$\mathbf{M^2AD}$: Multi-Sensor Multi-System Anomaly Detection through Global Scoring and Calibrated Thresholding

Sarah Alnegheimish[*]
MIT

Zelin He[*]
Pennsylvania State University

Matthew Reimherr[†]
Pennsylvania State University
Amazon

Akash Chandrayan
Amazon

Abhinav Pradhan
Amazon

Luca D'Angelo
Amazon

Abstract

With the widespread availability of sensor data across industrial and operational systems, we frequently encounter heterogeneous time series from multiple systems. Anomaly detection is crucial for such systems to facilitate predictive maintenance. However, most existing anomaly detection methods are designed for either univariate or single-system multivariate data, making them insufficient for these complex scenarios. To address this, we introduce $\mathrm{M^2AD}$, a framework for unsupervised anomaly detection in *multivariate* time series data from *multiple* systems. $\mathrm{M^2AD}$ employs deep models to capture expected behavior under normal conditions, using the residuals as indicators of potential anomalies. These residuals are then aggregated into a global anomaly score through a Gaussian Mixture Model and Gamma calibration. We theoretically demonstrate that this framework can effectively address heterogeneity and dependencies across sensors and systems. Empirically, $\mathrm{M^2AD}$ outperforms existing methods in extensive evaluations by 21% on average, and its effectiveness is demonstrated on a large-scale real-world case study on 130 assets in Amazon Fulfillment Centers. Our code and results are available at `https://github.com/sarahmish/M2AD`.

Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258. Copyright 2025 by the author(s).

1 Introduction

Over the past several years, there has been a tremendous shift towards sensor-based monitoring of mechanical and electrical assets, whether in manufacturing, healthcare, logistics, or energy management (Abdulmalek et al., 2022; Liu and Zhang, 2020; Tahan et al., 2017; Cao et al., 2019). These sensors generate a significant amount of data in real time, reinforcing the need for automated solutions for anomaly detection to reduce manual labor. Moreover, it emphasizes the importance of deploying models that are capable of processing large amounts of data concurrently and with high accuracy.

As the number of assets (machines) increases and data sources become more diverse, sensor-based monitoring provides richer, more complex information. For one, as the variety of sensors increases, we now have a multitude of metrics coming from various sources to assess the health condition of assets. For example, a conveyor belt in a warehouse is monitored through multiple sensors that measure temperature and vibrations at various locations of the belt, and the electric current needed by the motors to drive the conveyor. Moreover, additional measurements provide valuable insights into an asset's throughput and performance. For instance, in e-commerce settings such as Amazon's fulfillment centers, we can track how many items a machine processes. This information is particularly useful in domains where activity levels are highly seasonal and do not follow simple patterns. By integrating data from various systems, we gain a more comprehensive view of asset utilization and health, enabling more informed decisions about performance and maintenance.

Monitoring the condition of assets is crucial for ensuring reliable operation and minimizing unplanned

[*]Work done during Amazon internship.
[†]Corresponding author mreimherr@psu.edu.

downtime. In this context, *unsupervised* time series anomaly detection has emerged as a key approach to facilitate condition-based monitoring, allowing for the identification of anomalies without requiring previously labeled data. For example, Audibert et al. (2020) propose adversely trained autoencoders to generate anomaly scores, however, the authors apply manual thresholding which is both impractical and not scalable for real-world applications. On the other hand, Hundman et al. (2018) address this issue by developing a non-parametric thresholding model, yet it is designed for detecting anomalies in univariate time series, which could significantly increase the number of false alarms in multivariate data.

Detecting anomalies in multivariate time series presents additional challenges in real-world industrial settings. To be effective in practice, the model needs to meet key operational requirements. First, it must raise alarms judiciously to optimize technician workload and build trust in the system. Second, for diagnostic purposes, it is crucial to provide interpretability, allowing operators to identify which sensor in which system triggered the alarm. This gives the respective technician context and helps reduce the investigation time.

To address these challenges, we propose a multi-sensor multi-system anomaly detection framework, M^2AD, for time series data. The framework uses a prediction model to learn the expected pattern of the signal, we then use the discrepancies between the predicted and actual signals to compute a global anomaly score that factors in all sensors from available systems. The framework learns the appropriate threshold value estimated from anomaly scores seen during the training phase. Moreover, to preserve the element of interpretability, the system produces an ordered list of most contributing sensors to the anomaly score. We summarize our contributions below:

- Introduce **M^2AD**, an unsupervised **multi-sensor multi-system** time series **anomaly detection** method. M^2AD leverages an LSTM model to learn patterns from sensor observations and covariates, identifying discrepancies between expected and observed behavior. These sensor-wise discrepancies are aggregated into a global anomaly score through a novel combination of a Gaussian Mixture Model and Gamma calibration, effectively addressing heterogeneity and dependencies across sensors and systems.

- Provide **theoretical support** for M^2AD using learning theory, demonstrating that it provably reduces false detection and improves error distribution calibration, particularly in scenarios with dependencies and heterogeneity across sensors within multiple systems.

- Conduct a comprehensive **empirical evaluation** of M^2AD on three public multivariate time series datasets, benchmarking it against eight state-of-the-art deep learning models. Our model achieves a 21% improvement over existing methods. Additionally, we validate M^2AD through a real-world case study using proprietary data of 130 assets from an Amazon fulfillment center, evaluating its effectiveness.

2 Related Work

There has been a continuous effort to develop new unsupervised time-series anomaly detection methods using the latest models and architectures (Zamanzadeh Darban et al., 2024; Choi et al., 2021; Braei and Wagner, 2020; Kim et al., 2024). Relevant work can broadly be categorized into the following:

Univariate Anomaly Detection Methods. There is a variety of prediction– and reconstruction-based models for unsupervised anomaly detection in univariate time series (Wong et al., 2022; Ren et al., 2019; Munir et al., 2019). ARIMA (Box and Pierce, 1970) is a classic auto-regressive model to find the residuals between the real and predicted time series. Similarly, Hundman et al. (2018) utilizes an LSTM with a non-parametric threshold to locate the position of anomalies. On the other hand, Geiger et al. (2020) proposes a reconstruction-based model using GANs to find anomalies in time series through the reconstruction error and discriminator scores. Univariate models neglect the correlation between various channels and cannot be scaled efficiently, making them unsuitable for multivariate time series.

Multivariate Anomaly Detection Methods. Reconstruction-based methods are also utilized in multivariate time series anomaly detection (Xu et al., 2022; Tuli et al., 2022; Wu et al., 2023; Xu et al., 2024; Nam et al., 2024; Fang et al., 2024). Malhotra et al. (2016) develops an LSTM autoencoder and calculates an anomaly score using reconstruction error. Audibert et al. (2020) applies an autoencoder model with a two-phase adversarial training scheme. Li et al. (2021) uses a hierarchal variational autoencoder with explicit low-dimensional inter-metric and temporal embeddings. OmniAnomaly uses RNNs to learn the normal representation of the data, utilizing reconstruction probabilities and an automatic threshold (Su et al., 2019). Except for the last model, all the proposed solutions construct a score for anomalies without properly configuring the appropriate threshold, requiring ad-hoc tuning of threshold values. In contrast to Su et al. (2019), we provide a simpler prediction-based model to generate a multivariate time series and derive a data-driven threshold.

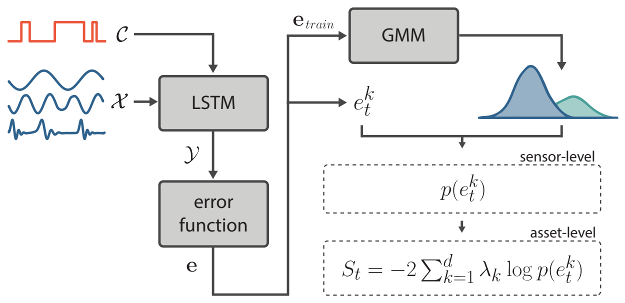


Figure 1: Overview of M^{2}AD workflow. The frameworks trains an LSTM to predict $\mathcal{Y}$, which models the expected pattern of $\mathcal{X}$. Then, M^{2}AD calculates the errors ($\mathbf{e}$) between the observed and expected signal for each sensor. Lastly, we use a Gaussian Mixture Model to find p-values and construct a global anomaly score for a particular asset from all its sensor information.

Anomaly Detection on Multiple Systems. Working with multiple data sources has become critical as the number of monitoring devices increase. Canizo et al. (2019) extracts features for each sensor independently using CNNs. The features are then concatenated and passed to an RNN to classify if they are anomalous or not. Extracting features of each sensor independently ignores the relationship between different sensors. Moreover, Park et al. (2018) proposes a reconstruction-based approach using variational autoencoders for multiple sensory data from different systems using LSTM layers for robotics. The algorithm computes the anomaly score from a global negative log-likelihood of an observation, making it is difficult to interpret which sensor triggered the anomaly.

3 M^{2}AD: Multi-Sensor Multi-System Anomaly Detection

Given an asset, our objective is to identify if any anomalous behaviour is observed. Figure 1 depicts an overview of our Multi-system Multivariate Time Series Anomaly Detection (M^{2}AD) framework.

3.1 Problem Statement

Consider anomaly detection for an asset, where a set of sensors observes multivariate time series data from potentially different systems. For simplicity of notation, we denote the combined observations as a single multivariate time series $\mathcal{X} = \{x_1, \ldots, x_T\}$, where $x_t \in \mathbb{R}^d$ and d represents the total number of sensors. Optionally, we have covariates $\mathcal{C} = \{c_1, \ldots, c_T\}$ which is a categorical time series representing the status of the asset. Our goal is to identify anomalous regions by finding $\mathcal{A} = \{t : z_t = 1\}$ in a given asset where $z_t \in \{0, 1\}$. Note that we do not have any prior information on the existence of anomalies, in fact $\mathcal{A}$ is often an empty set.

M^{2}AD, is composed from two phases: training and inference. We denote $\mathcal{D}_{train} = \{(x_t, c_t)_{i=1}^n\}$ with $n < T$ as the training dataset, where we assume $\mathcal{D}_{train}$ is free from anomalies and is operating normally. The remainder of the dataset will be used for testing $\mathcal{D}_{test} = \{(x_t, c_t)_{i=n}^T\}$.

3.2 Time Series Prediction

The first step of the model is to model what the behavior of the time series should look like under normal conditions. To train the model, we use the training dataset $\mathcal{D}_{train}$. We denote $\mathcal{Y}$ as the predicted time series, learned from the sensor– and status-type time series. We then aim to find anomalies in our inference data $\mathcal{D}_{test}$ where it may or may not have anomalies. Multiple models can be used to estimate $\mathcal{Y}$ from $\mathcal{X}$ and covariates $\mathcal{C}$ (Taylor and Letham, 2018; Das et al., 2024; Rasul et al., 2023; Ansari et al., 2024). We use a Long Short-Term Memory (LSTM) model to capture the temporal pattern of the signal (Hochreiter and Schmidhuber, 1997). Moreover, having a multivariate prediction helps us capture the dependencies across multiple variables:

$$y_{t+1} = f(x_{w_t}, c_{w_t}),$$

where w_t refers to the historical observations we want to include at time t and window size w, namely for input x_{w_t} we can express it as $x_{w_t} := x_{t \ldots t-w}$. We use the squared L_2 norm to compute the loss with respect to the sensor-type time series $\|x_t - y_t\|_2^2$.

3.3 Discrepancy Calculation

When calculating the discrepancy $\mathbf{e} = \{e_1, e_2, \ldots, e_T\}$, we want a function that highlights the possible occurrence of an anomaly, this is highly dependent on the behavior of the time series and we deem as being sensor-specific. Here, we propose two methods for finding this gap. Hundman et al. (2018) proposes a simple point-wise error to capture the differences $e_t = |x_t - y_t|$. Note that by taking the absolute value, we are not interested in the direction of the error but only the magnitude.

In many applications, whether the expectation is higher/lower than what is actually observed matters. For example, in temperature data, the model should be sensitive to whether there is a substantial increase. However it should not consider drops in temperature as typically that is not problematic. Moreover, point-wise error is sensitive to singular extreme values, which makes them unsuitable to very noisy systems. To factor in these nuances, we use an area difference error

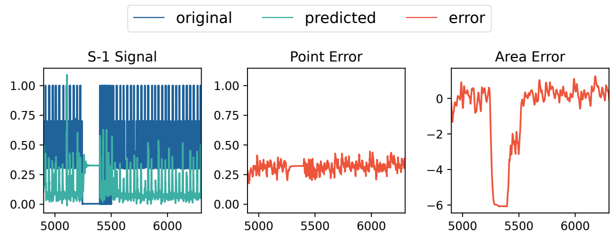


Figure 2: Example of S-1 signal from SMAP dataset. (left) depiction of the original time series and predicted one. (center) error signal using point-wise difference. (right) error signal using area difference.

function:

$$e_t = \frac{1}{2l}\int_{t-l}^{t+1} (x_t - y_t)\, dt,$$

where l is the number of time points to include before and after the time t. Figure 3 shows an example of computing the difference between the original and predicted signals using point and area errors respectively. The location of the anomaly in the area error signal is more apparent than the point error signal.

Choosing between point and area error depends on the characteristics of the data and anomaly. With point errors, one can effectively find out-of-range anomalies by measuring deviations in individual data points. On the other hand, area error is better at discovering contextual anomalies by capturing deviations in patterns, even when individual values are within expected ranges.

3.4 Anomaly Score

Once we obtain the discrepancy measure for each sensor, it remains to design a global anomaly score for the whole asset. We frame the detection task as a multiple testing problem, where the null hypothesis H_0 assumes that none of the sensors exhibit anomalous behavior at time t. Formally, for an asset with d sensors, the null hypothesis is:

$$H_0 : e_t^k \sim f_{\text{normal}}^k \text{ for all } k = 1, \ldots, d.$$

Determining an anomaly score consists of two steps: first calculating sensor-level scores and then aggregating scores to form a global anomaly score for the asset.

Sensor-Level Anomaly Score. For each sensor, we model the distribution of training error values using a Gaussian Mixture Model (GMM). Different sensors, especially those from different systems, often show distinct multimode error distributions due to various operational conditions. The GMM allows us to account for such a variation by fitting a multimode distribution. The GMM for sensor k is defined as follows:

$$f^k(x) = \sum_{j=1}^{m_k} p_j \cdot \mathcal{N}(x \mid \mu_j, \sigma_j^2),$$

where $\mathcal{N}(x \mid \mu_j, \sigma_j^2)$ is the normal density with mean μ_j and variance σ_j^2, and p_j is the weight of the j-th component. Here, m_k represents the number of components in the mixture for sensor k. See Appendix A.4 for more details on choosing m_k.

We estimate the parameters μ_j and σ_j from the training errors, which represents normal operational behavior. To determine whether an observed error score e_t^k at time t is anomalous, we calculate a two-sided p-value using the cumulative distribution function (CDF) of the GMM. The CDF, based on the estimated parameters, is given by:

$$F^k(x) = \sum_{j=1}^{m_k} p_j \cdot \Phi\left(\frac{x - \hat{\mu}_j}{\hat{\sigma}_j}\right),$$

where Φ is the standard normal CDF. The two-sided p-value is computed as $p(e_t^k) = 2\min\{F^k(e_t^k), 1 - F^k(e_t^k)\}$. A small p-value indicates an anomalous pattern is observed on sensor k. Note that with unidirectional errors, it is sufficient to conduct an upper– or lower-tailed test.

Asset-Level Anomaly Score. Once the sensor-level p-values have been obtained, we aggregate them to form a global anomaly score for the asset. One widely used aggregation method is the Fisher's method (Fisher, 1928), which calculates the global anomaly score S_t for an asset with d sensors at time t is calculated as:

$$S_t = -2\sum_{k=1}^{d} \lambda_k \log p(e_t^k),$$

where $p(e_t^k)$ is the p-value from sensor k at time t, and λ_k's are sensor-specific or uniform weights. If all the p-values are independent, with uniform weights, S_t follows a $\chi^2(2d)$ distribution under the null hypothesis. However, the p-values from different sensors may be heavily dependent on each other as this is a multi-sensor system monitoring the same asset. Therefore, we adjust classical Fisher's method by fitting a Gamma distribution $\Gamma(\alpha, \theta)$ using p-values obtained from the training phase. Specifically, the shape and scale parameters $\hat{\alpha}$ and $\hat{\theta}$ are estimated from the training data by matching the moments of the aggregated log-transformed p-values:

$$\hat{\alpha} = \frac{(E_n[S_t])^2}{\text{Var}_n[S_t]}, \quad \hat{\theta} = \frac{\text{Var}_n[S_t]}{E_n[S_t]}.$$

To classify a time point t as anomalous, we compare the global anomaly score S_t to a threshold derived from the fitted Gamma distribution. A time point is classified as anomalous if the global anomaly score exceeds a pre-defined threshold:

$$z_t := \mathbb{1}[S_t > \gamma],$$

where γ is the threshold value derived from the Gamma distribution $\Gamma(\hat{\alpha}, \hat{\theta})$ for a desired significance. By following this two-step process, we can robustly combine sensor-level anomaly scores and detect anomalies at the asset level.

3.5 Interpretability of Anomaly Score

The modular design of M^{2}AD helps us verify which sensor from which system flagged an anomaly. Having this level of interpretability is key to adoption at Amazon warehouses. This is because on-site technicians need to investigate and examine part of the asset that raised an alarm. To achieve this, we simply find the most contributing sensor to the anomaly score at time t:

$$\arg\max_k \frac{\lambda_k \log p(e_t^k)}{S_t}.$$

Top contribution can belong to different systems. In practice, we show the top 5 contributing sensors alongside their score and their respective systems.

4 Properties of M^{2}AD Scoring

In this section, we demonstrate the statistical properties of the anomaly score obtained from M^{2}AD.

4.1 Improvement in Error Quantification

One of the key feature of M^{2}AD in the sensor-level scores is the use of GMM model instead of the widely used univariate Gaussian distribution. If for one system the underlying true distribution of the anomaly score is well approximated by a mixed Gaussian, the univariate Gaussian approach will be heavily biased. Here we characterize the bias introduced due to such a misspecification. Consider the case where the true model distribution, $\mathbb{P}_*$, is a mixture of a two-component Gaussian location mixture given by

$$\mathbb{P}_* = \frac{1}{2}\mathcal{N}\left(\mu^*(1+\rho), \sigma^2\right) + \frac{1}{2}\mathcal{N}\left(\mu^*(1-\rho), \sigma^2\right), \quad (1)$$

where the positive scalar ρ characterizes the separation between two cluster means $\mu^*(1+\rho)$ and $\mu^*(1-\rho)$. The standard univariate Gaussian approach fits the data with the model

$$\mathbb{P}_\mu = \mathcal{N}\left(\mu, \sigma^2\right). \quad (2)$$

In an ideal scenario with an infinite number of samples, the best fit with the univariate approach would be $\bar{\mu} \in \arg\min_\mu \mathrm{KL}\left(\mathbb{P}_*, \mathbb{P}_\mu\right)$ where KL denotes the KL divergence measure. In the following position, we adapt the existing technique on mis-specified GMM (Dwivedi et al., 2018) to quantify the bias of such a best fit $\bar{\mu}$ and the true parameter μ^*.

Proposition 1. *Given the true model (1) and any $\rho > 0$ and $\sigma > 0$, we have*

$$|\mu^* - \bar{\mu}| \leq \rho\,|\mu^*| + c\left(\frac{\rho\,|\mu^*|}{\sigma}\right)^{1/4}, \quad (3)$$

where c is a universal positive constant.

The proof of the proposition utilizes several bounds between KL divergence, Hellinger distance, and Wasserstein distance and is deferred to the Appendix. Bound (3) quantifies the bias of the parameter of the best fitted model from class (2) (Dwivedi et al., 2018). Proposition 1 shows that under such a misspecification scenario, there could be a non-negligible bias of the traditional univariate approach in estimating the anomaly score distribution, and such a bias grows with the separation degree ρ. On the other hand, it's known that a properly specified GMM model could consistently approach the true parameter μ^*, resulting in an improvement in error quantification (Dwivedi et al., 2018).

4.2 Improvement in Score Aggregation

In M^{2}AD, we fit a Gamma distribution instead of the traditional chi-square distribution to the aggregated test statistic S_t to account for dependencies among the p-values. We now explore how this relaxation from the chi-square to the Gamma distribution leads to better detection accuracy. To formalize this, we examine the asymptotic behavior of the p-values as the test statistic grows large:

Proposition 2. *Suppose the true distribution of the test statistic S_t follows a Gamma distribution $\Gamma(\alpha, \theta)$. When using a chi-square distribution $\chi^2(2\alpha)$ to approximate the p-value, the ratio of the correct p-value to the incorrect p-value satisfies:*

$$\lim_{S_t \to \infty} \frac{p_{correct}}{p_{wrong}} = \begin{cases} 1 & \text{if } \theta = 2, \\ \infty & \text{if } \theta < 2, \\ 0 & \text{if } \theta > 2. \end{cases}$$

This theorem confirms that using the chi-square distribution for dependent test statistics can lead to an incorrect p-value calibration. Specifically, when $\theta < 2$, the p-value under the chi-square assumption becomes disproportionately smaller than the correct Gamma-derived p-value, resulting in overly aggressive detections and a higher risk of false positives. On the other hand, when $\theta > 2$, the correct p-value can be much

Table 1: Public Datasets Summary. The table presents the size of the dataset and its dimension, the number of known anomalies, and the average signal length. These datasets are publicly accessible.

Dataset	# Assets	Dimension	# Anomalies	Avg. Signal
MSL	27	55	36	4890.59
SMAP	54	25	67	10618.86
SMD	28	38	327	50600.89

smaller, making the test overly conservative and increasing the risk of false negatives, potentially missing critical anomalous behaviors. Adopting the Gamma distribution for score aggregation in dependent scenarios mitigates these issues, leading to more accurate score aggregation.

5 Experimental Results

In this section, we assess the performance of M^{2}AD and conduct studies to verify our design choices.

5.1 Evaluation Setup

Datasets. We use three multivariate public datasets with known ground truth anomalies summarized in Table 1. All datasets are pre-split into training and testing sets. **Mars Science Laboratory (MSL)** rover dataset is a real public dataset provided by NASA (Hundman et al., 2018). This dataset contains 27 assets each has 55 variables; one variable is continuous and the remaining are binary. MSL contains 36 anomalies that were manually labeled by a team of experts. **Soil Moisture Active Passive (SMAP)** satellite is another dataset provided by NASA. This dataset has 54 assets with 25 variables each with a similar structure to MSL. This dataset has 69 anomalies. Both MSL and SMAP are publicly available [1]. **Server Machine Dataset (SMD)** is a multivariate public dataset [2] featuring 28 assets collected from a large internet company with 327 anomalies (Su et al., 2019). Each asset in this dataset has 38 measurements composing from a variety of continuous and categorical variables.

Models. For model comparisons, we evaluate eight models that are competitive to M^{2}AD. **LSTM-DT** is a prediction-based model with non-parametric dynamic thresholding developed originally on the NASA datasets (Hundman et al., 2018). **LSTM-AE** is an autoencoder architecture with LSTM model that works on reconstructing the signal. When the model fails to reconstruct the signal properly, this indicates the possibility of an anomaly (Malhotra et al., 2016). **LSTM-VAE** works on reconstructing time series and utilizes the reconstruction loss to determine if an observation is anomalous. **USAD** is an autoencoder model with two decoders, one of which is adversarially trained. The anomaly score is then constructed from loss values of the weighted sum between the first and second decoder (Audibert et al., 2020). **OmniAnomaly (OA)** is another reconstruction-based model using RNNs with stochastic variable connections to learn the normal pattern of the time series. The reconstruction probability is then used as an anomaly score (Su et al., 2019). **AnomalyTransformer (AT)** is a reconstruction-based transformer model that uses attention mechanism to distinguish normal patterns from anomalies (Xu et al., 2022). **TimesNet** is a 2D variational representation of temporal signals (Wu et al., 2023). **FITS** is a compact model operating in the frequency domain to learn the amplitude and shift of the time series (Xu et al., 2024). Both TimesNet and FITS leverage reconstruction error to find anomalies.

Evaluation Criteria. The nuances in time series data makes point-based evaluations unsuitable (Tatbul et al., 2018). To compute precision and recall metrics, we refer to the commonly used evaluation criteria in Hundman et al. (2018), where we assess the performance of the model as to whether or not it was able to locate the anomaly. From the perspective of associates who monitor the data, the desired outcome is to (1) locate true alarms that requires the attention of the associate; and (2) achieve that with very minimal false alarms (false positives). Therefore, we record **true positive (tp)** if the detected anomaly overlaps with any part of a known ground truth; **false positive (fp)** if the detected anomaly does not have any counter-part in the ground truth dataset; and **false negative (fn)** for all ground truth alarms that were not detected. Details on the calculation of scores are provided in Appendix C.2.

5.2 Comparing M^{2}AD to Existing Methods.

Table 2 showcases the performance of M^{2}AD compared to models described earlier in Section 5.1. The table depicts the $F1$ and $F_{0.5}$ scores where we favor precision in the latter. On average, M^{2}AD achieves the highest scores and outperforms other methods, surpassing the next-best model by 21% and 25% in F_1 and $F_{0.5}$ scores, respectively. There is a clear performance gain on MSL and SMD datasets. However, we see that LSTM models are competitive in the SMAP dataset.

Competing models suffer in performance on SMD datasets compared to MSL and SMAP, since the number of observations and variable dimension is higher. Recall that MSL and SMAP have only one sensor variable and the remaining are covariates. Moreover, the latter models struggle to score above average when it

[1] `https://github.com/khundman/telemanom`

[2] `https://github.com/NetManAIOps/OmniAnomaly`

Table 2: Benchmark Results. The table summarizes $F1$ and $F_{0.5}$ scores across MSL, SMAP, and SMD datasets. Each model is executed for 5 iterations and we record the mean score and standard deviation.

	MSL		SMAP		SMD		Average	
Model	$F1$	$F_{0.5}$	$F1$	$F_{0.5}$	$F1$	$F_{0.5}$	$F1$	$F_{0.5}$
M^{2}AD	**0.749 ± 0.014**	**0.752 ± 0.018**	**0.716 ± 0.010**	0.656 ± 0.012	**0.913 ± 0.004**	**0.906 ± 0.005**	**0.789 ± 0.009**	**0.769 ± 0.012**
USAD	0.700 ± 0.037	0.631 ± 0.044	0.650 ± 0.050	0.565 ± 0.057	0.608 ± 0.002	0.644 ± 0.002	0.653 ± 0.030	0.613 ± 0.035
OA	0.562 ± 0.035	0.457 ± 0.034	0.577 ± 0.021	0.564 ± 0.031	0.576 ± 0.003	0.471 ± 0.003	0.572 ± 0.020	0.497 ± 0.022
LSTM-DT	0.483 ± 0.006	0.408 ± 0.005	0.713 ± 0.008	**0.677 ± 0.009**	0.433 ± 0.008	0.394 ± 0.007	0.543 ± 0.008	0.493 ± 0.007
LSTM-AE	0.493 ± 0.023	0.490 ± 0.027	0.672 ± 0.011	0.651 ± 0.016	0.402 ± 0.007	0.395 ± 0.005	0.522 ± 0.014	0.512 ± 0.016
LSTM-VAE	0.492 ± 0.009	0.476 ± 0.010	0.649 ± 0.005	0.615 ± 0.005	0.412 ± 0.003	0.411 ± 0.002	0.518 ± 0.006	0.501 ± 0.006
AT	0.391 ± 0.018	0.455 ± 0.027	0.359 ± 0.014	0.387 ± 0.034	0.533 ± 0.005	0.470 ± 0.009	0.428 ± 0.012	0.437 ± 0.023
TimesNet	0.276 ± 0.017	0.208 ± 0.014	0.169 ± 0.008	0.117 ± 0.006	0.409 ± 0.007	0.310 ± 0.006	0.285 ± 0.011	0.212 ± 0.009
FITS	0.386 ± 0.005	0.338 ± 0.005	0.136 ± 0.003	0.097 ± 0.002	0.316 ± 0.005	0.286 ± 0.006	0.279 ± 0.004	0.240 ± 0.004

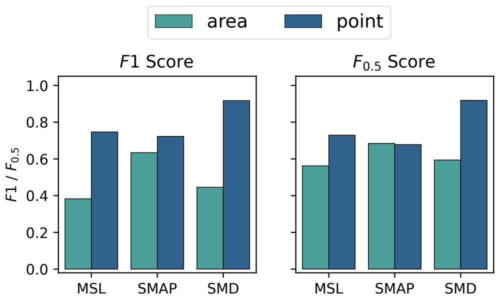


Figure 3: Performance under different error functions on MSL, SMAP, and SMD datasets.

comes to smaller datasets such as MSL and SMAP. This suggests that our model is both data-efficient and scalable in contrast to the rest.

5.3 Sensitivity to error functions.

The objective of the error function is to unveil the location of the anomalies based on the behavior of the signal and the anomaly type the model is searching for. The selection of this function highly impacts the proceeding steps and the ability to detect anomalies.

Figure 3 shows the performance scores under the two error functions suggested in Section 3.3. For MSL and SMD, the deviation is substantial, specifically in F1 score where we see an improvement by a factor of $\times 2$ when using point errors. As for SMAP, we notice that with area errors we obtain a precision of 0.72 compared to 0.65 when using point errors. This emphasizes that depending of the objective criteria, we can favor one discrepancy function over another.

5.4 Ablation Study

We perform an ablation study to test the specifications of our anomaly score model under three settings: (1) we apply a static threshold ($\mu \pm 4\sigma$) on the error values; (2) we compare GMMs to a normal distribution; (3) we compare Gamma to chi-square calibration. Table 3 presents the results, with M^{2}AD configuration shown in the bottom row. First, we see that using Gamma calibration, which properly accounts for the dependency between p-values during score aggregation, leads to substantial gains in both F_1 and $F_{0.5}$ scores compared to chi-square calibration. Furthermore, when adopting the GMM model for anomaly score aggregation, we observe the best performance across all baselines.

Table 3: Ablation Study on SMD dataset.

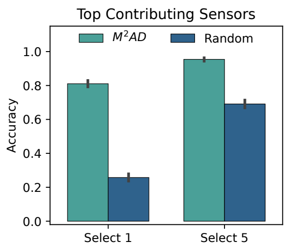
Variation	F_1	$F_{0.5}$
static	0.745 ± 0.000	0.724 ± 0.000
GMM with χ^2 calibration	0.258 ± 0.032	0.426 ± 0.043
normal with χ^2 calibration	0.317 ± 0.050	0.490 ± 0.059
normal with Γ calibration	0.905 ± 0.000	0.890 ± 0.001
GMM with Γ calibration	0.913 ± 0.004	0.906 ± 0.005

5.5 Interpretability

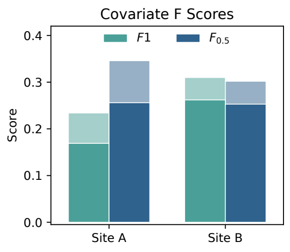
One of most important aspects in our design is that we can isolate the cause of the anomaly by finding the most contributing sensor to the anomaly score. Fortunately, the SMD dataset provides an interpretability document that summarizes for each ground truth anomaly, which sensors caused it. Figure 4a shows how the top contributing sensors detected by M^{2}AD agree with the sensors provided as ground truth in SMD for the correctly identified anomalies (297 out of 327). When we select the top contributing sensors and compare it to randomly selecting one and five sensors, respectively, we find that M^{2}AD is correct 81% of time. Moreover, if we observed the top-5 contributing sensors to the anomaly score, we reach 95%.

6 Case Study: Amazon Fulfillment Centers

The motivation of M^{2}AD began to enhance Amazon's predictive maintenance efforts. At Amazon's fulfillment centers, there are many equipment that are monitored on each individual sensor separately. The systems alert the respective technician when the expected threshold repeatedly exceeds the standard threshold. This pro-



(a) Interpretability.



(b) Scores with covariates.

Figure 4: (a) Accuracy of top contributing sensors to the anomaly score compared to a random selection in SMD. (b) Change in model performance without covariates (dark) and with covariates (light).

Table 4: Data Summary. Site A has a larger number of equipment data that we collected compared to Site B. Moreover, the number of sensors attached to the equipment are higher in Site A.

		dimension		
	# equipment	median	min	max
Site A	82	51	11	121
Site B	48	21	21	191

cess results in a high number of false alarms especially when the sensory data is noisy. In this section, we present a real-world evaluation of our system on equipment data collected from Amazon fulfillment centers.

Dataset. Our study was performed on Amazon's proprietary data featuring equipment sensors. We collected 10 months worth of data for two sites Site A and Site B with varying measurements. These measurements are gathered from two independent systems: amperage which summarizes amperage from VFD motors, and Monitron which collects vibration and temperature measurements. In addition, we collect status variables to use as covariates. Covariates capture non-physical contextual information that helps us model system behavior. For example, we include throughput, which is the number of processed packages, after discretizing it into low, medium, and high bins. Moreover, we use the percentage of time where the machine is available on planned production time as equipment availability. Each equipment has a varying number of sensors that monitors its condition. In other words, the dimension of the data varies by asset. Table 4 shows a summary of the data used.

Like most industrial datasets, this dataset does not have exact labels on whether or not an anomaly occurred. To measure the performance of our model, we use filed work orders as a proxy to know if the asset has

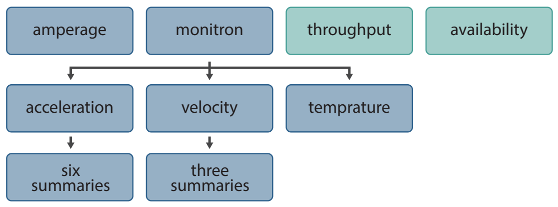


Figure 5: Hierarchical structure of data sources. We collect data from amperage and Monitron which contain sensory information (blue). We also leverage the throughput and availability as covariates (green).

been investigated. We look at work orders from two perspective: *breakdown* work orders which describes if the equipment broke down and needed attention and action to be taken by a technician; and the *total* work orders that had caused downtime to the equipment (including breakdowns) and needed technician intervention for unplanned maintenance.

Baseline. The current production model at Amazon is based on a combination of statistical and ISO thresholds for vibration sensors only. Out of all the raised alarms in Sites A and B, only 19.2% and 23.1% of them are correct, respectively. More details on the baseline performance are provided in Appendix C.4.

Experiment Setup. We utilize hourly data with median aggregations and select the hyperparameters of the model based on the nature of the problem and dataset. For Monitron sensors, we set the number of components for the GMM as $m_k = 1$. For amperage, we set the number of components to $m_k = 2$. Having two components allowed us to fit both "operation" mode and "shutdown" mode better (He et al., 2024).

Based on the hierarchy organization of the data shown in Figure 5, and that one data source (Monitron) outnumbers the number of sensors in the amperage system, we construct an equally divided weighing vector: $\lambda_k = 1/(\# \text{ systems} \cdot \# \text{ sensors} \cdot \# \text{ summaries})$. This ensures that under-represented sensors, e.g. amperage, does not lose its contribution to the anomaly score.

Predictive Performance. Table 5 depicts the performance of M^{2}AD on the two sites. On the left-hand side, we view the performance under a predictive setting. This means that we only count a detection as true positive if it happens between a day to 7 days *before* a work order is initiated. The assumption here is that a precursor to the actual breakdown can be detected within the monitored time series. Compared to the baseline model, M^{2}AD improves precision by 16% in Site A and 4% in Site B.

We also demonstrate the performance of the framework using the detection strategy, this follows the regular

Table 5: M^2AD evaluation on Amazon data. The table depicts precision, recall, and F scores under two evaluation strategies. Predictive is a strict strategy where the detection must be at least a day before the work order. Detection is lenient strategy where any detection within a 7-day window counts towards a correct identification.

		Predictive				Detection			
site	work order type	Precision	Recall	$F1$	$F_{0.5}$	Precision	Recall	$F1$	$F_{0.5}$
Site A	total	0.353	0.174	0.233	0.293	0.605	0.362	0.453	0.534
	breakdowns	0.274	0.178	0.216	0.247	0.482	0.364	0.415	0.453
Site B	total	0.269	0.448	0.336	0.293	0.369	0.632	0.466	0.403
	breakdowns	0.143	0.528	0.225	0.167	0.192	0.717	0.303	0.225

evaluation criteria introduced in Section 5. Naturally, we observe a higher performance in the detection evaluation strategy compared to the predictive one.

Importance of Covariates. Figure 4b showcases the model performance with and without including "throughput" and "availability" as covariates to the model. There is a clear improvement on all fronts after adding covariates to the model. Specifically, the number of false positives reduced by 16.5% in Site A and 12% in Site B on average for both work order types. Note that the scores shown in Figure 4b differ from Table 5 since the subset of assets and work orders has changed. This stems from the data collection process where not all asset have an associated throughput and availability measurements. Only 46% and 50% of assets have covariates in Site A and Site B respectively.

Discussion. In Site A, we observe high precision and low recall scores while Site B exhibits counter scores. This traces back to number of *breakdown* work orders created in Site A, which are 6× the order of Site B, making it more likely to find anomalies. On average, the model predicts one anomaly every 2 months per asset. Having a low number of flagged anomalies is crucial to gain the trust of on-site technicians.

The flexibility of M^2AD has been key when certain sensor data becomes unavailable. Since the framework design is at sensor-level, we simply omit the contribution of that sensor. However, limitations of our model arise when the training distribution completely shifts from the target distribution, necessitating model retraining.

7 Conclusion

In this paper, we propose M^2AD, a modular framework for detecting anomalies in multi-sensor multi-system time series data. We learn the temporal patterns and cross-sensor dependencies using a forecasting model which then feeds to a discrepancy function to capture the error vectors. The model uses a data-driven approach using GMMs and Fisher's method to construct a global anomaly score for each observation in the time series dataset.

We show the effectiveness of our model by comparing it to 8 competitive methods on three multivariate public datasets. Moreover, we highlight our case study in Amazon fulfillment centers in utilizing the proposed framework, where it improved on the current baseline precision by 16%. M^2AD's flexibility allows it to function even when some sensors are unavailable, enhancing its practicality for deployment.

References

Abdulmalek, S., Nasir, A., Jabbar, W. A., Almuhaya, M. A. M., Bairagi, A. K., Khan, M. A.-M., and Kee, S.-H. (2022). IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review. *Healthcare*, 10(10).

Alnegheimish, S., Liu, D., Sala, C., Berti-Equille, L., and Veeramachaneni, K. (2022). Sintel: A Machine Learning Framework to Extract Insights from Signals. In *Proceedings of the 2022 International Conference on Management of Data*, pages 1855—-1865.

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., Zschiegner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Wilson, A. G., Bohlke-Schneider, M., and Wang, Y. (2024). Chronos: Learning the Language of Time Series. *Transactions on Machine Learning Research*.

Audibert, J., Michiardi, P., Guyard, F., Marti, S., and Zuluaga, M. A. (2020). USAD: UnSupervised Anomaly Detection on Multivariate Time Series. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3395–3404.

Box, G. E. and Pierce, D. A. (1970). Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *Journal of the American statistical Association*, 65(332):1509–1526.

Braei, M. and Wagner, S. (2020). Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art.

Canizo, M., Triguero, I., Conde, A., and Onieva, E. (2019). Multi-head CNN–RNN for Multi-Time Series Anomaly Detection: An Industrial Case Study. *Neurocomputing*, 363:246–260.

Cao, Q., Giustozzi, F., Zanni-Merk, C., de Bertrand de Beuvron, F., and Reich, C. (2019). Smart Condition Monitoring for Industry 4.0 Manufacturing Processes: An Ontology-Based Approach. *Cybernetics and Systems*, 50(2):82–96.

Choi, K., Yi, J., Park, C., and Yoon, S. (2021). Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines. *IEEE Access*, 9:120043–120065.

Das, A., Kong, W., Sen, R., and Zhou, Y. (2024). A decoder-only foundation model for time-series forecasting. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Dwivedi, R., Khamaru, K., Wainwright, M. J., Jordan, M. I., et al. (2018). Theoretical guarantees for em under misspecified gaussian mixture models. *Advances in Neural Information Processing Systems*, 31.

Fang, Y., Xie, J., Zhao, Y., Chen, L., Gao, Y., and Zheng, K. (2024). Temporal-Frequency Masked Autoencoders for Time Series Anomaly Detection. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1228–1241.

Fisher, R. A. (1928). *Statistical methods for research workers*. Number 5. Oliver and Boyd.

Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., and Veeramachaneni, K. (2020). TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 33–43.

He, Z., Reimherr, M., Alnegheimish, S., and Chandrayan, A. (2024). Weakly-supervised multi-sensor anomaly detection with time-series foundation models. In *Proceedings of the 2024 Annual Conference on Neural Information Processing Systems (NeurIPS) Time Series in the Age of Large Models Workshop*.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., and Soderstrom, T. (2018). Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 387–395.

Hunter, J. S. (1986). The exponentially weighted moving average. *Journal of quality technology*, 18(4):203–210.

Kim, H., Kim, S., Min, S., and Lee, B. (2024). Contrastive Time-Series Anomaly Detection. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5053–5065.

Li, Z., Zhao, Y., Han, J., Su, Y., Jiao, R., Wen, X., and Pei, D. (2021). Multivariate Time Series Anomaly Detection and Interpretation using Hierarchical Inter-Metric and Temporal Embedding. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 3220–3230.

Liu, Z. and Zhang, L. (2020). A review of failure modes, condition monitoring and fault diagnosis methods for large-scale wind turbine bearings. *Measurement*, 149:107002.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff, G. (2016). Lstm-based encoder-decoder for multi-sensor anomaly detection. In *Proceedings of the 2016 International Conference on Machine Learning (ICML) Anomaly Detection Workshop*.

Munir, M., Siddiqui, S. A., Dengel, A., and Ahmed, S. (2019). DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. *IEEE Access*, 7:1991–2005.

Nam, Y., Yoon, S., Shin, Y., Bae, M., Song, H., Lee, J.-G., and Lee, B. S. (2024). Breaking the Time-Frequency Granularity Discrepancy in Time-Series Anomaly Detection. In *Proceedings of the ACM Web Conference 2024*, pages 4204–4215.

Park, D., Hoshi, Y., and Kemp, C. C. (2018). A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551.

Rasul, K., Ashok, A., Williams, A. R., Ghonia, H., Bhagwatkar, R., Khorasani, A., Bayazi, M. J. D., Adamopoulos, G., Riachi, R., Hassen, N., et al. (2023). Lag-Llama: Towards Foundation Models for Probabilistic Time Series Forecasting. *arXiv preprint arXiv:2310.08278*.

Ren, H., Xu, B., Wang, Y., Yi, C., Huang, C., Kou, X., Xing, T., Yang, M., Tong, J., and Zhang, Q. (2019). Time-Series Anomaly Detection Service at Microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 3009–3017.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., and Pei, D. (2019). 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*, pages 2828–2837.

Tahan, M., Tsoutsanis, E., Muhammad, M., and Abdul Karim, Z. (2017). Performance-based health monitoring, diagnostics and prognostics for condition-based maintenance of gas turbines: A review. *Applied Energy*, 198:122–144.

Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., and Gottschlich, J. (2018). Precision and Recall for Time Series. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31.

Taylor, S. J. and Letham, B. (2018). Forecasting at Scale. *The American Statistician*, 72(1):37–45.

Tuli, S., Casale, G., and Jennings, N. R. (2022). TranAD: deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6):1201–1214.

Villani, C. et al. (2009). *Optimal transport: old and new*, volume 338. Springer.

Wong, L., Liu, D., Berti-Equille, L., Alnegheimish, S., and Veeramachaneni, K. (2022). AER: Auto-encoder with regression for time series anomaly detection. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 1152–1161.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. (2023). TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*.

Xu, J., Wu, H., Wang, J., and Long, M. (2022). Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In *International Conference on Learning Representations*.

Xu, Z., Zeng, A., and Xu, Q. (2024). FITS: Modeling Time Series with 10k Parameters. In *International Conference on Learning Representations*.

Zamanzadeh Darban, Z., Webb, G. I., Pan, S., Aggarwal, C., and Salehi, M. (2024). Deep Learning for Time Series Anomaly Detection: A Survey. *ACM Computing Survey*.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Section 3.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] Will be submitted with supplementary material.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] Section 3 and supplementary material.
   (b) Complete proofs of all theoretical results. [Yes] supplementary material.
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] supplementary material.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] Section 5 and supplementary material.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] supplementary material.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Not Applicable]
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   (d) Information about consent from data providers/curators. [Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials for M^{2}AD: Multi-Sensor Multi-System Anomaly Detection using Global Scoring and Calibrated Thresholding

In this supplementary material, we provide more details on the framework as well as additional experiments. Additionally, we illustrate the limitations of our methodology.

A Additional M^{2}AD Details

A.1 Problem Statement

Given $\mathcal{S}$ systems, where we observe a multivariate time series $\mathcal{X}_s = \{x_{s_1}, \ldots, x_{s_T}\}$ for all $s \in \mathcal{S}$ where $x_{s_t} \in \mathbb{R}^{d_s}$ and system s has d_s sensors, and T is the length of the time series, we want to find the anomalies present in $\mathcal{X}_s$ for all $s \in \mathcal{S}$.

For simplicity of notation, we denote the combined observations as a single multivariate time series $\mathcal{X} = \{x_1, \ldots, x_T\}$, where $x_t \in \mathbb{R}^d$ and d represents the total number of sensors. Optionally, we have covariates $\mathcal{C} = \{c_1, \ldots, c_T\}$ which is a categorical time series representing the status of the asset.

Below is the expected input of M^{2}AD

$$\mathcal{D} = \left[\begin{array}{cccc|ccc} x_{11} & x_{12} & \ldots & x_{1d} & c_{11} & \ldots & c_{1m} \\ x_{21} & x_{22} & \ldots & x_{2d} & c_{21} & \ldots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_{T1} & x_{T2} & \ldots & x_{Td} & c_{T1} & \ldots & c_{Tm} \end{array}\right]_{T \times (d+m)}$$

where T is the length of the time series, d is the number of sensors we have in total, and m is the number of status time series we include as covariates in the model.

Our goal is to identify anomalous regions by finding $\mathcal{A} = \{t : z_t = 1\}$ in a given asset where $z_t \in \{0, 1\}$. Note that we do not have any prior information on the existence of anomalies, in fact $\mathcal{A}$ is often an empty set.

The model works in two phases: training and inference. We split the data into $\mathcal{D}_{train} = \{(x_t, c_t)_{i=1}^{n}\}$ and $\mathcal{D}_{test} = \{(x_t, c_t)_{i=n}^{T}\}$ where $n < T$ and we will use $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ train our model and test it respectively. The main assumption of the model is that both $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$ are operating under the same condition. Moreover, $\mathcal{D}_{train}$ does not contain anomalies such that we learn a proper model of normal data.

A.2 Time Series Prediction

In order to feed the time series into an LSTM model, we use a sliding window approach to slice the data. We select a window size w, we find a corresponding slice which gives us x_{w_t} and c_{w_t} starting at time t.

$$x_{w_t} = \{x_t, x_{t-1}, \ldots, x_{t-w}\}$$
$$c_{w_t} = \{c_t, c_{t-1}, \ldots, c_{t-w}\}$$

where we now have $y_{t+1} = f(x_{w_t}, c_{w_t})$. We find y_t for all $t \in [w+1, n]$ in the training phase.

A.3 Discrepancy Calculation

With absolute point-wise difference error, we observe many spikes. In an effort to reduce extreme values, we apply an exponentially-weighted average (EWMA) to produce smoothed errors (Hunter, 1986; Hundman et al., 2018).

A.4 Choice of Number of Components in GMM

In practice, the number of components m_k is determined by the operational characteristics of each sensor system. For example, amperage sensors, which frequently undergo shutdowns and restarts (He et al., 2024), often require a Gaussian mixture model (GMM) with two components ($m_k = 2$), while more stable systems, such as Monitron, can typically be modeled with a single component ($m_k = 1$).

When such prior knowledge is unavailable, we determine m_k using the Bayesian Information Criterion (BIC):

$$\mathrm{BIC} = -2 \cdot \mathrm{Likelihood} + k \log(n),$$

where k is the number of parameters and n is the sample size. BIC balances model complexity and goodness of fit, providing a principled approach for selecting the optimal number of components.

B Proof of Propositions

B.1 Proof of Proposition 1

To prove the proposition, we follow the similar arguments as in Dwivedi et al. (2018). We first establish an upper bound on the Wasserstein distance between the true distribution and the best fit model from the misspecified distribution, then proceed to connect such a Wasserstein distance with the bias term $|\bar{\mu} - \mu^*|$.

We first introduce the notation for the measure of true distribution (1) and the misspecified distribution (2):

$$G_* = \frac{1}{2}\delta_{\mu^*(1+\rho)} + \frac{1}{2}\delta_{\mu^*(1-\rho)}, \text{ and } G(\mu) = \delta_\mu.$$

where δ_μ denotes the dirac measure at μ. Here G_* is the mixing measure of interest and $G(\mu)$ is a measure with a (misspecified) univariate Gaussian distribution. Since the true model does not belong to the univariate Gaussian distribution class, the "best fit" would be the projection parameter

$$\bar{\mu} \in \underset{\mu}{\operatorname{argmin}} \, \mathrm{KL}\left(P_{G_*}, P_{G(\mu)}\right).$$

where $\mathrm{KL}(\mathbb{P}, \mathbb{Q}) = \int p(x) \log \frac{p(x)}{q(x)} dx$ is the Kullback-Leibler divergence between two distribution $\mathbb{P}$ and $\mathbb{Q}$. Further denote $W_2^2(\mathbb{P}, \mathbb{Q})$ as the second-order Wasserstein distance between $\mathbb{P}$ and $\mathbb{Q}$ (Villani et al., 2009), applying the arguments from (26) to (29) in Dwivedi et al. (2018) yields

$$2\sigma\sqrt{C} W_2^4\left(G_*, G(\bar{\mu})\right) \leqslant W_2\left(G_*, G\left(\mu^*\right)\right) \tag{4}$$

where C is a universal constant and σ is the standard deviation defined in (1). Note that $G\left(\mu^*\right) \neq G_*$ as they belong to different distribution classes. Next, we establish an upper bound for the distance $W_2\left(G_x, G(\bar{\mu})\right)$ by deriving an upper bound for the distance $W_2\left(G_x, G\left(\mu^*\right)\right)$. According to variational formulation in (24d) of Dwivedi et al. (2018), we have that for the pair $(G^*, G(\mu^*))$, we can construct the coupling matrix T and the corresponding distance matrix D as

$$T = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}, \quad D = \begin{bmatrix} (\rho\mu^*)^2 \\ (-\rho\mu^*)^2 \end{bmatrix},$$

and further obtain

$$W_2^2\left(G_*, G\left(\mu^*\right)\right) \leqslant \sum_{i=1}^{2}\sum_{j=1}^{2} T_{ij} D_{ij} = \rho^2 {\mu^*}^2. \tag{5}$$

Combining (5) with (4) implies

$$W_2\left(G_*, G(\bar{\mu})\right) \leqslant c\left(\frac{\rho\left|\mu^*\right|}{\sigma}\right)^{1/4} \tag{6}$$

with $c = 1/(4C)^{1/8}$. We then apply the variation formulation (4d) in Dwivedi et al. (2018) again but this time w.r.t. the pair $(G_*, G(\bar{\mu}))$ with

$$T' = T = \begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix} \quad D' = \begin{bmatrix} [(1-\rho)\mu^* - \bar{\mu}]^2 \\ [(1+\rho)\mu^* - \bar{\mu}]^2 \end{bmatrix},$$

we then have

$$W_2\left(G_*, G(\bar{\mu})\right) \geqslant \min_{i,j} \sqrt{D'_{i,j}} = \min\left\{(1-\rho)\mu^* - \bar{\mu}, (1+\rho)\mu^* - \bar{\mu}\right\} \tag{7}$$

$$\geqslant |\mu^* - \bar{\mu}| - \rho\,|\mu^*|, \tag{8}$$

this together with (6) leads to the results in Proposition 1.

B.2 Proof of Proposition 2

Denote $F_S(s;\alpha,\theta)$ as the CDF of the Gamma distribution $\Gamma(\alpha,\theta)$ and $F_S(s;\alpha,2)$ as the CDF of the $\chi^2(2\alpha)$ distribution (as $\chi^2(2\alpha)$ is equivalent to $\Gamma(\alpha,2)$). Notice that here we neglect the subscript t for notation simplicity. We aim to analyze the asymptotic behavior of

$$\frac{p_{\text{correct}}}{p_{\text{wrong}}} = \frac{1 - F_S(s;\alpha,\theta)}{1 - F_S(s;\alpha,2)}.$$

as $s \to \infty$. Notice that the PDF of $F_S(S;\alpha,\theta)$ is $f_S(s;\alpha,\theta) := \frac{s^{\alpha-1}\exp(-s/\theta)}{\Gamma(\alpha)\theta^\alpha}$ with $\Gamma(\alpha)$ being the Gamma function. Therefore, by L'Hospital's Rule we have

$$\lim_{s\to\infty} \frac{p_{\text{correct}}}{p_{\text{wrong}}} = \lim_{s\to\infty} \frac{f_S(s;\alpha,\theta)}{f_S(s;\alpha,2)} = \frac{2^\alpha}{\theta^\alpha} \lim_{s\to\infty} \exp\left(-s\left(\frac{1}{\theta} - \frac{1}{2}\right)\right).$$

which leads to the result in Proposition 2.

C Experiments

In this section we provide more details on the experimental setup as well as some addition results.

C.1 Models

We compare our model to eight state-of-the-art machine learning models for unsupervised time series anomaly detection. These models were chosen based on their similarity in nature to our proposed model.

LSTM-DT is a prediction-based model that uses LSTM to forecast a time series, then applies dynamic thresholding on the absolute point-wise errors computed from the original signal and forecasted one (Hundman et al., 2018). LSTM-DT uses a window size of 250 based on recommendations by the original authors.

LSTM-AE is an autoencoder architecture with LSTM model that works on reconstructing the signal. When the model fails to reconstruct the signal properly, this indicates the possibility of an anomaly (Malhotra et al., 2016). We set the window size to 100 since it is a reconstruction based model.

LSTM-VAE Similar to LSTM-AE, LSTM-VAE works on reconstructing time series using a variational autoencoder. The reconstruction error is utilized to locate anomalies.

USAD is an autoencoder model with two decoders, one of which is adversarially trained. The anomaly score is then constructed from a weighted sum between the loss first and second decoder (Audibert et al., 2020). USAD does not include a thresholding technique, therefore, we experimented with multiple values and found that 98th percentile yielded the best results.

OmniAnomaly (OA) is another reconstruction based model using RNNs with stochastic variable connections to learn the normal pattern of the time series. The reconstruction probability is then used as an anomaly score (Su et al., 2019).

AnomalyTransformer (AT) is a transformer model that uses attention mechanism to compute the association discrepancy, which distinguishes normal patterns from anomalies. They use a minimax training strategy to amplify the dichotomy between normal and abnormal patterns (Xu et al., 2022).

TimesNet is a novel framework which transforms 1D temporal signals into a 2D representation consisting of intraperiod- and interperiod-variations. Their framework can be used for multiple tasks, specifically for anomaly detection, they employ reconstruction error (Wu et al., 2023).

FITS is a compact neural network model for time series analysis operating in the frequency domain. The model aims to learn the amplitude and shift of the time series by interpolating the frequency representation. They leverage reconstruction error for testing the anomaly detection task (Xu et al., 2024).

To run these models, we used the code provided directly by `https://github.com/sintel-dev/Orion` (Alnegheimish et al., 2022), `https://github.com/manigalati/usad` (Audibert et al., 2020), `https://github.com/NetManAIOps/OmniAnomaly` (Su et al., 2019), `https://github.com/thuml/Time-Series-Library` (Wu et al., 2023), and `https://github.com/VEWOXIC/FITS` (Xu et al., 2024).

C.2 Scoring

We compute how many true positives, false positives, and false negatives we obtain on a signal level. Because often times the model fails to detect any ground truth anomalies, the number of true positives is zero, making precision and recall scores undefined. To aggregate the results in a meaningful approach, we compute the performance scores with respect to the dataset D.

$$\text{precision}_D = \frac{\sum_{i=1}^{|D|} TP_i}{\sum_{i=1}^{|D|} TP_i + FP_i} \qquad \text{recall}_D = \frac{\sum_{i=1}^{|D|} TP_i}{\sum_{i=1}^{|D|} TP_i + FN_i}$$

where i denotes one signal from the dataset D.

Similarly, we compute the F scores in a similar fashion

$$F_{1_D} = 2 \cdot \frac{\text{precision}_D \cdot \text{recall}_D}{\text{precision}_D + \text{recall}_D} \qquad F_{0.5_D} = (1 + 0.5^2) \cdot \frac{\text{precision}_D \cdot \text{recall}_D}{0.5^2 \cdot \text{precision}_D + \text{recall}_D}$$

C.3 Number of Components in GMM

Using the criterion described in Appendix A.4, we conducted an exploratory experiment to calculate the BIC scores for models with varying numbers of components on benchmark datasets (see Table 6). Notably, we observed that beyond three components, the improvement in BIC scores diminishes, particularly on the SMD dataset. This observation aligns with our choice of components and shows how to use the BIC criterion as a principled method for determining the number of components.

Table 6: BIC values across datasets with different number of GMM components.

# components	MSL	SMAP	SMD
1	-11783.36	-15964.53	-166770.01
2	-12885.25	-17811.11	-190736.41
3	-13168.31	-17955.04	-193107.46
4	-13012.05	-17853.51	-193909.34
5	-13213.42	-17976.60	-194323.92

C.4 Case Study

Baseline. The table below shows the baseline performance of the current condition-based monitoring model at Amazon sites. The model is based on a combination of statistical and ISO thresholds for vibration sensors in Monitron data. However, the existing model in production cannot be easily extended to include more sensors.

Table 7: Baseline performance compared to M^{2}AD on Site A and Site B.

	Baseline		M^{2}AD	
	Precision	Recall	Precision	Recall
Sita A	19.2%	2.7%	35.3%	17.4%
Site B	23.1%	22.5%	26.9%	44.8%

Our experiments show up to 16% improvement in precision and 15% improvement in recall in Site A over the current baseline model, and an increase of 4% in precision and 22% in recall scores in Site B.

Experiment Setup. We apply a categorical transformation to availability and throughput. Availability is the percentage of time where the machine is available on planned production time, we convert this percentage to a binary value where $> 98\%$ indicates that the asset is available. As for throughput, which measures how many packages have been processed, is an asset-specific measure which we categorize into: low, medium, high based on quantile values of the training data. With respect to pre-processing, first the data is resampled to an hourly-level with the median aggregation. In addition, we scale the data into the range (-1, 1) and interpolate missing values. We select the hyperparameters of the model based on the nature of the problem and dataset. We choose a window size of 120, which is equivalent to 5-day span of historical values. Moreover, we use an area-error with $l = 2$, a small l is chosen as it determines the lag time needed before being able to execute the model. In our setting, we prioritize an earlier detection of anomalies. We train the LSTM model for 30 epochs and early stopping, which was enough for the model to converge in our preliminary experiments. For monitron sensors, we set the number of components for the GMM as $m_k = 1$. For amperage, we set the number of components to $m_k = 2$, having two components allowed us to fit the amperage errors better. More specifically, the pattern of amperage data changes between "operation" mode and "shutdown" mode, which exhibits a bimodal distribution.

Based on the hierarchy organization of the data show in Figure 5, and that one data source (monitron) out-numbers the number of sensors in the amperage system, we construct an equally divided weighing vector:

$$\lambda_k = \frac{1}{\#\text{ systems} \cdot \#\text{ sensors} \cdot \#\text{ summaries}}$$

For example, each measurement in acceleration will have: $\lambda_{\text{acceleration}} = 1/(2 \times 3 \times 6)$. This ensures that under-represented sensors, e.g. amperage, does not lose its contribution to the anomaly score.

C.5 Benchmark

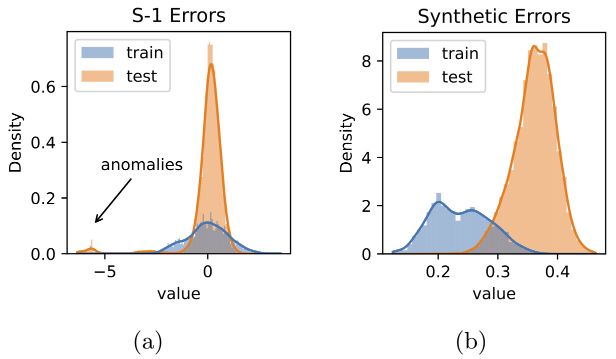
We run the benchmark on a total of 109 signals from MSL, SMAP, and SMD dataset. Table 8 shows the overall performance for the datasets used.

Table 8: Benchmark Results. The table summarizes precision (pre), recall (rec), F_1 and $F_{0.5}$ scores across NASA dataset. Each model is executed for 5 times and we record the mean score and standard deviation

Model		F_1	$F_{0.5}$	Precision	Recall
M^{2}AD	MSL	0.749 ± 0.032	0.752 ± 0.040	0.754 ± 0.046	0.744 ± 0.023
	SMAP	0.716 ± 0.021	0.656 ± 0.028	0.622 ± 0.031	0.845 ± 0.013
	SMD	0.913 ± 0.004	0.906 ± 0.005	0.904 ± 0.004	0.926 ± 0.001
USAD	MSL	0.700 ± 0.083	0.631 ± 0.098	0.593 ± 0.104	0.861 ± 0.028
	SMAP	0.650 ± 0.111	0.565 ± 0.128	0.521 ± 0.133	0.893 ± 0.012
	SMD	0.608 ± 0.004	0.644 ± 0.005	0.670 ± 0.007	0.557 ± 0.005
OA	MSL	0.562 ± 0.079	0.457 ± 0.076	0.406 ± 0.072	0.922 ± 0.036
	SMAP	0.577 ± 0.047	0.564 ± 0.069	0.557 ± 0.083	0.603 ± 0.008
	SMD	0.576 ± 0.006	0.471 ± 0.006	0.419 ± 0.006	0.923 ± 0.002
LSTM-DT	MSL	0.483 ± 0.014	0.408 ± 0.012	0.370 ± 0.011	0.694 ± 0.020
	SMAP	0.713 ± 0.018	0.677 ± 0.021	0.655 ± 0.023	0.782 ± 0.017
	SMD	0.433 ± 0.019	0.394 ± 0.015	0.373 ± 0.020	0.523 ± 0.073
LSTM-AE	MSL	0.493 ± 0.052	0.490 ± 0.061	0.489 ± 0.067	0.500 ± 0.044
	SMAP	0.672 ± 0.025	0.651 ± 0.035	0.638 ± 0.041	0.710 ± 0.013
	SMD	0.402 ± 0.015	0.395 ± 0.012	0.392 ± 0.011	0.412 ± 0.020
LSTM-VAE	MSL	0.492 ± 0.020	0.476 ± 0.021	0.466 ± 0.023	0.522 ± 0.023
	SMAP	0.649 ± 0.011	0.615 ± 0.012	0.595 ± 0.013	0.713 ± 0.016
	SMD	0.412 ± 0.008	0.411 ± 0.005	0.410 ± 0.007	0.415 ± 0.019
AT	MSL	0.391 ± 0.039	0.455 ± 0.061	0.516 ± 0.095	0.322 ± 0.050
	SMAP	0.359 ± 0.031	0.387 ± 0.076	0.415 ± 0.121	0.334 ± 0.044
	SMD	0.533 ± 0.012	0.470 ± 0.019	0.436 ± 0.023	0.689 ± 0.022
TimesNet	MSL	0.276 ± 0.038	0.208 ± 0.031	0.179 ± 0.027	0.611 ± 0.071
	SMAP	0.169 ± 0.018	0.117 ± 0.014	0.098 ± 0.012	0.630 ± 0.036
	SMD	0.409 ± 0.015	0.310 ± 0.013	0.266 ± 0.012	0.887 ± 0.028
FITS	MSL	0.386 ± 0.011	0.338 ± 0.010	0.312 ± 0.010	0.506 ± 0.012
	SMAP	0.136 ± 0.007	0.097 ± 0.005	0.082 ± 0.004	0.397 ± 0.020
	SMD	0.316 ± 0.012	0.286 ± 0.013	0.269 ± 0.013	0.383 ± 0.011

D Limitations

While we assume that the training data is free from anomalies and both the training and testing data exhibit similar patterns, that it is not always true in practice. If the errors observed follow our assumptions, we see a case similar to Figure 6a where there are error scores that greatly deviate from the expected error observed during the training phase. On the other hand, when the testing data differs significantly, like we observe in Figure 6b of a synthetically generated time series, a larger number of observations will be marked as "anomalous". This is a clear limitation of our model, triggering a need to retrain the origin model to conform to the new distribution of the data.



(a) (b)

Figure 6: Model errors on (a) S-1 signal from SMAP dataset, where the anomalous errors are clearly separated from normal errors. (b) synthetic data were the testing error values do not have a similar distribution to training errors.