Towards Real-Time Detection of Anomalous Behavior in Crowds from Accelerometer Time Series

Bahar Kor[1], Bipin Gaikwad[1], Abani Patra[2], Eric L. Miller[1]
[1]*Dept. of Electrical and Computer Engineering, Tufts University, Medford, USA*
[2]*Dept. of Computer Science, Tufts University, Medford, USA*
email: first.last@tufts.edu

Abstract—**Here we propose an unsupervised anomaly detection framework for identifying abnormal dynamics in multi-agent systems from multivariate accelerometer observations. The approach, which is agnostic to the number of agents providing data, employs neural autoencoder architectures to model normal behavior at the level of the individual entity. To detect meaningful crowd-level anomalies, we compute the mean of the reconstruction error across all entities at each time step. This aggregation mitigates the influence of isolated or transient anomalies in individual entities, reducing the likelihood of false positives and highlighting prominent irregularities in collective behavior. Unlike video-based methods, which require complex real-time processing and face challenges from, e.g., poor visibility and occlusions, we leverage crowd-sourced accelerometer data with lightweight processing, avoiding external assumptions or complex models. The proposed autoencoder architecture further reduces training time while maintaining efficiency. Experiments using a new synthetic crowd movement dataset, generated through simulation specifically for crowd anomaly detection, along with another dataset, demonstrate the method's effectiveness and high precision in detecting anomalous behaviors. Given the lack of publicly available datasets in this domain, our dataset fills a critical gap, offering a valuable resource for advancing research in this area.**

Index Terms—**Crowds, Anomaly Detection, Multi-Entity, Multivariate time series.**

I. Introduction

Anomaly detection in multivariate time series data is a critical problem in various fields, including industrial systems [1], finance [2], and healthcare [3]. The objective is to identify deviations from normal behavior, which can indicate underlying entity failures or other irregularities. This task is particularly challenging because in most real-world scenarios we only have access to "normal" data, and anomalies can take many forms that are not predefined. As such, the problem becomes one of detecting changes in system behavior without prior knowledge of how these changes might manifest, which makes anomaly detection a highly complex problem. Many applications of practical interest require online methods, where the model continuously processes incoming data and monitors the system for any signs of anomalous behavior. In the context of industrial systems, for example, real-time identification of critical events, such as system failures, is required rather than post-analysis of historical data. Such is also the case for the primary application driving this work: detecting anomalous behavior in multi-agent systems; that is, crowds, from time series provided by the individual agents. Our approach is motivated by security applications where video data is not available but we can acquire "data streams of opportunity", such as accelerometer time series, from members of a community (e.g. school, commuters at a transit station, etc.) who opt into the use of such a system [4]. In such cases, real-time anomaly detection is crucial for minimizing risks and improving decision-making.

Challenges: This work focuses on crowd anomaly detection, where the challenge is compounded by the fact that the number of entities contributing to the multivariate data (e.g., individuals) is not known a priori and may change over time. Most multivariate time series anomaly detection methods are designed for entity-level analysis, but crowd surveillance must account for diverse behavioral variations that do not necessarily indicate anomalies. Moreover, these methods fail to balance detection performance with computational overhead [5], which poses challenges in maintaining efficiency in real-time performance in large-scale crowd monitoring.

Existing Methods and Limitations: Existing crowd anomaly detection methods, such as SIMulated crowd data for anomaly detection and prediction (SIMCD) [6] and Anomaly Detection Using Hierarchical Temporal Memory in Crowd Management (HTM) [7], rely on aggregating individual-level features (e.g., speed, heading) into global averages and using higher-level metrics such as crowd density, agent count, and level of crowdness. However, these approaches face several limitations. Natural variations in behavior can cause false positives or negatives, making it difficult to distinguish genuine anomalies from typical crowd dynamics. Additionally, extracting such features often requires video or similar input sources, which poses challenges such as computational overhead and potential privacy concerns. Similarly, the method in [8] extracts statistical features from continuous acceleration and discrete motion data (e.g., mean, skewness, standard deviation, among others).

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Awards 22STESE00001-01-00, 22STESE00001-02-00, and 22STESE00001-03-02. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security. ELM acknowledges support from the National Science Foundation's ID/R Program.

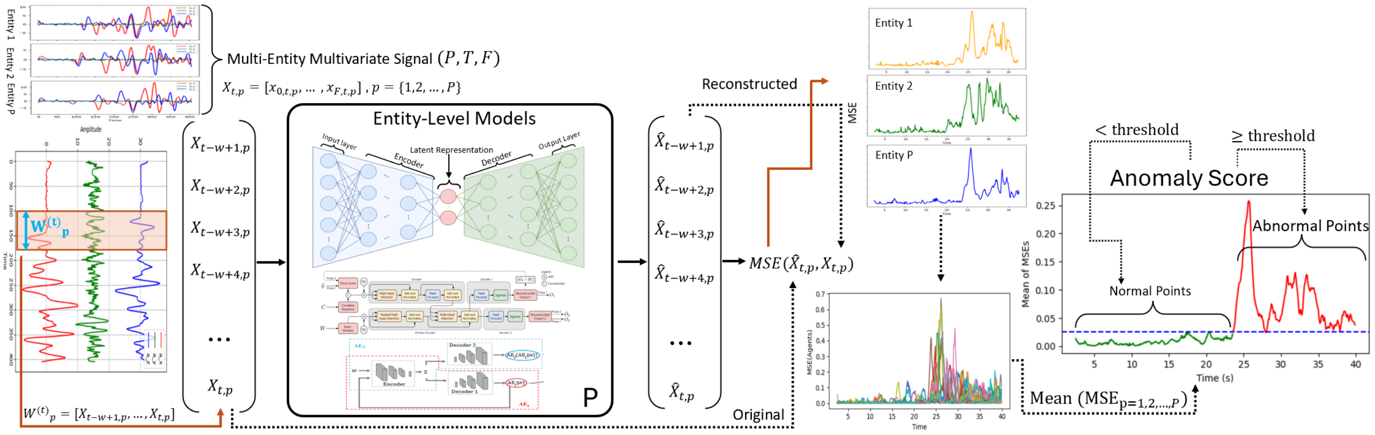

Fig. 1. **Proposed method pipeline.** Anomaly detection framework utilizing an entity-level model for processing F sensor data streams over T points in time from P entities. The data for each of the P agents is fed into the pre-trained model, and the reconstruction error is computed using Mean Squared Error (MSE) across the F sensors. A unified anomaly score is obtained by aggregating the MSE across entities at each time step. A thresholding approach is applied to label data points.

Although this reduces data dimensionality and facilitates the use of basic classifiers, such as Random Forest and KNN, which were employed in their study, it oversimplifies the rich, dynamic information present in raw sensor data. Moreover, this approach relies on supervised learning, where ground truth labels for normal and anomalous events guide the model's learning. However, labeled crowd anomaly data are often scarce, difficult to obtain, and subjective, limiting the model's generalization to new or unseen anomalies. Furthermore, periodic retraining with updated labels is resource-intensive and restricts scalability and real-time applicability.

Contribution: In this work, shown in Figure 1, we propose a novel crowd anomaly detection approach that addresses key limitations of existing methods such as [6]–[8]. We feed crowd-sourced data into a model to learn the underlying normal crowd dynamics. By calculating the reconstruction error for each individual, we capture localized anomalies that may be overlooked when aggregating features into global averages. To compute a comprehensive anomaly score, we calculate the mean reconstruction error across all entities at each time step to capture the collective behavior of the crowd and detect global anomalies. The final step involves comparing the anomaly score with a threshold to distinguish normal from abnormal behavior.

The remainder of this paper is organized as follows. Section II defines the problem and Section III reviews related work on multivariate anomaly detection at the entity level. In Section IV, we present the proposed method, including the autoencoder, statistical analysis, and thresholding technique. Section V describes the new crowd dataset generated using Unity. Sections VI and VII outline the experimental setup, evaluation metrics, and results.

II. Problem Formulation

Here we consider data generated by P entities, each associated with F sensors (e.g., an accelerometer would provide $F = 3$ signals corresponding to the acceleration in each of three orthogonal directions). For entity $p \in [P] \equiv \{1, 2, \dots, P\}$, at time step $t \in [T]$, $x_{s,t,p}$ is the datum collected from sensor s, and $X_{t,p} \in \mathbb{R}^F$ denotes the vector containing all F sensor's reading. The complete dataset forms a tensor $\mathcal{X} \in \mathbb{R}^{F \times T \times P}$.

To make the model more robust, we normalize the data using z-score normalization and convert it to time-series windows, both for training and testing. We normalize the time series data as follows:

$$x_{s,t,p} \leftarrow \frac{x_{s,t,p} - \mu_s}{\sigma_s} \qquad (1)$$

where μ_s and σ_s are the mean and standard deviation of sensor s readings from train set, respectively.

To model the dependence of a data point $X_{t,p}$ at time step t, we consider a window of size w. For $t \geq w$ the data from entity p over a window of size w consecutive time steps is represented as:

$$W_p^{(t)} = [X_{t-w+1,p}, X_{t-w+2,p}, \dots, X_{t,p}] \in \mathbb{R}^{F \times w}. \qquad (2)$$

For $t < w$, we apply replication padding [9] to ensure that the window length remains consistent at w. For all entities $p \in [P]$, the data from all F sensors with the t-th window is aggregated into a tensor $\mathcal{S}_t \in \mathbb{R}^{F \times w \times P}$ whose frontal faces [10] are given by $W_p^{(t)}$.

We aim to detect anomalies in the behavior of P entities over time, specifically by identifying deviations from the normal collective behavior of the system. An anomaly represents a significant deviation in the signals from one or more entities that differs from the expected behavior of the entire system. Although we compute an anomaly score for each point within the window, we focus solely on the score at the last time step to enable real-time anomaly detection and respond to anomalies quickly [11]. Formally, the problem can be described as determining a mapping $A(\mathcal{S}_t) : \mathbb{R}^{F \times w \times P} \rightarrow \{0, 1\}$. The function $A(\mathcal{S}_t)$ returns a single binary value for each time step t, where a zero means that the crowd behavior at time t is normal, while a one denotes an anomalous behavior.

III. Related Works at the Entity-Level

Multivariate time series anomaly detection has drawn significant research attention, leading to models that detect complex patterns and anomalies at the entity level. We review two state-of-the-art models, USAD [12] and TranAD [9] which address challenges like temporal dependencies, multivariate correlations, and subtle anomaly detection in complex patterns.

USAD [12] utilizes a dual autoencoder structure with a shared encoder and two decoders to learn patterns in multivariate time series without labeled anomalies. It trains in two phases: a conventional autoencoder phase followed by an adversarial phase that amplifies reconstruction errors for anomalous data, helping to distinguish normal from abnormal patterns. USAD captures complex relationships within multivariate time series data with minimal supervision.

TranAD [9] is a Transformer-based anomaly detection model designed to address the limitations of recurrent neural networks by capturing both short- and long-range dependencies in time series. Using a self-attention mechanism, it assigns dynamic weights to each time step, highlighting crucial parts of the sequence to detect subtle anomalies. With an encoder-decoder architecture, TranAD reconstructs the input and identifies anomalies through reconstruction errors. Its ability to capture global dependencies makes it ideal for complex temporal tasks.

IV. Proposed Method

Similar to [13], our method uses an autoencoder-based architecture to capture normal patterns in the data, followed by statistical analysis to compute the mean reconstruction error across individuals for anomaly assessment. Anomalies are detected when the calculated anomaly score exceeds a predefined threshold, indicating abnormal crowd behavior. Unlike methods such as [6]–[8], which rely on video or more complex processes, our approach processes each entity's data stream and aggregates the reconstruction errors across multiple entities. It enables the detection of collective anomalies without requiring video processing or extensive feature engineering. The method consists of three stages: Autoencoder for learning normal patterns, anomaly score calculation, and anomaly detection.

Autoencoder: As illustrated in Fig. 1, we consider three schemes for entity-based data analysis: USAD, TranAD, and a straightforward autoencoder with dense layers. At each time window indexed by t, and for the p-th entity, these models process $W_p^{(t)}$ and output a reconstruction of the input, from which a scalar reconstruction error is computed for each timestep within the window. Reconstruction errors are calculated for each entity p at each time step t within a window, based on the output generated by the decoder. The mean squared error (MSE) between the reconstructed feature vector $\hat{X}_{t,p}$ and the input feature vector, $X_{t,p}$ is used as a function of time and entity for further processing. Specifically, using the notation at the start of Section II, the sensor-averaged mean squared error for entity p at time t is calculated as follows:

$$e_{t,p} = MSE(\hat{X}_{t,p}, X_{t,p}) = \frac{1}{F}\sum_{s=1}^{F}(\hat{x}_{s,t,p} - x_{s,t,p})^2 \tag{3}$$

Anomaly Score: To compute the crowd-wide anomaly score, μ_t, we calculate the mean of MSE values, $e_{t,p}$, across all P entities at each time step t within the window :

$$\mu_t = \frac{1}{P}\sum_{p=1}^{P} e_{t,p} \tag{4}$$

This aggregation allows us to capture the collective crowd behavior, as individual entity anomalies might not fully reflect true anomalies in the crowd. For example, a temporary increase in the reconstruction error of one entity may not indicate a crowd anomaly if the reconstruction errors of other entities remain stable.

Anomaly Detection: A time point is detected as an anomaly if the calculated anomaly score, μ_t, exceeds a specified threshold. Specifically, if the anomaly score at time t is greater than a threshold $\tau > 0$, then the behavior of the systems at that time step is anomalous:

$$A(\mathcal{S}_t) = \begin{cases} 1 & \text{if } \mu_t \geq \tau \\ 0 & \text{if } \mu_t < \tau \end{cases} \tag{5}$$

V. Datasets for Evaluation

The proposed method is evaluated on two datasets: a synthetic crowd motion dataset and the publicly available Tennessee Eastman Process (TEP) dataset [14]. Due to the lack of suitable open crowd accelerometer dataset, we generate our synthetic crowd dataset to assess the effectiveness of our method in detecting anomalies. While not a crowd detection dataset, TEP is widely used for anomaly detection [15]. We use TEP to benchmark our approach against recent methods and demonstrate its potential applicability to other domains. This combination of controlled crowd scenarios and industrial process data allows a comprehensive assessment of the robustness and adaptability of the framework.

Crowd Simulation: This study uses Unity®, a cross-platform game engine developed by Unity Technologies, to simulate crowd behavior in a synthetic train station scenario. The simulation includes both normal and abnormal scenarios, as shown in Fig. 2, with adjustable parameters such as the number of agents, time of arrival, and appearance rate to model different crowd sizes and behaviors. This provides a flexible environment for testing various situations.

In the normal scenario, agents follow a predictable sequence of activities: arriving at the entrance gates, moving toward the train door, and then proceeding to the exit gate. In the abnormal scenario, an object simulates an emergency, causing agents to abandon their usual behaviors and rush toward the exit gate for evacuation, mimicking real-world emergency reactions. To increase realism, two types of agents are used: active agents, which provide accelerometer data, and passive agents, which have random movement patterns that disrupt the active agents. This randomization reflects diverse real-world crowd behaviors.

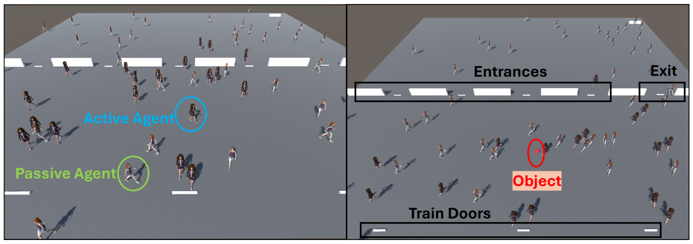

Fig. 2. Normal (left) and abnormal (right) scenarios in a crowd simulation within a simplified train station scene. The entrance gates, exit gate, train doors, and the object are labeled for clarity.

During simulation, the x, y, and z coordinates of each agent's upper left leg are recorded at each time step. These positional data are then smoothed using a Savitzky-Golay filter [16] to reduce artifacts from the Unity simulation. A second-order numerical differentiator is applied to generate accelerometer data, simulating a device in the agent's pocket. The smoothing step is essential for improving data quality by minimizing noise and inconsistencies, similar to challenges faced with real-world data. All data, along with detailed documentation on the simulation setup and post-processing techniques, are available in the GitHub repository https://github.com/bkor-git/AD-Project, enabling further exploration and replication of this study.

Tennessee Eastman Process: The TEP data set [15] consists of 52 multivariate time series representing operational features (e.g., temperature, pressure, flow rates). The dataset includes one training set with 500 observations of normal operation and a test set each containing 960 observations for 21 fault scenarios. For the purpose of model validation, we treat the dataset as a special case by considering it as a single entity, where all 52 features are treated as sensor readings. Faults 3, 9, and 15 are excluded due to their lack of observable deviations [17]. This special case setup allows for the assessment of the model's ability to detect system-wide anomalies in the context of individual entities, providing a foundation for further evaluation in multi-entity settings.

VI. Training Setup

In this study, we employ TranAD, USAD, and a simple autoencoder (SAE) architecture with dense layers to capture normal patterns in the data. SAE and USAD utilize the Adam optimizer [18], while TranAD follows its original implementation with AdamW. For SAE, we optimize the mean squared error (MSE) loss with an initial learning rate of 1×10^{-4} and adjust it dynamically using the ReduceLROnPlateau callback with a patience of 5, a factor of 0.5, and a minimum learning rater of 1×10^{-6}. Training is performed with a batch size of 128 for 50 epochs, applying early stopping if the validation loss does not improve after 5 consecutive epochs. Under these conditions, SAE loss converge in ~ 15 epochs. A dropout rate of 0.1 is applied after each layer to prevent over-fitting. We fine-tune hyperparameters, including hidden units and dropout rate, using grid search. For TranAD and USAD this results in an initial learning rate of 1×10^{-3} for these models. While the batch size remains the same, we find that training for 10 epochs is sufficient to achieve comparable performance.

VII. Experimental Results

Evaluation Metrics: To evaluate our approach, we use the **F1 Score** and the area under the receiver operating characteristic curve (**AUROC**), which are effective for assessing classification accuracy and the trade-off between true and false positives [9], [19]. For the non-anomalous dataset, all ground truth labels are 0, and for the anomalous dataset, data points are labeled 0 before the anomaly and 1 after. To classify the anomaly score, we experimentally determine an optimal global threshold on the test set anomaly scores by evaluating multiple threshold values, selecting the one that maximizes the F1 score [12]. The F1-Score is the harmonic mean of precision (P) and recall (R):

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \quad \mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad F1 = 2 \cdot \frac{\mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}}$$

Here, TP represents the truly detected anomalies, FP stands for the falsely detected anomalies, and FN is the misclassified normal samples. These metrics ensure a robust evaluation and enable consistent comparisons with state-of-the-art methods.

TABLE I
Results in terms of traditional performance metrics of evaluated state-of-the-art methods (AUROC, F1-score).

Method	TEP		Crowd	
	AUROC	F1	AUROC	F1
USAD	0.929	0.893	0.990	0.948
TranAD	0.928	0.893	0.989	0.944
Ours (SAE)	**0.948**	**0.917**	**0.993**	**0.954**

Results: The performance of our SAE method was evaluated on the TEP and Crowd dataset, alongside TranAD and USAD, as state-of-the-art baselines for entity-level anomaly detection. Most existing methods for multivariate time series anomaly detection utilize a point adjustment approach, originally proposed by [20], to calculate performance metrics. However, its reliance on ground truth labels limits its applicability in real-world scenarios, where such labels are often unavailable. To address this, we present the results without point adjustment, which allows for an assessment that aligns more closely with real-world anomaly detection scenarios. Table I presents the results in terms of AUROC and F1-Score. While all the methods demonstrate strong performance, our approach achieves the highest AUROC across both dataset: 0.948 on the single-entity dataset TEP, and 0.993 on the multi-entity dataset Crowd. Additionally, SAE achieves the highest F1-score on both datasets, demonstrating a better balance between precision and recall. Despite its simple autoencoder-based architecture, our method slightly outperforms USAD and TranAD, suggesting its ability to learn robust anomaly representations without the added complexity of advanced architectures. These results highlight the effectiveness of our proposed model not only in multi-entity scenarios but also in single-entity cases, demonstrating its broad applicability in real-world multivariate anomaly detection tasks.

Training Time: To evaluate computational efficiency, we compared its training time per epoch with that of existing baseline methods in the same computational environment.

TABLE II
COMPARISON OF TRAINING TIMES (SECONDS/EPOCH) FOR THE CROWD DATASET.

Method	USAD	TranAD	Ours
Training Time (s/epoch)	2020.75	85.16	**11.04**

Table II shows significant differences in the efficiency of training time between methods. In particular, the proposed method achieves a training time of 11.04 seconds per epoch, making it approximately 7.7 times faster than TranAD and 183 times faster than USAD. This substantial efficiency gain, primarily due to the simplicity of the model architecture, underscores its suitability for real-time and resource-constrained applications.

VIII. CONCLUSION

In this study, we develop and evaluate an autoencoder-based anomaly detection model for multi-entity multivariate time series data, with a specific focus on identifying crowd anomalies using synthetic accelerometer data. The model analyzes changes in the MSE of entities over time to detect anomalous behavior. Comparative evaluation against state-of-the-art methods, such as TranAD and USAD, demonstrated the effectiveness of the proposed approach in identifying anomalous events within crowd scenarios and enabling fast training. This work lays the foundation for several promising research directions. First, integrating accelerometer data with additional sensor modalities, such as gyroscope and Wi-Fi signals, could enhance the robustness and reliability of the model in real-world crowd-monitoring applications. Furthermore, while the current study emphasizes anomaly detection, future research will investigate advanced change point detection techniques. Specifically, applying these methods to the anomaly score may significantly improve the precision and timeliness of detecting critical changes in complex and dynamic environments.

REFERENCES

[1] L. Jiang, H. Xu, J. Liu, X. Shen, S. Lu, and Z. Shi, "Anomaly detection of industrial multi-sensor signals based on enhanced spatiotemporal features," *Neural Computing and Applications*, vol. 34, no. 11, pp. 8465–8477, 2022. [Online]. Available: https://doi.org/10.1007/s00521-022-07101-y

[2] S. O. Pinto and V. A. Sobreiro, "Literature review: Anomaly detection approaches on digital business financial systems," *Digital Business*, vol. 2, no. 2, p. 100038, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666954422000187

[3] O. Salem, Y. Liu, A. Mehaoua, and R. Boutaba, "Online anomaly detection in wireless body area networks for reliable healthcare monitoring," *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, pp. 1541–1551, 09 2014.

[4] A. Aldayri and W. Albattah, "Taxonomy of anomaly detection techniques in crowd scenes," *Sensors*, vol. 22, no. 16, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/16/6080

[5] X. Yang, E. Howley, and M. Schukat, "Adt: Time series anomaly detection for cyber-physical systems via deep reinforcement learning," *Computers Security*, vol. 141, p. 103825, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404824001263

[6] A. Bamaqa, M. Sedky, T. Bosakowski, B. Bakhtiari Bastaki, and N. O. Alshammari, "Simcd: Simulated crowd data for anomaly detection and prediction," *Expert Systems with Applications*, vol. 203, p. 117475, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422008065

[7] A. Bamaqa, M. Sedky, T. Bosakowski, and B. B. Bastaki, "Anomaly detection using hierarchical temporal memory (htm) in crowd management," in *Proceedings of the 2020 4th International Conference on Cloud and Big Data Computing*, ser. ICCBDC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 37–42. [Online]. Available: https://doi.org/10.1145/3416921.3416940

[8] M. Irfan, L. Tokarchuk, L. Marcenaro, and C. Regazzoni, "Anomaly detection in crowds using multi sensory information," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2018, pp. 1–6.

[9] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *CoRR*, vol. abs/2201.07284, 2022. [Online]. Available: https://arxiv.org/abs/2201.07284

[10] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009. [Online]. Available: https://doi.org/10.1137/07070111X

[11] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, and H. Qiao, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 187–196. [Online]. Available: https://doi.org/10.1145/3178876.3185996

[12] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "Usad: Unsupervised anomaly detection on multivariate time series," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 3395–3404. [Online]. Available: https://doi.org/10.1145/3394486.3403392

[13] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=BJJLHbb0-

[14] C. A. Rieth, B. D. Amsel, R. Tran, and M. B. Cook, "Additional Tennessee Eastman Process Simulation Data for Anomaly Detection Evaluation," 2017. [Online]. Available: https://doi.org/10.7910/DVN/6C3JR1

[15] F. Xue and W. Yan, "Multivariate time series anomaly detection with few positive samples," in *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1–7.

[16] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, pp. 1627–1639, Jan. 1964.

[17] M. Gorman, X. Ding, L. Maguire, and D. Coyle, "Anomaly detection in batch manufacturing processes using localized reconstruction errors from 1-d convolutional autoencoders," *IEEE Transactions on Semiconductor Manufacturing*, vol. 36, no. 1, pp. 147–150, 2023.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[19] Z. Tian, M. Zhuo, L. Liu, J. Chen, and S. Zhou, "Anomaly detection using spatial and temporal information in multivariate time series," *Scientific Reports*, vol. 13, 03 2023.

[20] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, and H. Qiao, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," *CoRR*, vol. abs/1802.03903, 2018. [Online]. Available: http://arxiv.org/abs/1802.03903