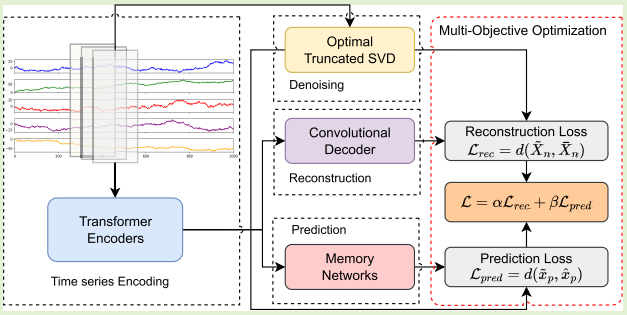
MTAD: Multiobjective Transformer Network for Unsupervised Multisensor Anomaly Detection

Mohammed Ayalew Belay, *Graduate Student Member, IEEE*, Adil Rasheed, and Pierluigi Salvo Rossi, *Senior Member, IEEE*

***Abstract*—Multisensor anomaly detection plays a crucial role in several applications, including industrial monitoring, network-intrusion detection, and healthcare monitoring. However, the task poses significant challenges due to the presence of massive unlabeled data, the difficulty of identifying normal patterns in the spatio-temporal data, and the inherent complexity of defining an anomaly. Moreover, noisy sensor measurements could potentially result in models erroneously detecting noise as an anomaly, and the existence of different types of anomalies adds to the complexity. Existing multisensor anomaly detection methods are mostly designed for labeled datasets and often disregard crucial factors such as spatio-temporal dependencies, noise presence in training data, and the existence of multiple types of anomalies; thus, their applicability is limited. In this article, we propose a novel framework called multiobjective transformer networks for anomaly detection (MTAD) that leverages the power of transformer architectures and optimal truncated singular value decomposition (OT-SVD) for robust unsupervised multisensor anomaly detection. MTAD comprises a multihead transformer encoder for effective time series representation learning, a convolutional decoder for reconstruction, and a memory network for predictive analysis. The model processes denoised (via OT-SVD) input through the network and computes both reconstruction and prediction losses. MTAD jointly optimizes the modules in an end-to-end mechanism to minimize the combined weighted loss. We compare MTAD with other state-of-the-art methods using several metrics and demonstrate that our approach outperforms existing solutions. Furthermore, we conducted an ablation to demonstrate the contribution of each module to the overall performance.**



***Index Terms*—Multiobjective training, multisensor anomaly detection, optimal truncated singular value decomposition (OT-SVD), transformer encoders, unsupervised learning.**

I. Introduction

ANOMALY detection in multisensor systems has gained significant importance across various domains, such as industrial monitoring [1], [2], network-intrusion detection [3], [4], wireless sensor networks [5], medical applications [6], and autonomous vehicles [7]. Multisensor systems generate complex, high-dimensional (multivariate) data streams that capture the temporal evolution of several physical parameters, potentially enabling anomaly detection and system behavior prediction. In industrial applications, multisensor anomaly detection is crucial for health or safety monitoring of industrial systems such as manufacturing processes, power plants, and oil refineries. Detecting anomalies can help to identify system faults, predict equipment failures, and improve overall operational efficiency. Similarly, in network-intrusion detection, anomaly detection algorithms can identify unusual network traffic patterns or user behavior that may indicate a potential cyber-attack. The continuous collection of massive multisensor data led to the development of several data-driven machine learning methods for anomaly detection. These techniques are implemented through supervised, semi-supervised, or unsupervised learning modes [8]. The first two approaches face several challenges including the limited availability of labeled data in real-world scenarios. Furthermore, supervised and semi-supervised approaches are specialized in recognizing specific anomalies from training data and are likely to fail to detect new types. Dynamic environments require periodic

Manuscript received 8 March 2024; accepted 27 April 2024. Date of publication 9 May 2024; date of current version 14 June 2024. This work was supported in part by the Research Council of Norway through the project DIGITAL TWIN within the PETROMAKS2 Framework under Project 318899. The associate editor coordinating the review of this article and approving it for publication was Dr. Zhenghua Chen. *(Corresponding author: Mohammed Ayalew Belay.)*

Mohammed Ayalew Belay is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7034 Trondheim, Norway (e-mail: mohammed.a.belay@ntnu.no).

Adil Rasheed is with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7034 Trondheim, Norway (e-mail: adil.rasheed@ntnu.no).

Pierluigi Salvo Rossi is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7034 Trondheim, Norway, and also with the Department of Gas Technology, SINTEF Energy Research, 7491 Trondheim, Norway (e-mail: salvorossi@ieee.org).

Digital Object Identifier 10.1109/JSEN.2024.3396690

retraining with new labeled data, and data scarcity can lead to overfitting and limited generalization. As a result of these challenges, unsupervised anomaly detection techniques are increasingly gaining importance in multisensor anomaly detection [9]. These methods utilize unlabeled data, cope well with imbalanced data problems, detect previously unobserved anomalies, and effectively adapt to non-static environments where data distributions are changing.

A. Related Work

Unsupervised anomaly detection methods can be categorized into three groups: 1) conventional methods relying on classical statistical and/or machine learning algorithms; 2) deep neural network (DNN) methods; and 3) hybrid models combining approaches from the two previous groups [9].

Conventional methods include autoregressive models [10], control charts [11], discord search [12], and clustering (K-means, DBSCAN, LOF) [13], [14], [15]. Their main advantages are simplicity and interpretability. However, they often struggle with high dimensionality and nonlinearities, thus their effectiveness in complex multisensor environments is limited.

DNN-based models utilize recurrent neural networks (RNNs) [16], [17], [18], [19], convolutional neural networks (CNNs) [20], [21], autoencoders [22], [23], [24], [25], generative adversarial networks [26], [27], and graph neural networks [28], [29]. These methods excel at capturing complex nonlinear interactions and temporal correlations. However, their main disadvantage is the requirement for large datasets and high computational resources. Also, they often exhibit limited interoperability.

Hybrid models combine conventional statistical methods and machine learning with deep learning for robust anomaly detection. Examples include long short-term memory networks with variational autoencoders [30], OmniAnomaly [31], multiscale convolutional recurrent encoder–decoder networks [32], deep autoencoding Gaussian mixture models (DAGMMs) [33], and adversely trained autoencoders [34]. Hybrid models can be complex to implement and require careful tuning to achieve optimal results. These models mostly employ CNNs and RNNs for spatio-temporal representation learning, though sequential input makes RNNs training slow and long-term representations are challenged by vanishing-gradient problems. Transformer models [35] are capable of dealing effectively with both issues via multihead self-attention mechanisms that enable parallel sequence processing and have become state-of-the-art for sequential modeling of multisensor time series. As a result, several transformer-based models have been proposed for anomaly detection [36], [37], [38], [39], [40], [41], [42], [43], [44].

B. Challenges and Paper Contribution

Although several unsupervised multisensor anomaly detection algorithms have been proposed over the years, significant challenges are still relevant and require improved approaches.

1) Detection of Multiple Types of Anomalies: Anomalies in sensor measurements can be point, contextual, or sub-sequence (collective) anomalies. *Point anomalies* occur when a single sensor measurement significantly deviates from the rest, often due to unreliable sensors or localized operational issues. *Contextual anomalies* involve unusual measurements within a specific context. *Collective anomalies* involve sub-sequences of sensor measurements behaving differently, making detection difficult for conventional methods. Existing methods often identify a single type of anomaly, thus limiting their broad applicability.

2) Noise Removal From Normal Training Data: In many real-world scenarios, sensor measurements are inherently noisy, leading to a higher rate of false positives (FPs) where noise is mistaken for anomalies. Noise can distort the actual underlying structure of the data, making it difficult for unsupervised algorithms to model normal behavior accurately. Noise in multisensor anomaly detection is often overlooked.

3) Spatio-Temporal Correlation: Spatio-temporal data involves both temporal (recorded over time) and spatial (recorded across different sensors or locations) correlation (or more generally, nonlinear dependencies). Since anomalies frequently display unique patterns among multiple sensor measurements, identifying these correlations is crucial.

However, several approaches treat sensor readings independently, which significantly reduces their effectiveness.

4) Multiple Anomaly Scoring and End-to-End Training: Multiple anomaly scoring allows the detection of different types of anomalies that a single scoring approach might miss. Implementing an end-to-end training approach with multiple scoring methods simplifies the training process, avoids local optima, and enhances performance. However, most of the algorithms are based on single anomaly scoring and disjoint training, which compromises the performance of the models.

To address the aforementioned issues, we propose a novel framework named multiobjective transformer networks for unsupervised multisensor anomaly detection (MTAD). The proposed method processes the noisy input using a denoising approach based on singular value decomposition (SVD) analysis and combines two networks focusing on reconstruction and prediction, respectively. The framework jointly optimizes the networks in an end-to-end approach with the goal of minimizing the combined reconstruction loss and prediction loss. This comprehensive training strategy improves the performance of the anomaly detection model, addressing issues that often limit other methods.

Specifically, our contribution is summarized as follows.

1) We proposed a novel modular framework called MTAD that employs an encoder–decoder-memory network designed to identify multiple types of anomalies in sensor measurements effectively. MTAD incorporates a reconstruction network for detecting point anomalies and a predictive memory network for detecting sub-sequence anomalies.
2) We designed MTAD to effectively handle noisy sensor measurements by using optimal truncated SVD (OT-SVD).
3) The proposed framework employs a multihead self-attention network to manage spatio-temporal correlations between sensors effectively, thereby

improving its ability to detect anomalies that display irregular patterns.

4) The MTAD framework merges multiple anomaly scoring techniques in an end-to-end training procedure for robust anomaly detection.
5) We performed an extensive performance comparison with several other state-of-the-art methods using various publicly available datasets.

The rest of the article is structured as follows: Section II describes the proposed model and the mathematical tools behind its design; Section III presents the experimental setup and the datasets; while Section IV illustrates the performance analysis and related discussion; finally, conclusions and future research directions are given in Section V.

II. Proposed Method

In this section, we present the problem statement and provide a detailed description of the MTAD architecture, along with the mathematical theory of OT-SVD. Furthermore, we present combined scoring and end-to-end optimization approaches employed by MTAD. Finally, we describe the training and inference algorithms for implementing the MTAD model.

A. Problem Statement

We consider a system with S sensors and denote $x_k[n] \in \mathbb{R}$ the measurement of the kth sensor at discrete time n. The *measurement vector* $\boldsymbol{x}[n] = (x_1[n], x_2[n], \ldots, x_S[n])^T \in \mathbb{R}^S$ collects all the measurements at discrete time n, and the entire set of measurement vectors related to N discrete times steps is arranged into the *measurement matrix* $\boldsymbol{X} = (\boldsymbol{x}[1], \boldsymbol{x}[2], \ldots, \boldsymbol{x}[N]) \in \mathbb{R}^{S\times N}$.

In the context of unsupervised methods, we assume that a training measurement matrix ($\boldsymbol{X}_{\text{train}}$) is available and is representative of the system behavior under normal conditions. The training process aims to create an accurate representation $\mathcal{G}(\cdot)$ that captures the normal behavior of the system. For performance evaluation, a testing measurement matrix ($\boldsymbol{X}_{\text{test}} \in \mathbb{R}^{S\times M}$) with $M \ll N$ is available. The testing measurement matrix includes data related to both normal and anomalous conditions. Additionally, we assume that side information (labels) for the testing measurement matrix is available in the form of the *label vector* $\boldsymbol{y} = (y_1, y_2, \ldots, y_M)^T$, with $y_m = 1$ (resp. $y_m = 0$) denoting the presence (resp. absence) of an anomaly at discrete time m. The objective of the model is to find a representation such that $\hat{\boldsymbol{y}} = \mathcal{F}(\boldsymbol{X}_{\text{test}})$ closely resembles the vector $\boldsymbol{y}$ according to a predefined metric.

B. Architecture Overview

The proposed framework introduces a novel and modular twofold architecture based on prediction and reconstruction tasks, addressing the challenges of unsupervised anomaly detection in multisensor systems. The MTAD architecture is depicted in Fig. 1 and consists of four key components: 1) a stack of transformer encoders for capturing spatio-temporal patterns; 2) a convolutional decoder for the reconstruction task; 3) a prediction network for the predictive task; and

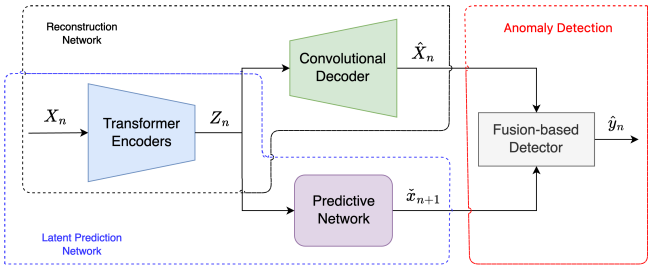


Fig. 1. MTAD architecture.

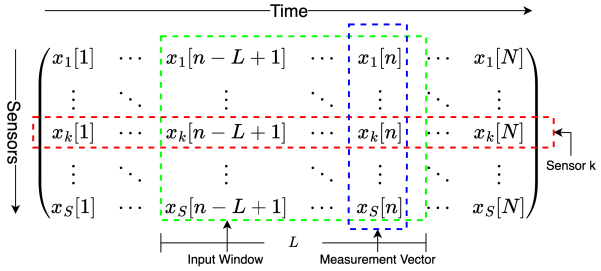


Fig. 2. Input structure.

4) a fusion-based detector for anomaly detection combining the anomaly scores from both reconstruction and prediction tasks. With reference to the generic measurement $x_k[n]$, three different estimates are produced within the architecture: 1) by the convolutional decoder, denoted $\hat{x}_k[n]$; 2) by the prediction network, denoted $\check{x}_k[n]$; and 3) by the OT-SVD block, denoted $\tilde{x}_k[n]$. Estimated measurement and vector and matrices are then denoted accordingly.

The system operates on the basis of a sliding-window mechanism collecting sensor measurements. We denote $\boldsymbol{X}_n = (\boldsymbol{x}[n], \boldsymbol{x}[n-1], \ldots, \boldsymbol{x}[n-L+1]) \in \mathbb{R}^{S\times L}$ the system input at discrete time n, where L represents the window size. The structure of the input and the window is shown in Fig. 2. The input $\boldsymbol{X}_n$ undergoes the following processing steps within the MTAD framework.

1) The transformer encoders generate a latent representation $\boldsymbol{Z}_n$, capturing the underlying patterns and dependencies in the input data.
2) The latent representation $\boldsymbol{Z}_n$ is fed into the convolutional decoder, which produces an estimated version of the system input, denoted as $\hat{\boldsymbol{X}}_n \in \mathbb{R}^{S\times L}$.
3) The prediction network utilizes the latent representation $\boldsymbol{Z}_n$ to generate a one-step-ahead prediction of the measurement vector, denoted as $\check{\boldsymbol{x}}_{n+1} \in \mathbb{R}^{S\times 1}$.
4) Partial anomaly scores for both reconstruction and prediction tasks are computed and combined into the global anomaly score.

The proposed system is trained with reference to the architecture shown in Fig. 3. The design of MTAD is motivated by the concept of denoising autoencoders. However, unlike traditional denoising autoencoders that intentionally add noise to the noiseless input data, MTAD assumes that the sensor measurements inherently contain noise. A denoising block is introduced to build a reference signal from the system input. More specifically, during the training procedure, the sensor measurements are processed via an OT-SVD block to produce

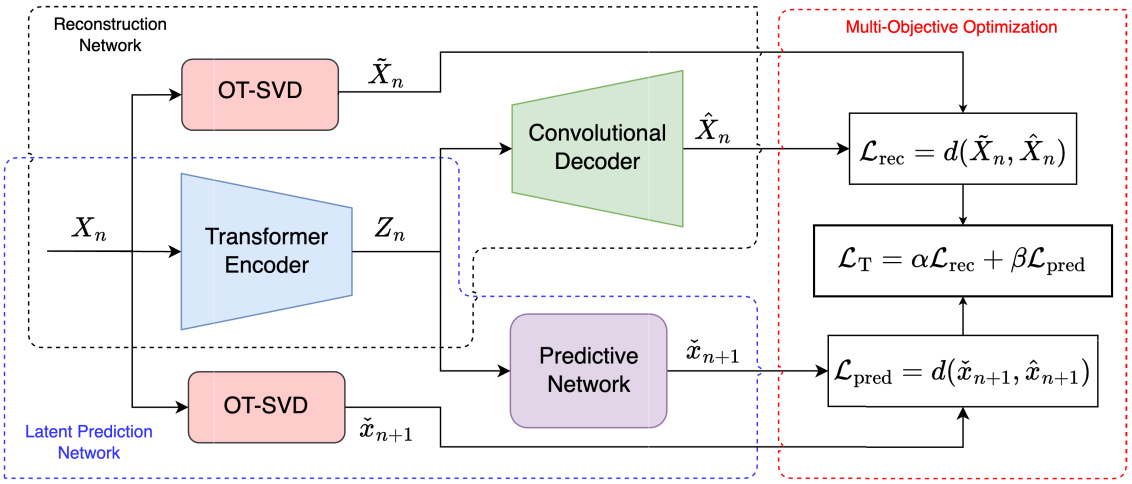


Fig. 3. MTAD training architecture.

reference signals for the reconstruction and prediction tasks, denoted $\tilde{\boldsymbol{X}}_n \in \mathbb{R}^{S\times L}$ and $\tilde{\boldsymbol{x}}_{n+1} \in \mathbb{R}^{S\times 1}$, respectively.

MTAD is basically composed of two networks operating together, and the total loss ($\mathcal{L}$) is calculated as a linear combination of their individual losses ($\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{pred}}$); thus, the proposed system can optimize and trade-off reconstruction accuracy and prediction accuracy, facilitating the detection of different types of anomalies in sensor measurements. The reconstruction loss $\mathcal{L}_{\text{rec}}$ and the prediction loss $\mathcal{L}_{\text{pred}}$ are computed using the reference signals for the reconstruction $\tilde{\mathbf{X}}_n$ and for the prediction $\tilde{\boldsymbol{x}}_{n+1}$, respectively, paired with the corresponding outputs from the convolutional decoder and prediction network. These losses are linearly combined (α and β denote the weights) to improve the model's ability to reconstruct the input data accurately and make reliable predictions. This fusion strategy allows MTAD to inherit the strengths of transformer-based representation learning, denoising techniques, and multinetwork loss computation, thus offering a robust and effective solution for unsupervised anomaly detection in multisensor systems.

C. Description of the Individual Blocks

1) Transformer Encoders: The first block plays a crucial role in the proposed framework by capturing the intricate spatio-temporal patterns in multisensor data. As shown in Fig. 4, it consists of a stack of L_T transformer-encoder modules (we denote $\mathcal{T}(\cdot)$ the mathematical operator representing the individual transformer encoder), each composed of multihead self-attention mechanism, position-wise feed-forward networks, and normalization layers with residual connections. In contrast to recurrent networks, transformer networks process the inputs in parallel, resulting in loss of sequential information, thus positional encoding, via use of a positional encoding matrix $\boldsymbol{P}$ matrix, is required to avoid it [35]. We define the query, key, and value matrices as $\boldsymbol{Q}_i = (\boldsymbol{X}_n + \boldsymbol{P})\boldsymbol{W}_{\text{Q},i}$, $\boldsymbol{K}_i = (\boldsymbol{X}_n + \boldsymbol{P})\boldsymbol{W}_{\text{K},i}$, and $\boldsymbol{V}_i = (\boldsymbol{X}_n + \boldsymbol{P})\boldsymbol{W}_{\text{V},i}$, respectively, where $\boldsymbol{W}_{\text{Q},i} \in \mathbb{R}^{S\times(S/H)}$, $\boldsymbol{W}_{\text{K},i} \in \mathbb{R}^{S\times(S/H)}$, and $\boldsymbol{W}_{\text{V},i} \in \mathbb{R}^{S\times S}$ contain the learned weights for the ith attention head, and H denotes the number of heads. It is worth noticing that the model dimension and the value dimension equal the number of sensors (S). Then,

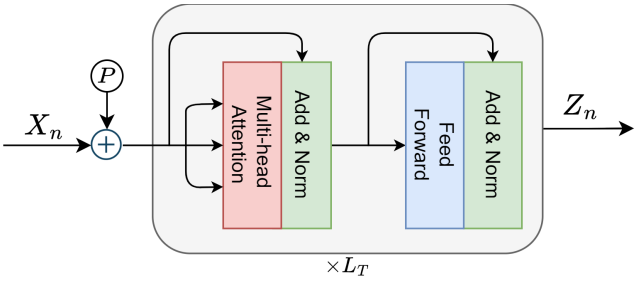


Fig. 4. Transformer encoders.

an attention score ($\boldsymbol{A}_i$) is computed for each head according to the standard procedure based on matrix multiplication, scaling, and column-wise Softmax normalization (details are found in [35]).

In the multihead attention mechanism, multiple scaled-dot attention operations are employed in parallel by splitting the projected vectors into different heads. This enables the model to focus on different positions in the input sequence simultaneously, capturing different aspects of the input sequence: the model learns separate weight matrices for each attention head. The outputs from all attention heads are then concatenated and linearly transformed to give the output of the multihead attention layer

$$\boldsymbol{A} = \text{Concat}(\boldsymbol{A}_1, \ldots, \boldsymbol{A}_H)\boldsymbol{W}_{\text{O}} \tag{1}$$

where $\boldsymbol{W}_{\text{O}} \in \mathbb{R}^{HS\times S}$ is the output projection weight matrix.

The output of the multihead self-attention mechanism is added to the initial input and then processed through the normalization layer. Then, a position-wise feed-forward neural network (FFNN) is applied to capture complex nonlinear behavior and finally an analogous block with input addition plus normalization layer is employed. The normalization layer helps with stability during the learning process by normalizing the inputs across the features instead of across the batch and reducing the co-variate shift problem.

The relation between the input sequence ($\boldsymbol{X}_n$) and the output sequence ($\boldsymbol{Z}_n$) in a block with L_T transformer-encoder modules may be represented as

$$\boldsymbol{Z}_n = \mathcal{T}^{L_T}(\boldsymbol{X}_n + \boldsymbol{P}). \tag{2}$$

2) Convolutional Decoder: The convolutional decoder in the MTAD framework is designed to reconstruct the input sequence from the latent feature representation produced by the stack of transformer-encoder modules. The decoder employs a 1-D convolutional network to perform this reconstruction (we denote $\mathcal{C}(\cdot)$ the mathematical operator representing the 1-D convolutional network). The filters traverse in one direction on the latent matrix ($\boldsymbol{Z}_n$), performing a convolution operation to produce the reconstructed sequence ($\hat{\boldsymbol{X}}_n$), that is,

$$\hat{\boldsymbol{X}}_n = \mathcal{C}(\boldsymbol{Z}_n). \tag{3}$$

Through this process, the decoder utilizes reconstruction error to detect point anomalies from the multisensor measurement.

3) Predictive Network: The predictive network in the MTAD framework is designed to predict the future values of the sensor measurements based on the latent feature representation generated by the stack of transformer-encoder modules. It consists of a feed-forward network with sigmoid activation (we denote $\mathcal{F}(\cdot)$ the mathematical operator representing the feed-forward network). More specifically, the predictive network predicts a vector one time step into the future ($\check{\boldsymbol{x}}[n+1]$) by applying a linear transformation to the latent representation ($\boldsymbol{Z}_n$), that is,

$$\check{\boldsymbol{x}}[n+1] = \mathcal{F}(\boldsymbol{Z}_n). \tag{4}$$

The key capability of the predictive network lies in its ability to detect point anomalies and sub-sequence anomalies (the latter being common in multisensor measurements). This dual detection capability significantly improves the performance. When paired with the reconstruction decoder, the predictive network provides a robust mechanism for unsupervised anomaly detection in multisensor data.

4) OT-SVD: To mitigate the impact of noise on anomaly detection, MTAD employs OT-SVD as a denoising technique during the training stage. OT-SVD improves anomaly detection accuracy by helping the MTAD architecture to distinguish actual anomalies from noise-induced variations in the sensor measurements. OT-SVD relies on classical SVD of the matrix ($\boldsymbol{X}$), that is,

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sum_{i=1}^{\min(S,L)} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T \tag{5}$$

where $\boldsymbol{U} \in \mathbb{R}^{S\times S}$ and $\boldsymbol{V} \in \mathbb{R}^{L\times L}$ are orthogonal matrices, that is, $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}_S$ and $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}_L$, and $\boldsymbol{\Sigma} \in \mathbb{R}^{S\times L}$ is a non-negative diagonal matrix. The columns of $\boldsymbol{U}$, denoted $\boldsymbol{u}_i$, are the left singular vectors of $\boldsymbol{X}$ and correspond to the eigenvectors of $\boldsymbol{X}\boldsymbol{X}^T$. The columns of $\boldsymbol{V}^T$, denoted $\boldsymbol{v}_i^T$, are the right singular vectors of $\boldsymbol{X}$ and correspond to the eigenvectors of $\boldsymbol{X}^T\boldsymbol{X}$. The diagonal elements of $\boldsymbol{\Sigma}$, denoted σ_i, are the singular values and represent the square roots of the eigenvalues of $\boldsymbol{X}^T\boldsymbol{X}$ or $\boldsymbol{X}\boldsymbol{X}^T$, arranged in descending order (i.e., $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(S,L)}$).

In multisensor systems, it is common to observe dependencies among measurements from different sensors (e.g., due to physical proximity, shared environmental conditions, or interactions within the monitored system). Consequently, it is realistic to assume that signals captured by the sensors exhibit low-order structures, which translate into rank deficiency of the input data matrices. As a result, low-rank approximation appears to be an appealing strategy for data representation. We consider the TSVD estimator (also known as partial SVD). According to the Eckart–Young–Mirsky (EYM) theorem [45], the optimal rank-r approximation ($\tilde{\boldsymbol{X}}_{(r)}$) that minimizes the Frobenius norm is obtained by retaining only the first r singular values and their associated singular vectors of the original matrix ($\boldsymbol{X}$), thus

$$\tilde{\boldsymbol{X}}_{(r)} = \underset{\hat{\boldsymbol{X}}:\, \mathrm{rank}(\hat{\boldsymbol{X}})\leq r}{\arg\min} \|\boldsymbol{X} - \hat{\boldsymbol{X}}\|_F^2 \tag{6}$$

$$= \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T = \tilde{\boldsymbol{U}}_{(r)} \tilde{\boldsymbol{\Sigma}}_{(r)} \tilde{\boldsymbol{V}}_{(r)}^T \tag{7}$$

where $\tilde{\boldsymbol{U}}_{(r)}$ and $\tilde{\boldsymbol{V}}_{(r)}$ denote the first r columns of $\boldsymbol{U}$ and $\boldsymbol{V}$, and $\tilde{\boldsymbol{\Sigma}}_{(r)}$ contains the leading $r \times r$ sub-block of $\boldsymbol{\Sigma}$.

Selecting the proper rank (r) is a relevant problem which can be handled with simple approaches such as selecting the elbow on the scree plot of the singular values (in decreasing order) or more advanced methods based on cross-validation techniques or information-theoretic criteria. We consider an information-theoretic approach exploiting random matrix theory [46], [47]. A threshold-dependent rank is introduced

$$r(\tau) = \max\{i : \sigma_i > \tau\} \tag{8}$$

with $\tau > 0$, then the optimal threshold (τ^*) is found by minimizing the asymptotic[1] MSE between the original matrix and the approximation in (7), that is,

$$\tau^* = \arg\min_{\tau} \lim_{S\to\infty} \mathbb{E}\left[\|\boldsymbol{X} - \tilde{\boldsymbol{X}}_{(r)}\|_F^2\right]. \tag{9}$$

Under the assumption that the measurements are affected by additive white Gaussian noise, the optimal threshold (τ^*) is given by

$$\tau^* = \omega(\rho)\sigma_{\mathrm{med}} \tag{10}$$

where σ_{med} is the median singular value, and $\omega(\rho)$ is computed as

$$\omega(\rho) \approx \begin{cases} 2.858, & S = L \\ 0.56\rho^3 - 0.95\rho^2 + 1.82\rho + 1.43, & S \neq L \end{cases} \tag{11}$$

being $\rho = \min\{S, L\}/\max\{S, L\}$.

5) Fusion-Based Detector: MTAD computes the anomaly score as the sum of the prediction error and the reconstruction error. Both errors are measured using the ℓ_2-norm or Euclidean distance. The anomaly score (s_n) at the generic time step for a given test data point ($\boldsymbol{x}[n]$) is given by

$$s_n = \alpha_s \|\boldsymbol{x}[n] - \hat{\boldsymbol{x}}[n]\|_2 + \beta_s \|\boldsymbol{x}[n] - \check{\boldsymbol{x}}[n]\|_2 \tag{12}$$

where α_s and β_s determine the contribution of the individual scores in the testing phase.

[1] With respect to the dimension S, representing here the number of sensors.

D. Loss Functions and Joint Optimization

MTAD employs two networks for loss computation: a reconstruction network and a latent prediction network. These networks are trained in an end-end-to-end mechanism, with the total loss being computed as a linear combination of their individual outputs to simultaneously optimize the accuracy of both reconstruction and prediction, thereby enhancing the ability to detect different types of anomalies in sensor measurements.

The reconstruction network computes the loss as the MSE between the low-rank OT-SVD denoised matrix sequences ($\tilde{\boldsymbol{X}}_n = (\tilde{\boldsymbol{x}}[n], \tilde{\boldsymbol{x}}[n+1], \ldots, \tilde{\boldsymbol{x}}[n+L-1])$) and the reconstructed sequences ($\hat{\boldsymbol{X}}_n = (\hat{\boldsymbol{x}}[n], \hat{\boldsymbol{x}}[n+1], \ldots, \hat{\boldsymbol{x}}[n+L-1])$). The loss function for the reconstruction network for a single sequence is computed as[2]

$$\mathcal{L}_{\text{rec}} = \frac{1}{S}\frac{1}{L}\sum_{k=1}^{S}\sum_{\ell=0}^{L-1}\left(\tilde{x}_k[n-\ell] - \hat{x}_k[n-\ell]\right)^2. \tag{13}$$

The reconstruction network aims at reconstructing accurately normal data samples, with anomalous samples resulting in larger reconstruction errors.

The latent prediction network employs the transformer encoder and the predictive network to predict future values for each sliding window in an autoregressive manner. The loss is calculated as the MSE between the low-rank OT-SVD denoised vector ($\tilde{\boldsymbol{x}}_{n+1}$) and the predicted vector ($\check{\boldsymbol{x}}_{n+1}$) as follows:

$$\mathcal{L}_{\text{pred}} = \frac{1}{S}\sum_{k=1}^{S}\left(\tilde{x}_k[n+1]) - \check{x}_k[n+1]\right)^2. \tag{14}$$

The overall loss function in the MTAD approach linearly combines the reconstruction and prediction losses (with weights α and β controlling the contribution of each loss term)

$$\mathcal{L} = \alpha\mathcal{L}_{\text{rec}} + \beta\mathcal{L}_{\text{pred}}. \tag{15}$$

The training procedure updates the overall loss function via back-propagation using batch size of B. The complete procedure for training the MTAD architecture is summarized in Algorithm 1.

E. Inference and Anomaly Detection

Anomaly detection ($\hat{y}_m$) is done based on a threshold-based rule applied to the anomaly score (s_m), that is,

$$\hat{y}_m = \begin{cases} 1, & s_m > \lambda^* \\ 0, & s_m \leq \lambda^* \end{cases} \tag{16}$$

where the threshold (λ^*) can be selected according to various strategies. The inference procedure for the MTAD architecture is summarized in Algorithm 2.

III. Experimental Setup

In this section, we provide descriptions of the datasets with the related pre-processing methods. Also, we present the evaluation metrics, the baseline methods, the training hyperparameters, and various implementation details.

[2]This computation is performed in total for (N/L) number of sequences from the training data.

Algorithm 1 MTAD Training Algorithm

Input: Normal Training Dataset $\boldsymbol{X} = (\boldsymbol{x}[1], \boldsymbol{x}[2], \ldots, \boldsymbol{x}[N])$, window size (L), number of epochs (e), batch size (M), number of batches (b), number of transformer blocks (L_T), key dimension (d_k), number of heads (h), forward dimension (d_{ff}), hyperparameters $(\alpha, \beta, \epsilon, lr)$

Output: Trained MTAD model parameters (Transformer encoder $(\mathcal{T}_{\text{w}})$, Decoder $(\mathcal{C}_{\text{w}})$, Memory $(\mathcal{F}_{\text{w}})$)

```
Data Pre-processing, resampling, scaling.
T_w, C_w, F_w ← initialize model parameters
k ← 1
repeat
    for j ← 1 to b = N/B do
        Z_n = T(X_n + P)                                   ▷ Eq. 2
        X̂_n = C(Z_n)                                       ▷ Eq. 3
        x̂[n+1] = F(Z_n)                                    ▷ Eq. 4
        U, Σ, V ← SVD(X_n)                                 ▷ Eq. 5
        σ_med ← median(diag(Σ))
        if L ≥ S then
            ρ ← L/S
        else
            ρ ← S/L
        end if
        if L = S then
            τ* ← 2.858 σ_med
        else
            τ* = ω(ρ) σ_med                                ▷ Eq. 10
        end if
        r(τ) ← max{i : σ_i > τ*}                           ▷ Eq. 8
        X̃_n ← Σ_{i=1}^{r} σ_i u_i v_i^T                    ▷ Eq. 7
        L_{j,rec} ← MSE(X̃_n, X̂_n)                          ▷ Eq. 13
        L_{j,pred} ← MSE(x̃[n+1], x̌[n+1])                   ▷ Eq. 14
        L ← α L_{j,rec} + β L_{j,pred}                     ▷ Eq. 15
        T_w, C_w, F_w ← T_w, C_w, F_w − lr∇L
    end for
    k ← k + 1
until k = e
```

A. Datasets

We evaluate the performance of the proposed framework using three publicly available real-world multisensor datasets.

1) *Secure Water Treatment (SWaT) Dataset [48], [49]:* It is collected from a testbed that simulates the physical process and control system of a real-world water treatment system. The dataset features diverse network traffic, sensor, and actuator measurements. It includes 11 days of continuous operation data (seven days of normal operation and four days under both normal and attack scenarios).
2) *Water Distribution (WADI) Dataset [50]:* It is collected from a testbed that expands upon the SWaT system, forming a comprehensive and realistic network for water treatment, storage, and distribution. It includes 16 days of data (14 days of normal operation and two days under attack scenarios).
3) *Server Machine Dataset (SMD) [31]:* It contains server metrics collected from a large Internet company. The

Algorithm 2 MTAD Inference Algorithm

Input: Test dataset containing normal and anomaly data: $\boldsymbol{X} = (\boldsymbol{x}[1], \boldsymbol{x}[2], \ldots, \boldsymbol{x}[M])$, window size (L), True labels: $\boldsymbol{y} = (y_1, y_2, \ldots, y_M)^T$, Threshold (λ^*),

Output: Predicted Labels: $\hat{\boldsymbol{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_M)^T$, F1-score, AUC, AUPR

```
Data Pre-processing, resampling, scaling.
for m ← 1 to M do
    x̂_m ← C(T(X_n))
    x̌_m ← F(T(X_n))
    s_m ← α_s ‖x_m − x̂_m‖_2 + β_s ‖x_m − x̌_m‖_2        ▷ Eq. 12
    if s_m > λ* then
        y_m ← 1
    else
        y_m ← 0
    end if                                              ▷ Eq. 16
end for
```

dataset consists of 38 entities representing different operations of the servers. For our performance analysis, we used two entities (SMD-1 and SMD-2) of the dataset. It includes data from five weeks with a 1-min sampling frequency (both normal operational data and data under various anomaly scenarios).

4) *Soil Moisture Active Passive (SMAP) Satellite Dataset [51]:* It is a labeled MTS dataset from NASA, and it is used for detecting anomalies in soil moisture levels. It comprises 55 entities with 25 monitoring metrics. For our analysis, we considered eight entities from the dataset.

Table I presents a summary of the attributes and statistics of each dataset.

TABLE I
SUMMARY OF DATASETS

Attributes	Datasets				
	SWaT	WADI	SMD-1	SMD-2	SMAP
Entities	1	1	1/28	2/28	8/55
No. of channels	51	123	38	38	25
Average Train size	495,000	1,209,601	25,300	23,693	135183
Average Test size	449,919	172,801	25,301	23,694	427617
Anomaly rate	12.140%	5.99%	4.21%	4.93%	13.13%

B. Data Pre-Processing and Tools

We apply some pre-processing steps as downsampling and feature normalization before the time series are processed according to the proposed framework.

We perform downsampling using a median filter with a 1-min window size and no overlap, as in [34], for both training and test data. Labels for the downsampled test data are built such that a sample is declared anomalous if the corresponding window contains at least one anomaly, otherwise labeled as normal. Downsampling both accelerates the training process of the neural networks and denoises the normal training data.

As for feature normalization, we employed min-max scaling to ensure stable model training

$$x = \frac{\xi - \xi_{\min}}{\xi_{\max} - \xi_{\min}} \tag{17}$$

where ξ (resp. x) represents the actual (resp. scaled) measurement, while $\xi_{\min}$ and $\xi_{\max}$ are the minimum and maximum measured values in the training set, respectively.

We utilized both the PyTorch and TensorFlow deep-learning frameworks for model training and evaluation. Additionally, the scikit-learn machine learning library was used for data pre-processing. All models were trained in the Google Colab Pro environment using NVIDIA T4 Tensor Core GPU processors.

C. Baseline Methods

To evaluate the performance of the proposed method, we selected the following state-of-the-art conventional and deep-learning anomaly detection methods.

1) Isolation forest (IF) [52], that is, an unsupervised anomaly detection algorithm based on decision trees.
2) One-class support vector machines (OC-SVMs) [53], that is, a method for anomaly detection building one hypersphere around normal data points.
3) Multilayer perceptron autoencoder (MLP-AE) [22], that is, a deep-learning method using a feed-forward encoder–decoder neural network to reconstruct normal data and identifying anomalies based on the reconstruction error.
4) Gated recurrent unit (GRU) [54], that is, an RNN learning sequential patterns and detecting anomalies by using the prediction error.
5) Convolutional LSTM (ConvLSTM) [32], [55], that is, a hybrid neural network combining convolutional and LSTM layers to capture spatial and temporal patterns for anomaly detection;
6) UnSupervised anomaly detection (USAD) [34], that is, a method based on adversarially trained autoencoders.
7) DAGMM [33], that is, a deep-learning architecture made of a compression network and an estimation network.
8) Multivariate anomaly detection strategy with GAN (MAD-GAN) [29], that is, an unsupervised method that uses a generative adversarial network (GAN) to learn the underlying normal data distribution and detect anomalies based on the difference between the real data and the generated data (LSTM used as generator and discriminator to handle time-series data).

Moreover, we include the following MTAD variants as baselines to justify the importance of each component in the framework.

1) *MTAD-P:* The reconstruction network is ignored, and only the prediction error is used as an anomaly score, that is, $\alpha_s = \alpha = 0$.
2) *MTAD-R:* The latent prediction network is ignored, and only the reconstruction error is used as an anomaly score, that is, $\beta_s = \beta = 0$.
3) *MTAD-W:* The OT-SVD block is removed, and the reconstruction and latent prediction networks employ the actual noisy matrix as signal reference.

D. Implementation Details

We implement the selected baseline methods using open-source repositories and our own implementations. The

TABLE II
MTAD IMPLEMENTATION DETAILS

Hyperparameters	Datasets				
	SWaT	WADI	SMD-1	SMD-2	SMAP
L_T	1	1	1	2	1
H	2	8	4	8	2

conventional IF and OC-SVM methods are implemented using the open-source PyOD python package [56]. IF employs a 100-tree ensemble as its estimator, with each tree performing splits at a single node using a single feature. For OC-SVM, we utilize a polynomial kernel with a degree of 5. The MLP-AE consists of a three-layer encoder and a three-layer decoder. The input channels are reduced to a 16-D latent space vector, and the model is trained to minimize the mean square loss for 100 epochs. For predictive GRU and ConvLSTM deep-learning models, a look-back of 120 observations is utilized to predict one-step ahead. Both GRU and ConvLSTM architectures comprise three layers of stacked cells, each containing 64 neurons. For ConvLSTM, two sub-sequences of 60× steps are created to apply convolution before feeding the result into LSTM cells. The USAD method is implemented with default hyperparameters. The model architecture comprises one encoder and two decoders, where each module consists of three linear layers with rectified linear unit (ReLU) activation functions in between and a sigmoid activation function for the final layer. The compression network for DAGMM model uses four-layer encoder and decoder with a 10-D latent space and hyperbolic tangent activation function. The estimation network uses a GMM with four mixture components for determining likelihood. MAD-GAN uses an LSTM network with a depth of 3 and 100 hidden units for the generator, and a simpler LSTM network with 100 hidden units and depth 1 for the discriminator. The dimension of the latent space is 15.

For our proposed MTAD framework, we utilized Tensorflow KerasTuner [57] for hyperparameters tuning. The model is trained using the Adam optimizer with a learning rate of 0.01 and a batch size of 64. The number of epochs for training is set based on the convergence of the loss function. During back-propagation, we employ mini-batch gradient descent with the adaptive moment estimation (ADAM) optimizer [58], using a learning rate of 10^{-3}. To prevent overfitting, an early stopping criterion is set using a validation split of 5%. In the hidden layers, we utilize the ReLU as the activation function, while the output layer of reconstruction and prediction network employs a Sigmoid function. MTAD implementation details are given in Table II.

E. Evaluation Metrics

In our performance evaluations, we approach the anomaly detection problem as a binary classification task using labeled test datasets. We study the behavior of the various methods under various common metrics relying on the number of correctly detected anomalies [i.e., true positives (TPs)], the number of erroneously detected anomalies (i.e., FPs or false alarms), the number of correctly identified normal samples [i.e., true negatives (TNs)], and the number of erroneously identified normal samples [i.e., false negatives (FNs)]. More specifically, we define the TP rate (TPR) and the FP rate (FPR) as

$$\text{TPR} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \quad \text{FPR} = \frac{\text{FP}}{\text{FP}+\text{TN}} \tag{18}$$

while precision (P), recall (R), and $F1$-score (F_1) as

$$\text{P} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad \text{R} = \text{TPR}, \quad F_1 = \frac{2\cdot\text{P}\cdot\text{R}}{\text{P}+\text{R}}. \tag{19}$$

The receiver operating characteristic (ROC) and the precision–recall (PR) curves (namely the curves describing TPR-vs.-FPR and P-vs.-R, respectively) offer a complete view for performance comparison of different approaches, while the corresponding area bounded by those curves [namely the area under the ROC curve (AUC) and the area under the PR curve (AUPR)] is considered as a relevant synthetic indicator for performance assessment.

Due to the unbalanced nature of anomaly detection problem (the number of anomalous samples is significantly smaller than the number of normal samples), we specifically consider F_1 and AUPR as preferable performance indicators.

IV. RESULTS AND DISCUSSION/ANALYSIS

A. Overall Performance

The performance analysis of MTAD and selected baselines from state-of-the-art methods, as applied to the considered datasets, is presented in Table III. The values of the $F1$-score correspond to the following choice for the threshold:

$$\lambda^* = \hat{\mu}_s + 3\hat{\sigma}_s \tag{20}$$

where $\hat{\mu}_s = (1/N)\sum_{n=1}^{N} s_n$ and $\hat{\sigma}_s^2 = (1/N)\sum_{n=1}^{N}(s_n - \hat{\mu}_s)^2$ represent the maximum-likelihood estimates of the mean anomaly score and related variance computed from the training set. It is apparent how the proposed MTAD method outperforms all other considered methods, indicating that it best balances precision and recall. This behavior is also confirmed in Fig. 5, which shows the ROC performance of the various considered approaches.

B. Combined Network Performance

We investigate how changing the balance between the reconstruction and the prediction errors (via the weights α and β, respectively) in the MTAD architecture affects its performance. More specifically, performance with different combinations of the weights on the SWaT dataset is presented in Fig. 6. It is worth noticing that these results include also the comparison with the two variants MTAD-P and MTAD-R (corresponding to the combinations ($\alpha = 0,\ \beta = 1$) and ($\alpha = 1,\ \beta = 0$), respectively). Results according to different performance metrics confirm that a proper balance between reconstruction and prediction capabilities provides additional benefits with respect to focusing on reconstruction alone or prediction alone.

TABLE III
RESULTS ON SWAT, WADI, SMD, AND SMAP DATASETS

Methods	SWaT			WADI			SMD-1			SMD-2			SMAP		
	F_1	AUC	$AUPR$	F_1	AUC	$AUPR$	F_1	AUC	$AUPR$	F_1	AUC	$AUPR$	F_1	AUC	$AUPR$
IF	0.3502	0.8426	0.7577	0.0554	0.7080	0.0987	0.1178	0.8379	0.0547	0.3448	0.8439	0.5623	0.5401	0.7216	0.3833
OC-SVM	0.2932	0.8216	0.7358	0.0965	0.7023	0.1554	0.0857	0.8176	0.2617	0.4083	0.8931	0.8278	0.5059	0.6615	0.6645
MLP-AE	0.3120	0.8263	0.7289	0.0994	0.6708	0.0867	0.0102	0.8425	0.4207	0.3046	0.8879	0.7398	0.5037	0.8495	0.7009
GRU	0.3115	0.8312	0.7403	0.0959	0.6701	0.1191	0.0101	0.8676	0.3818	0.3006	0.9129	0.8837	0.5220	0.8500	0.7039
ConvLSTM	0.2849	0.8435	0.7397	0.0886	0.7041	**0.2219**	0.0092	0.9075	0.5396	0.2979	0.9174	0.8840	0.4803	0.8012	0.6740
USAD	0.3256	0.8046	0.7031	0.1103	0.6763	0.1196	0.1036	0.8387	0.3238	0.4709	0.9103	0.8787	0.4542	0.8659	0.7162
DAGMM	0.3253	0.8017	0.6917	0.1569	0.7033	0.1359	0.0857	0.8352	0.3692	0.3854	0.9149	**0.8968**	0.4127	0.8682	0.7194
MAD-GAN	0.3277	0.7205	0.2518	0.0719	0.5899	0.0484	0.0660	0.5448	0.0218	0.0080	0.6760	0.2308	0.4798	0.8701	0.7188
MTAD	**0.6269**	**0.9166**	**0.8555**	**0.1820**	**0.8313**	0.1766	**0.1429**	**0.9735**	**0.5641**	**0.8178**	**0.9429**	0.8806	**0.6307**	**0.8778**	**0.7325**

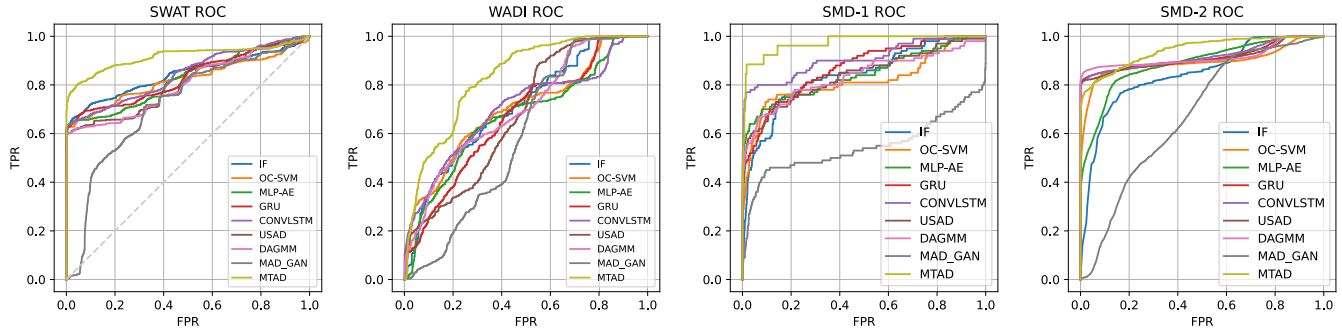


Fig. 5. Performance comparison in terms of ROC curves. (a) SWaT. (b) WADI. (c) SMD-1. (d) SMD-2.

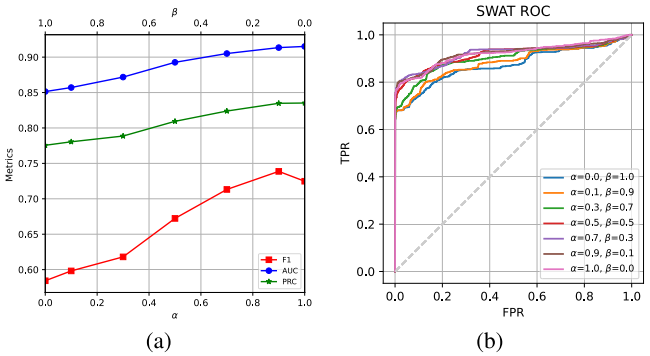


Fig. 6. Performance for different reconstruction and prediction coefficients on the SWaT dataset. (a) *F1*-score, AUC, and AUPR. (b) ROC curves.

C. Parameter Sensitivity Analysis

We examine sensitivity of the MTAD architecture to different parameters: the window size (L), the number of transformer encoders (L_T), and the number of heads (H). The results are shown in Fig. 7, where the default configuration for the fixed parameters is $L = 200$, $L_T = 1$, $H = 2$.

As for the window size, we performed experiments with $L \in \{50, 250\}$ and noticed a general improvement of the performance metrics with the window-size increasing. However, it is also apparent that a large-size window is also paired with a large computational cost.

As for the number of transformer encoders, we performed experiments with $L_T \in \{1, 4\}$ and noticed a non-monotonic relation with the performance metric, indicating that a more-complex model does not necessarily provide beneficial effects and might introduce overfitting.

As for the number of heads, we performed experiments with $H \in \{2, 32\}$ and noticed a non-monotonic behavior of the performance metric, suggesting similar considerations to the case with the number of transformer encoders.

This results suggest that hyperparameter optimization is not trivial for the MTAD architecture and should be carefully addressed. However, this issue is beyond the scope of this article.

D. OT-SVHT Analysis

Fig. 8 shows the scree plots for the application of OT-SVHT to multiple windows with size $L = 200$ on the SWaT dataset.

The estimated rank ranges between 19 and 25 (where the overall number of sensors is 51); thus, although the measurement noise causes the sensor measurements to have full rank, it is apparent that the sensor measurements are highly correlated. The threshold for cutting the singular values ranges between 0.0114 and 0.2799. The variations in the selected rank (or related optimal singular value cutoff point) suggest that different windows experience different noise levels.

E. Ablation Study

The ablation study compares the MTAD full architecture and its variant MTAD-W, which omits the OT-SVD mechanism. The comparison is summarized in Fig. 9, which apparently clarifies the relevance of the denoising approach introduced with the use of the OT-SVD.

F. Computational Complexity

The overall computational complexity of the proposed model is determined by several operations performed across

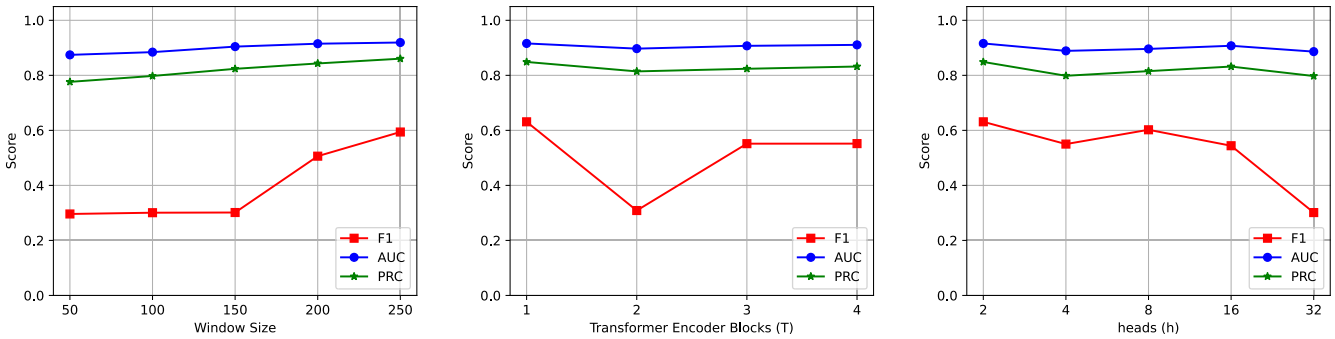


Fig. 7. Parameter sensitivity analysis of MTAD. (a) L. (b) L_T. (c) H.

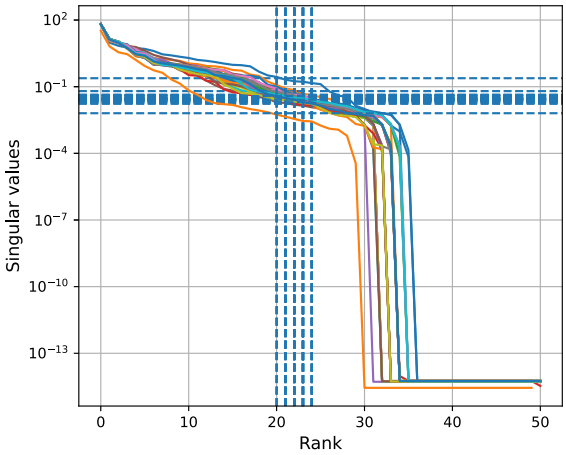


Fig. 8. OT-SVD analysis on the SWaT dataset.

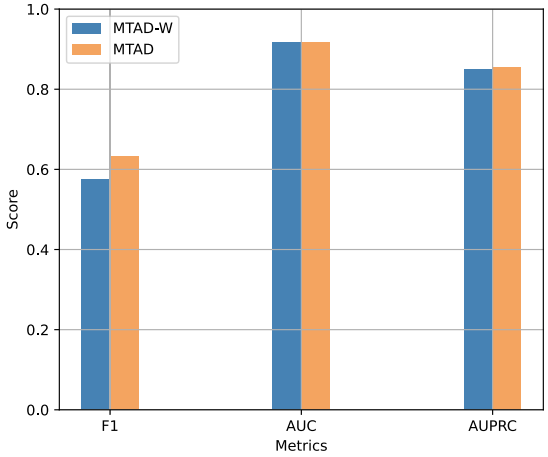


Fig. 9. Comparison of MTAD with its variant MTAD-W.

its modules. The OT-SVD process is performed pre-training, and hence it is not considered for model complexity analysis. The most computationally intensive operations arise from the encoder due to attention mechanisms. The computational complexity of each multihead attention layer is $\mathcal{O}(L^2 \cdot S)$. The FFNN within transformers add additional complexity, typically $\mathcal{O}(L \cdot S^2)$, depending on the inner dimension. The total computational complexity scales with the number of transformer blocks (L_T). The decoder employs 1-D convolutional layers, focusing on reconstructing the input sequence or generating new sequences based on encoded representations. The complexity of convolutional layers is $\mathcal{O}(k \cdot L \cdot S^2)$, where k is the kernel size. The predictor is an FFNN, and its complexity primarily depends on the number of layers, the number of neurons in each layer, and the operations performed at each neuron. The complexity is the product of the number of neurons in the current layer n_l and the number of neurons in the previous layer n_{l-1}, plus the bias term for each neuron in the current layer, that is, $\mathcal{O}(n_{l-1} \cdot n_l + n_l)$. Table IV presents a comparative analysis of training and inference times of the proposed model and baseline models, with MTAD exhibiting a notable reduction in inference and training time in comparison to alternative methods.

TABLE IV
COMPARISON OF TRAINING AND INFERENCE TIMES ON SMAP DATASET

Model	Average Train time (sec)	Average Test time (sec)
IF	0.3591	0.1560
OC-SVM	0.2847	0.4085
MLP-AE	8.6163	0.6625
GRU	27.4611	2.2350
ConvLSTM	23.3036	1.5864
USAD	81.0371	4.4837
DAGMM	69.3916	3.2364
MAD-GAN	80.4999	2.2768
MTAD	24.067	0.9908

More specifically, we stress that the complexity is polynomial both with respect to the number of sensors and the length of the sequence, thus posing no critical issue related to scalability when dealing with large-scale multisensor data in real-world applications. Optimizations such as reducing sequence length, dimensionality reduction techniques, or efficient attention mechanisms can help manage model efficiency and scalability.

V. CONCLUSION

The proposed MTAD architecture is a novel framework designed to effectively detect multiple types of anomalies in sensor measurements. It combines a reconstruction network and a latent prediction network focusing on point anomaly detection and sub-sequence anomaly detection, respectively. Also, MTAD addresses the challenge of noisy sensor measurements by utilizing OT-SVD, while spatio-temporal dependencies between sensors are exploited via a multihead self-attention network. In addition, the MTAD framework

combines multiple anomaly scoring techniques into an end-to-end training procedure. This integration ensures robust anomaly detection by leveraging the strengths of different scoring methods. In general, the proposed MTAD framework combines different advantages of several approaches resulting in a very effective approach for anomaly detection in multivariate time series from sensor measurements.

REFERENCES

[1] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.

[2] H. Nizam, S. Zafar, Z. Lv, F. Wang, and X. Hu, "Real-time deep anomaly detection framework for multivariate time-series data in industrial IoT," *IEEE Sensors J.*, vol. 22, no. 23, pp. 22836–22849, Dec. 2022.

[3] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Secur.*, vol. 28, nos. 1–2, pp. 18–28, 2009.

[4] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 303–336, 1st Quart., 2014.

[5] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," *IEEE Wireless Commun.*, vol. 15, no. 4, pp. 34–40, Aug. 2008.

[6] A. Ukil, S. Bandyoapdhyay, C. Puri, and A. Pal, "IoT healthcare analytics: The importance of anomaly detection," in *Proc. IEEE 30th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, May 2016, pp. 994–997.

[7] F. van Wyk, Y. Wang, A. Khojandi, and N. Masoud, "Real-time sensor anomaly detection and identification in automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 1264–1276, Mar. 2020.

[8] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.

[9] M. A. Belay, S. S. Blakseth, A. Rasheed, and P. Salvo Rossi, "Unsupervised anomaly detection for IoT-based multivariate time series: Existing solutions, performance analysis and future directions," *Sensors*, vol. 23, no. 5, p. 2844, Mar. 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/5/2844

[10] I. Melnyk, B. Matthews, H. Valizadegan, A. Banerjee, and N. Oza, "Vector autoregressive model-based anomaly detection in aviation systems," *J. Aerosp. Inf. Syst.*, vol. 13, no. 4, pp. 161–173, Apr. 2016, doi: 10.2514/1.i010394.

[11] V. D. C. C. de Vargas, L. F. Dias Lopes, and A. Mendonça Souza, "Comparative study of the performance of the CuSum and EWMA control charts," *Comput. Ind. Eng.*, vol. 46, no. 4, pp. 707–724, Jul. 2004.

[12] M. Hu, X. Feng, Z. Ji, K. Yan, and S. Zhou, "A novel computational approach for discord search with local recurrence rates in multivariate time series," *Inf. Sci.*, vol. 477, pp. 220–233, Mar. 2019.

[13] F. Nie, Z. Li, R. Wang, and X. Li, "An effective and efficient algorithm for K-means clustering with new formulation," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3433–3443, Apr. 2023.

[14] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, Portland, OR, USA, Aug. 1996, pp. 226–231.

[15] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104, doi: 10.1145/342009.335388.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986. [Online]. Available: https://www.nature.com/articles/323533a0

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] K. Cho et al., "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[19] Z. Ji, Y. Wang, K. Yan, X. Xie, Y. Xiang, and J. Huang, "A space-embedding strategy for anomaly detection in multivariate time series," *Expert Syst. Appl.*, vol. 206, Nov. 2022, Art. no. 117892.

[20] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, Nov. 1998.

[21] Y. He and J. Zhao, "Temporal convolutional networks for anomaly detection in time series," *J. Phys., Conf. Ser.*, vol. 1213, no. 4, Jun. 2019, Art. no. 042050, doi: 10.1088/1742-6596/1213/4/042050/meta.

[22] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proc. MLSDA 2nd Workshop Mach. Learn. Sensory Data Anal.*, Dec. 2014, pp. 4–11, doi: 10.1145/2689746.2689747.

[23] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 733–742.

[24] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019, doi: 10.1561/2200000056.

[25] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture IE*, vol. 2, pp. 1–18, Dec. 2015.

[26] I. Goodfellow et al., "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 139–144, doi: 10.1145/3422622.

[27] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)*, 2019, pp. 703–716.

[28] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*, May 2021, pp. 4027–4035.

[29] H. Zhao et al., "Multivariate time-series anomaly detection via graph attention network," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2020, pp. 841–850.

[30] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts, "Anomaly detection for time series using VAE-LSTM hybrid model," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Aug. 2020, pp. 4322–4326.

[31] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 2828–2837.

[32] C. Zhang et al., "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 1409–1416. [Online]. Available: www.aaai.org

[33] B. Zong et al., "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–19. [Online]. Available: https://openreview.net/forum?id=BJJLHbb0-

[34] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: UnSupervised anomaly detection on multivariate time series," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 3395–3404, doi: 10.1145/3394486.3403392.

[35] A. Vaswani et al., "Attention is All you Need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6000–6010.

[36] S. Tuli, G. Casale, and N. R. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," *Proc. VLDB Endow.*, vol. 15, no. 6, pp. 1201–1214, Feb. 2022.

[37] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, "Learning graph structures with transformer for multivariate time-series anomaly detection in IoT," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9179–9189, Jun. 2021.

[38] S. Guan, B. Zhao, Z. Dong, M. Gao, and Z. He, "GTAD: Graph and temporal neural network for multivariate time series anomaly detection," *Entropy*, vol. 24, no. 6, p. 759, May 2022, doi: 10.3390/e24060759.

[39] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," 2021, *arXiv:2110.02642*.

[40] B. Wu, C. Fang, Z. Yao, Y. Tu, and Y. Chen, "Decompose auto-transformer time series anomaly detection for network management," *Electronics*, vol. 12, no. 2, p. 354, 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/2/354

[41] L. Xu et al., "TGAN-AD: Transformer-based GAN for anomaly detection of time series data," *Appl. Sci.*, vol. 12, no. 16, p. 8085, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/16/8085

[42] G. Li, Z. Yang, H. Wan, and M. Li, "Anomaly-PTG: A time series data-anomaly-detection transformer framework in multiple scenarios," *Electronics*, vol. 11, no. 23, p. 3955, Nov. 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/23/3955

[43] C. Zhang, T. Zhou, Q. Wen, and L. Sun, "TFAD: A decomposition time series anomaly detection architecture with time-frequency analysis," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 2497–2507, doi: 10.1145/3511808.3557470.
[44] X. Wang, D. Pi, X. Zhang, H. Liu, and C. Guo, "Variational transformer-based anomaly detection approach for multivariate time series," *Measurement*, vol. 191, Mar. 2022, Art. no. 110791.
[45] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, no. 3, pp. 211–218, Sep. 1936, doi: 10.1007/bf02288367.
[46] M. Gavish and D. L. Donoho, "Optimal shrinkage of singular values," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2137–2152, Apr. 2017.
[47] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is $4/\sqrt{3}$," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 5040–5053, Aug. 2014.
[48] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.*, 2017, pp. 88–99.
[49] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *Proc. Int. Workshop Cyber-Phys. Syst. Smart Water Netw. (CySWater)*, Apr. 2016, pp. 31–36.
[50] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "WADI: A water distribution testbed for research in the design of secure cyber physical systems," in *Proc. 3rd Int. Workshop Cyber-Phys. Syst. Smart Water Netw.*, Apr. 2017, pp. 25–28, doi: 10.1145/3055366.3055375.
[51] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 387–395.
[52] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discovery From Data*, vol. 6, no. 1, pp. 1–39, Mar. 2012, doi: 10.1145/2133360.2133363.
[53] K.-L. Li, H.-K. Huang, S.-F. Tian, and W. Xu, "Improving one-class SVM for anomaly detection," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2003, pp. 3077–3081.
[54] Z. Qu, L. Su, X. Wang, S. Zheng, X. Song, and X. Song, "A unsupervised learning method of anomaly detection using GRU," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Jan. 2018, pp. 685–688.
[55] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 802–810.
[56] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A python toolbox for scalable outlier detection," *J. Mach. Learn. Res.*, vol. 20, no. 96, pp. 1–7, 2019. [Online]. Available: http://jmlr.org/papers/v20/19-011.html
[57] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, and L. Invernizzi. (2019). *KerasTuner*. [Online]. Available: https://github.com/keras-team/keras-tuner
[58] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Mohammed Ayalew Belay (Graduate Student Member, IEEE) received the B.Sc. degree in physics from Addis Ababa University, Addis Ababa, Ethiopia, in 2011, the M.Sc. degree in physics from Bahir Dar University, Bahir Dar, Ethiopia, in 2013, and the M.Sc. degree in computer science from Hawassa University, Hawassa, Ethiopia, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

His research interests include deep learning, unsupervised learning, anomaly detection, time-series analysis, and physics-informed neural networks.

Adil Rasheed is a Professor with the Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway. There, he works to advance the development of novel hybrid methods that combine big data, physics-driven modeling, and data-driven modeling in the context of real-time automation and control. In addition, he also holds a part-time Senior Scientist Position with the Department of Mathematics and Cybernetics, SINTEF Digital, Trondheim, Norway, where he previously served as the Leader of the Computational Sciences and Engineering Group from 2012 to 2018. His contributions in these roles have been the development and advancement of both the hybrid analysis and modeling and big data cybernetics concepts. Over the course of his career, he has been the driving force behind numerous projects focused on different aspects of digital twin technology, ranging from autonomous ships to wind energy, aquaculture, drones, business processes, and indoor farming. He is currently leading the digital twin and asset management related work in the FME Northwind Centre.

Pierluigi Salvo Rossi (Senior Member, IEEE) was born in Naples, Italy, in 1977. He received the Dr.Eng. (summa cum laude) degree in telecommunications engineering and the Ph.D. degree in computer engineering from the University of Naples "Federico II," Naples, in 2002 and 2005, respectively.

He is currently a Full Professor and the Deputy Head with the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway. He is also a part-time Research Scientist with the Department of Gas Technology, SINTEF Energy Research, Trondheim. Previously, he worked with the University of Naples "Federico II;" the Second University of Naples, Naples; NTNU; and Kongsberg Digital AS, Horten, Norway. He held visiting appointments with Drexel University, Philadelphia, PA, USA; Lund University, Lund, Sweden; NTNU; and Uppsala University, Uppsala, Sweden. His research interests fall within the areas of communication theory, data fusion, machine learning, and signal processing.

Prof. Salvo Rossi was awarded as an Exemplary Senior Editor of IEEE COMMUNICATIONS LETTERS in 2018. He is (or has been) on the Editorial Board of IEEE SENSORS JOURNAL, the IEEE Open Journal of the Communications Society, IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, IEEE COMMUNICATIONS LETTERS, and IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.