu & Keogh (2021)). Importantly, there is no universally best method, and the choice of the best method depends on the data at hand (Schmidl et al. (2022)).

Meta Learning and Model Selection. These approaches aim to identify the best model for the given dataset. The use of meta learning for model selection is a long-standing idea. For instance, Kalousis and Hilario used boosted decision trees to predict (propose) classifiers for a new dataset given data characteristics "meta-features") such as the number of classes, attributes, instances etc (Alexandros & Melanie, 2001)). More recently, Zhao et al. (2021) explored the use of meta-learning for outlier detection algorithms. Their algorithm relies on a collection of historical outlier detection datasets with ground-truth (anomaly) labels and historical performance of models on these meta-train datasets. The authors leverage meta-features of datasets to build a model that "recommends" an anomaly detection method for a dataset.

Ying et al. (2020) focused on model selection at the level of time-series. They characterize each time-series based on a fixed set of features, and use anomaly-labeled time-series from a knowledge base to train a classifier for model selection and a regressor for hyper-parameter estimation. A similar approach is taken by Zhang et al. (2021) who extracted time-series features, identified best performed models via exhaustive hyper-parameter tuning, and trained a classifier (or regressor) using the extracted time-series features and information about model performance as labels.

In contrast to methods mentioned in this sub-section, we are interested in unsupervised model selection. In particular, given an unlabeled dataset, we do not assume access to a similar dataset with anomaly labels.

Weak Supervision and Rank Aggregation. We consider a number of unsupervised metrics, and these can be viewed as weak labels or noisy human labelers. The question is then which metric to use or how to aggregate the metrics. There has been a substantial body of work on reconciling weak labels, mostly in the context of classification, and to a lesser extend for regression. Many such methods build a joint generative model of unlabeled data and weak labels to infer latent true labels. We refer the interested reader to a recent survey by Zhang et al. (2022).

Since we are interested in ranking anomaly detection methods rather than inferring their exact performance numbers, rank aggregation is a more relevant paradigm for our project. Rank aggregation concerns with producing a consensus ranking given a number of potentially inconsistent rankings (permutations). One of the dominant approaches in the field involves defining a (pseudo-)distance in the space of permutations and then finding a barycentric ranking, i.e., the one that minimizes distance to other rankings (Korba et al. (2017)). The distance distribution can then be modeled, e.g., using the Mallows model, and rank aggregation reliability can be added by assuming a mixture from Mallows distributions with different dispersion parameters (Collas & Irurozki (2021)). In our work, we draw on the ideas from this research.

A.2 ADDITIONAL RESULTS

The complete set of results are shown in Fig. 4 and Tab. 3.

We compare **17** individual surrogate metrics, including **5** prediction error metrics, **9** synthetic anomaly injection metrics, **3** metrics of centrality (average distance to 1, 3 and 5 nearest neighbors) (Section3); and **4** proposed robust variations of Borda rank aggregation, namely *Partial Borda*, *Trimmed Borda*, *Minimum Influence Metric* and *Robust Borda* (see Appendix A.6.2). We also compare Borda and its variations with exact Kemeny aggregation Appendix A.6.3).

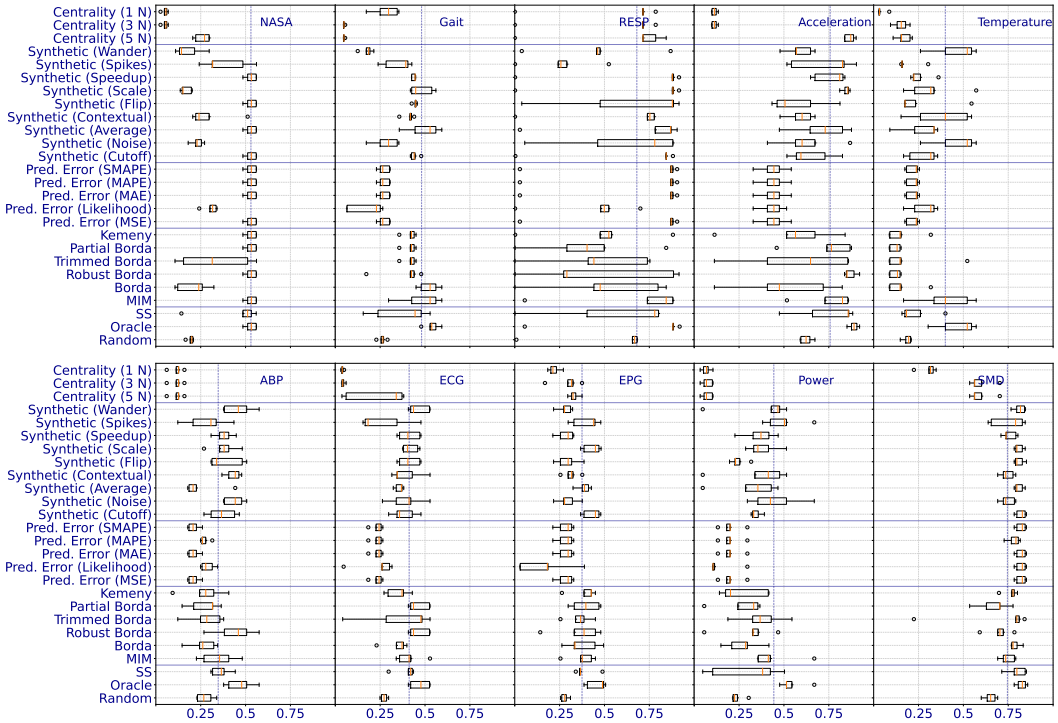

Figure 4: *Performance of the top-1 selected model*. Box plots of adjusted best F_1 of the model *selected* by each metric or baseline across 5 unique combinations of the selection and evaluation sets. Orange (|) and blue (|) vertical lines represent the median and average performance of each metric or baseline, and the minimum influence metric (`MIM`), respectively. Each box plot is organized into three sections: performance of individual metrics, Borda and its robust variations, and three baselines.

Ranking Methods/Baselines	Significant Wins		Significant Losses		
	SS	Random	Oracle	SS	Random
Random	0	0	10	4	0
Oracle	7	10	0	0	0
SS	0	4	7	0	0
MIM	1	7	6	0	0
Borda	0	3	9	3	0
Robust Borda	0	4	6	2	1
Trimmed Borda	0	1	8	0	0
Partial Borda	0	4	8	3	1
Kemeny	0	4	9	1	0
Pred. Error (MSE)	1	3	8	4	4
Pred. Error (Likelihood)	0	3	9	3	2
Pred. Error (MAE)	1	3	8	4	4
Pred. Error (MAPE)	1	3	9	4	3
Pred. Error (SMAPE)	1	3	8	4	4
Synthetic (Cutoff)	1	8	8	0	0
Synthetic (Noise)	2	5	6	0	0
Synthetic (Average)	0	6	7	1	0
Synthetic (Contextual)	1	5	7	0	0
Synthetic (Flip)	0	5	8	0	0
Synthetic (Scale)	1	8	7	0	0
Synthetic (Speedup)	1	6	9	1	0
Synthetic (Spikes)	0	3	8	1	1
Synthetic (Wander)	1	4	6	1	1
Centrality (5 N)	0	3	10	4	3
Centrality (3 N)	0	0	10	6	7
Centrality (1 N)	0	0	10	7	7

Table 3: *Results of pairwise-statistical tests.*. On a total of $n = 10$ datasets, minimum influence metric (MIM) and robust Borda have the fewest significant losses to oracle, and are much better than random model selection and Borda.

A.3 Synthetic Data Experiments

The Mallows's model (Fligner & Verducci, 1986) is a popular distance-based distribution to model rank data. The probability mass function of a Mallows's model with central permutation σ_0 and dispersion parameter θ is given by:

$$p(\sigma) = \frac{exp(-\theta d(\sigma, \sigma_0))}{\psi}$$

where d is the Kendall's τ distance. With increase in θ, there is stronger consensus, thereby making Kemeny aggregation easier.

We aim to empirically answer two research questions:

1. What is the impact of noisy permutations on the median rank?
2. Can empirical influence identify outlying permutations?

To this end, we draw k permutations from the Mallow's distribution $\sigma_1, \cdots \sigma_k \sim \mathcal{M}(\sigma_0, \theta)$ and $N - k$ permutations at uniformly at random from S_n. We compute the median rank $\hat{\sigma}$ using the Borda method. We experiment with $n \in \{50, 100\}$, $\theta \in \{0.05,\ 0.1,\ 0.2\}$ and $N = 50$ based on the experimental setup used by Ali & Meilă (2012).

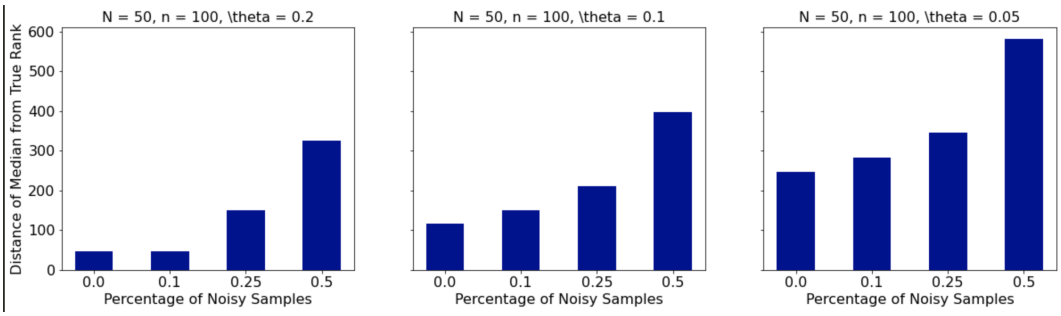

Figure 5: *Impact of Noisy Permutations*. With increase in noise, the distance of the median permutation ($\hat{\sigma}$) from the central permutation (σ_0) increase *i.e.* noise hurts rank aggregation.

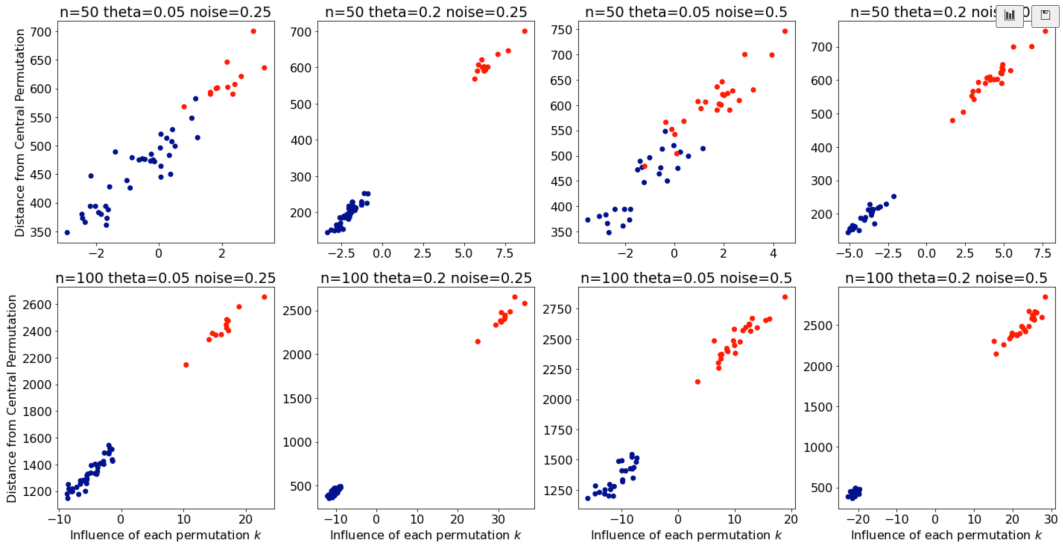

Figure 6: *Empirical Influence can Identify Outlying Permutations.* Influence of a permutation is directly proportional to its distance from the central permutation (σ_0). Under the Mallow's model, outlying (or low probability) permutations have a higher distance (σ_0).

A.4 Generating Synthetic Anomalies

Natural anomalies do not occur randomly, but are determined by the periodic nature of time-series. Therefore, before injecting anomalies we determine periodicity of time-series by finding the its auto-correlation, and only inject anomalies (barring spikes which are point-outliers) at beginning of a cycle. We only inject a single anomaly in each time-series in a repetition, and its length is chosen uniformly at random from `[1, max_length]`, the latter being a user defined parameter (Fig. 7).

Spike Anomaly. To introduce a spike, we first draw $m \sim \texttt{Bernoulli}(p)$, and if $m = 1$ we inject a spike whose magnitude is governed by a normal distribution $s \sim \mathcal{N}(0, \sigma^2)$.

Flip Anomaly. We reverse the time-series:

```
ts_a[i, start:end] = ts[i, start:end][::-1]
```

Speedup Anomaly. We increase or decrease the frequency of a time-series using interpolation, where the factor of change in frequency ($\frac{1}{2}$ or 2) is a user-defined parameter.

Noise Anomaly. We inject noise drawn from a normal distribution $\mathcal{N}(0, \sigma^2)$, where σ is a user-defined parameter.

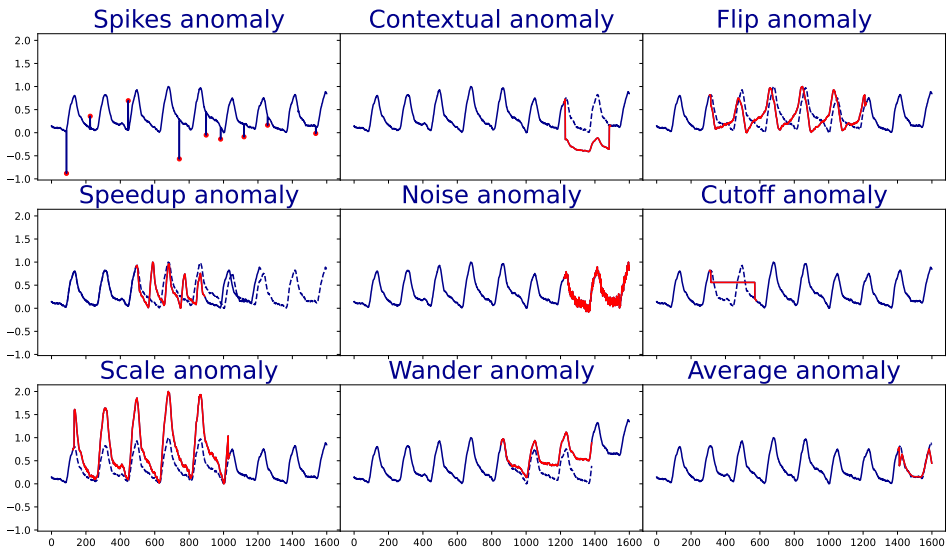

Figure 7: *Different kinds of synthetic anomalies*. We inject 9 different types of synthetic anomalies randomly, one at a time, across multiple copies of the original time-series. The - - is original time-series before anomalies are injected, – is the injected anomaly and the solid – is the time-series after anomaly injection.

Cutoff Anomaly Here, we set the time-series to either $\mathcal{N}(0, \sigma^2)$ or $\mathcal{N}(1, \sigma^2)$, where both the choice of location parameter and σ is user-defined.

Average Anomaly. We run a moving average where the length of the window (`len_window`) is determined by the user:

```
ts_a[i, start:end] = moving_average(ts[i, start:end], len_window)
```

Scale Anomaly. We scale the time-series by a user-defined factor (`factor`):

```
ts_a[i, start:end] = factor*ts[i, start:end]
```

Wander Anomaly. We add a linear trend with the increase in baseline (`baseline`) defined by users:

```
ts_a[i, start:end] = np.linspace(0, baseline, end-start) + ts[i, start:end]
```

Contextual Anomaly. We convolve a the time-series subsequence linear function $aY + b$, where $a \sim \mathcal{N}(1, \sigma_a^2)$ and $b \sim \mathcal{N}(0, \sigma_b^2)$

A.5 DESCRIPTION OF MODELS

There are over a hundred unique time-series anomaly detection algorithms developed using a wide variety of approaches. For our experiments, we implemented a representative subset of 5 popular methods (Table 1), across two different learning types (unsupervised and semi-supervised) and three different method families (forecasting, reconstruction and distance-based) (Schmidl et al., 2022).

In the context of time-series anomaly detection, "semi-supervised" means requiring a part of time-series known to be anomaly-free. Each time-series in the UCR and SMD collections comprises of a train part with no anomalies and a test part with one (UCR) or more anomalies (SMD). We used the train part for training semi-supervised models.

We emphasize that our surrogate metrics do not rely on availability of the "clean" train part (and we didn't use the train part for evaluation). Moreover, while semi-supervised models are *supposed* to be trained on anomaly-free part, in practice they can still work (perhaps sub-optimally) after training on time-series that potentially contained anomalies.

The models we implemented can also be categorized as forecasting, reconstruction and distance methods. Forecasting methods learn to forecast the next few time steps based on the current context window, whereas reconstruction methods encode observed sub-sequences into a low dimensional latent space and reconstruct them from the latent space. In both cases, the predictions (forecasts or reconstruction) of the models are compared with the observed values. Finally, distance based models used distance metrics (e.g. Euclidean distance) to compare sub-sequences, and expect anomalous sub-sequences to have large distances to sub-sequences they are most similar to.

We created a pool of **3** k-NN (Ramaswamy et al., 2000), and **4** moving average, **4** DGHL (Challu et al., 2022), **4** LSTM-VAE (Park et al., 2018) and **4** RNN (Chang et al., 2017) by varying important hyper-parameters, for a total **19** models. Finally, to efficiently train our models, we sub-sampled all time-series with $T > 2560$ by a factor of 10.

Model	Area	Learning Type	Method Family
`Moving Average`	Statistics	Unsupervised	Forecasting
k-`NN` (Ramaswamy et al., 2000)	Classical ML	Semi-supervised	Distance
`Dilated RNN` (Chang et al., 2017)	Deep Learning	Semi-supervised	Forecasting
`DGHL` (Challu et al., 2022)	Deep Learning	Semi-supervised	Reconstruction
`LSTM-VAE` (Park et al., 2018)	Deep Learning	Semi-supervised	Reconstruction

Table 4: *Models in the model set and their properties*.

Below is brief description of each model.

Moving Average. These models compare observations with the "moving" average of recent observations in a sliding window of predefined size. These methods assume that if the current observation differs significantly from moving average, then it must be an anomaly. Specifically, if $\boldsymbol{x}_t$ denotes the current observation, h represents the window size and let τ be an anomaly threshold, then $\boldsymbol{x}_t$ is flagged as an anomaly if:

$$\left|\left|\boldsymbol{x}_t - \frac{1}{h}\sum_{i=t-h-1}^{t-1} \boldsymbol{x}_i\right|\right|_2^2 > \tau,$$

Moving average is one of the simplest, efficient yet practical anomaly detection models, frequently used by industry practitioners. Moreover, moving average is unsupervised since it does not require any normal training data.

k-Nearest Neighbors (k-NN) (Ramaswamy et al., 2000). These distance-based anomaly detection models compare a window of observations to their k nearest neighbour windows in the training data. If the current window is significantly far its k closest windows on the training set, then it must be an outlier. In our implementation, the distance between time-series windows is measured using standard Euclidean distance. Specifically, let $\boldsymbol{w}_t = \{\boldsymbol{x}_{t-h}, \cdots, \boldsymbol{x}_t\}$ denote the current window of size h, $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_k$ be the k nearest windows in the train set to $\boldsymbol{w}_t$, $||\cdot||_2$ represent the euclidean

norm, and τ be an anomaly threshold. Then the current window is flagged as an anomaly if:

$$\frac{1}{k}\sum_{i=1}^{k}||\boldsymbol{w}_i - \boldsymbol{w}_t||_2^2 > \tau$$

Dilated RNN (Chang et al., 2017). Dilated Recurrent neural network (RNN) is a recently proposed improvement over vanilla RNNs, characterized by multi-resolution dilated recurrent skip connections. These skip connections alleviate problems arising from memorizing long term dependencies without forgetting short term dependencies, vanishing and exploding gradients, and the sequential nature of forward and back-propagation in vanilla RNNs. Given a historical window of time-series, we use a dilated RNN to forecast T_{la} time steps ahead. Let $w_t = \{\boldsymbol{x}_t, \boldsymbol{x}_{T_{la}}\}$ denote the current window of observations, $\hat{\boldsymbol{w}}_t = \{\hat{\boldsymbol{x}}_t, \cdots, \hat{\boldsymbol{x}}_{T_{la}}\}$ be the forecasts from the dilated RNN model, and τ be an anomaly threshold. Then $\boldsymbol{x}_t$ is flagged as an anomaly if:

$$||\hat{\boldsymbol{w}}_t - \boldsymbol{w}_t||_2^2 > \tau$$

DGHL (Challu et al., 2022). This model is a recent state-of-the-art Generative model for time-series anomaly detection. It uses a Top-Down Convolution Network (CNN) to map a novel hierarchical latent representation to a multivariate time-series windows. The key difference with other generative models as GANs and VAEs is that is trained with the Alternating Back-Propagation algorithm. The architecture does not include an encoder, instead, latent vectors are directly sampled from the posterior distribution. Let $\boldsymbol{x}_t$ be the observation and $\hat{\boldsymbol{x}}_t$ the reconstruction at timestamp t, and τ be an anomaly threshold. Then $\boldsymbol{x}_t$ is flagged as an anomaly if:

$$||\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t||_2^2 > \tau$$

LSTM-VAE (Park et al., 2018). The LSTM-VAE is a reconstruction based model that uses a Variational Autoencoder with LSTM components to model the probability of observing the target time-series. The model also uses the latent representation to dynamically adapt the threshold to flag anomalies based on the normal variability of the data. The anomaly scores corresponds to the negative log-likelihood of an observation, $\boldsymbol{x}_t$, based on the output distribution of the model, and it is flagged as an anomaly if the score is larger than a threshold τ:

$$-\log p(\boldsymbol{x}_t) > \tau$$

A.6 BORDA RANK AGGREGATION

Borda is a positional ranking system and is an unbiased estimator of the Kemeny ranking of a sample distributed according to the Mallows Model (Fligner & Verducci, 1988). Consider N models and a collection of M surrogate metrics $\mathcal{S} = \{\sigma_1, \cdots, \sigma_M\}$. We use σ_k to both refer to the surrogate metric and the ranking it induces. Then under the Borda scheme, for any given metric σ_k, the ith model receives $N - \sigma_k(i)$ points. Thus, the ith model accrues a total of $\sum_{k=1}^{M}(N - \sigma_k(i))$ points. Finally, all the models are ranked in decreasing order of their total points and ties are broken uniformly at random. Borda can be computed in quasi-linear ($\mathcal{O}(M + N \log N)$) time.

A.6.1 IMPROVING RANKING PERFORMANCE AT THE TOP

In model selection, we only care about top-ranked models. Thus, to improve model selection performance at the top-ranks, we only consider models at the top-k ranks, setting the positions of the rest of the models to N (Fagin et al., 2003). Here, k is a user-specified hyper-parameter. Specifically, under the *top-k* Borda scheme, the ith model accrues a total of $\sum_{m=1}^{M} \mathbb{I}[\sigma_m(i) \leq k](N - \sigma_m(i))$ points. Intuitively, this increases the probability of models which have top-k ranks consistently across surrogate metrics, to have top ranks upon aggregation.

A.6.2 ROBUST VARIATIONS OF BORDA

We consider multiple variations of Borda, namely *Partial Borda*, *Trimmed Borda*, *Minimum Influence Metric* (MIM) and *Robust Borda*. We collectively refer to these as robust variations of Borda rank aggregation.

Partial Borda improves model selection performance, especially at the top ranks by only considering models at the top-k ranks, as described in Appendix A.6.1.

Trimmed Borda is another robust variation of Borda, which only aggregates reliable ("good") permutations. Trimmed Borda identifies outlying or "bad" permutations as ones having high positive empirical influence.

Minimum Influence Metric (MIM) for a dataset is the surrogate metric with minimum influence. Recall that while high positive influence is indicative of "bad" permutations, low values of influence are indicative of "good" permutations.

Robust Borda performs rank aggregation only based on the top-k ranks while truncating permutations with high positive influence. Hence, Partial Borda, Trimmed Borda and Minimum Influence Metric can be viewed as special cases of Robust Borda.

A.6.3 EXACT KEMENY RANK AGGREGATION

The Kemeny rank aggregation problem can be viewed as a minimization problem on a weighted directed graph (Conitzer et al., 2006). Let $G = (V, E)$ denote a weighted directed graph with the models as vertices. For every pair of models, we define an edge $(i \rightarrow j) \in E$ and set its weight as $w_{ij} = |\sum_{k=1}^{N} \mathbb{I}_{i \succcurlyeq_k j} - \sum_{k=1}^{N} \mathbb{I}_{j \succcurlyeq_k i}|$. Here the indicator $\mathbb{I}_{i \succcurlyeq_k j} = 1$ denotes that metric k prefers model i over model j, and hence w_{ij} denotes the number of metrics which prefer model i over j.

Then we can formulate the rank aggregation problem as a binary linear program:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} w_e x_e \\
\text{subject to} \quad & \forall i \neq j \in V, x_{ij} + x_{ji} = 1 \quad \text{(Anti-symmetry and totality constraints)} \\
& \forall i \neq j \neq k \in V, x_{ij} + x_{jk} + x_{ki} \geq 1 \quad \text{(Transitivity constraints)} \\
& \forall i \neq j, x_{ij} \in \{0, 1\} \quad \text{(Binary constraint)}
\end{aligned}
$$

Intuitively, we incur a cost for every metric's pairwise comparison that we do not honor.

A.7 THEORETICAL JUSTIFICATION FOR BORDA RANK AGGREGATION

Suppose that we have several surrogate metrics for ranking our anomaly detection models. Also suppose that each of these metrics is better than random ranking. Why would it be beneficial to aggregate the rankings induced by these metrics? Theorem 1 provides an insight into the benefit of Borda rank aggregation.

Suppose that we have N anomaly detection models to rank. Consider two arbitrary models i and j with (unknown) ground truth rankings $\sigma^*(i)$ and $\sigma^*(j)$. Without the loss of generality, assume $\sigma^*(i) < \sigma^*(j)$, i.e., model i is better.

Next, let's view each surrogate metric $k = 1, \ldots, M$ as a random variable Ω_k taking values in permutations S_N (a set of possible permutations over N items), such that $\Omega_k(i) \in [N]$ denotes the rank of item i. We are interested in properties of Borda rank aggregation applied on realizations of these random variables. Theorem 1 states that, under certain assumptions, the probability of Borda ranking making a mistake (i.e., placing model i lower than j) is upper bounded in a way such that increasing M decreases this bound.

Theorem. *Assume that Ω_k's are pairwise independent, and that $\mathbb{P}(\Omega_k(i) < \Omega_k(j)) > 0.5$ for any arbitrary models i and j, i.e., each surrogate metric is better than random. Then the probability that Borda aggregation makes a mistake in ranking models i and j is at most $2\exp\left(-\frac{M\epsilon^2}{2}\right)$, for some fixed ϵ.*

Proof. We begin by defining random variables $X_k = 2\mathbb{I}[\Omega_k(i) < \Omega_k(j)] - 1$. Thus $X_k = 1$ if $\Omega_k(i) < \Omega_k(j)$ and $X_k = -1$, otherwise. Denote the expectation of these random variables as $E_k \triangleq \mathbb{E}[X_k]$. We assume that $E_k > \epsilon,\ \forall k \in [M]$ for some small $\epsilon > 0$, i.e., that each surrogate metric k is better than random.

Next, define $S_M \triangleq \sum_{k=1}^{M} X_k$, $E_M = \sum_{k=1}^{M} E_k$. Note that if $S_M > 0$ then Borda aggregation will correctly rank i higher than j.

If Ω_k are pairwise independent, then X_k are pairwise independent, and we can apply Hoeffding's inequality for all $t > 0$:

$$
\begin{aligned}
\mathbb{P}(|S_M - E_M| \geq t) &\leq 2\exp(-\frac{2t^2}{4M}) \\
\mathbb{P}(E_M - S_M \geq t) &\leq 2\exp(-\frac{t^2}{2M}) \\
\mathbb{P}(S_M \leq 0) &\leq 2\exp(-\frac{(E_M)^2}{2M}), \quad \text{setting } t = E_M \\
\mathbb{P}(S_M \leq 0) &\leq 2\exp(-\frac{M\epsilon^2}{2}), \quad \text{because } \epsilon < E_k, \forall k
\end{aligned}
$$

Recall, that $S_M < 0$ results in an incorrect ranking (ranking j higher than i), and $S_M = 0$ results in a tie (that we break uniformly at random). Therefore, the probability that Borda aggregation makes a mistake in ranking items i and j is upper bounded as per the above statement.

□

We do not assume that Ω_k are identically distributed, but we assume that Ω_k's are pairwise independent. We emphasize that the notion of independence of Ω_k as random variables is different from rank correlation between realizations of these variables (i.e., permutations). For example, two permutations drawn independently from a Mallow's distribution can have a high Kendall's τ coefficient (rank correlation). Therefore, our assumption of independence, commonly adopted in theoretical analysis (Ratner et al., 2016), does not contradict our observation of high rank correlation between surrogate metrics.

Corollary 1. *Under the same assumptions as Theorem 1, if Borda aggregation makes a mistake in ranking models i and j with probability at most $\delta > 0$, then $\min_k \mathbb{P}(\Omega_k(i) < \Omega_k(j)) > \sqrt{\frac{\log\frac{2}{\delta}}{2M}} + \frac{1}{2}$, where $k \in [M]$ and $i, j \in [N]$ are any two arbitrary models.*

Proof. The proof follows from upper bounding the probability that Borda makes a mistake in ranking models i and j by δ:

$$\mathbb{P}(S_M \leq 0) \leq 2\exp(-\frac{M\epsilon^2}{2}) \leq \delta \implies \epsilon \geq \sqrt{\frac{2\log\frac{2}{\delta}}{M}} \tag{3}$$

Consider the following relationship between ϵ and $\min_k \mathbb{P}(\Omega_k(i) < \Omega_k(j)))$ for $k \in [M]$:

$$\epsilon \leq \min_k E_k = \min_k 2\mathbb{P}(\Omega_k(i) < \Omega_k(j)) - 1 = 2(\min_k \mathbb{P}(\Omega_k(i) < \Omega_k(j))) - 1 \tag{4}$$

From Equations 3 and 4, we can conclude that:

$$\min_k \mathbb{P}(\Omega_k(i) < \Omega_k(j)) \geq \sqrt{\frac{\log\frac{2}{\delta}}{2M}} + \frac{1}{2}$$

□

To reduce chances of error, it follows from Corollary 1 that we can either increase the number of surrogate metrics, or collect a smaller number of highly accurate metrics. As an example, set $\delta = 0.05$ and consider $M = 20$ metrics. Then for the Borda to err at most 5% of times, each surrogate metric must prefer A_i over A_j with probability at least 0.7. This also highlights the importance of empirical influence in identifying and removing bad permutations from the Borda aggregation.