Measurement 283 (2026) 122144

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

TideMamba: a hybrid time-series decomposition and Mamba-based architecture for multivariate anomaly detection

Mingwei Feng [a,b], Xuping Zhang [a,b], Kangjian Di [c], Zikang Zeng [c], Yiru Wang [c], Bohan Wang [c], Tianrui Zhang [c], Yali Zhang [c], Yixin Zhang [a,b], Ningmu Zou [c,d,*]

[a] College of Engineering and Applied Sciences, Nanjing University, Nanjing 210023, China
[b] Key Laboratory of Intelligent Optical Sensing and Manipulation, Ministry of Education, Nanjing University, Nanjing 210093, China
[c] School of Integrated Circuits, Nanjing University, Suzhou 215163, China
[d] Interdisciplinary Research Center for Future Intelligent Chips (Chip-X), Nanjing University, Suzhou 215163, China

ARTICLE INFO

Keywords:
Semiconductor manufacturing process
Time-series anomaly detection
Mamba SSM
Trend decomposition
Sparse attention
Adaptive Gating Mechanism
Multi-sensor data

ABSTRACT

Anomaly detection in multivariate time series data is crucial for industrial processes such as semiconductor manufacturing, as early fault identification and timely warnings can improve wafer yield and safety. However, accurately distinguishing anomalies from complex time patterns under non-stationary conditions remains challenging. This paper proposes TideMamba, a novel dual-branch autoencoder architecture that integrates a selective state-space model (SSM) branch with an attention-based branch through an adaptive gating mechanism. By leveraging the long-range dependency modelling capabilities of the selective SSM and the dynamic focusing of the attention mechanism, TideMamba reconstructs normal multi-sensor behaviour while remaining sensitive to subtle anomalies. A preprocessing decomposition step removes trend and periodic components from sensor signals so that the model can focus on residual fluctuations where anomalies are more likely to appear. Experiments on a single real industrial Chemical Vapor Deposition (CVD) semiconductor dataset show that TideMamba achieves a strong precision-recall trade-off compared with representative baselines, including OmniAnomaly, InterFusion, AnomalyTransformer, and classical detectors. On this dataset, TideMamba obtains an Area Under the Receiver Operating Characteristic Curve (AUC-ROC) of 0.989 and a Precision-Recall Area Under the Curve (PR-AUC) of 0.791 while maintaining low inference time. These results support TideMamba as a practical and efficient framework for high-dimensional anomaly monitoring in the studied CVD setting. Broader generalization beyond this single industrial data source should be established through future cross-dataset validation.

1. Introduction

Anomaly detection in multivariate time-series is of central importance in modern industrial systems, particularly in semiconductor manufacturing, where subtle deviations in sensor behaviour may indicate equipment faults, process instabilities, or yield-impacting events [1]. Semiconductor wafer processes generate high-dimensional sensor streams with complex temporal dynamics and strong inter-sensor dependencies [2,3]. During each process cycle, numerous variables such as pressure, gas flow, temperature, and power are continuously monitored, producing data that are nonlinear, noisy, and often non-stationary [4]. These characteristics make reliable anomaly detection substantially more challenging than in conventional low-dimensional monitoring settings [5]. Fig. 1 shows the semiconductor manufacturing process flow and typical manufacturing anomalies, which highlight the importance of early anomaly detection. The abbreviation rules are defined in Table 1.

Traditional methods, including statistical thresholding, clustering, one-class classification, and principal component analysis (PCA) [6], have long been applied to industrial anomaly detection. One-class support vector machines [7] and Isolation Forests [8] seek to define the boundary of normal operating behaviour, whereas PCA-based methods detect deviations in low-variance subspaces [6]. Although useful in

* Corresponding author at: School of Integrated Circuits, Nanjing University, Interdisciplinary Research Center for Future Intelligent Chips (Chip-X), Nanjing University, Suzhou 215163, China.

E-mail addresses: mingweifeng@smail.nju.edu.cn (M. Feng), xpzhang@nju.edu.cn (X. Zhang), kangjiandi@smail.nju.edu.cn (K. Di), 502024730015@smail.nju.edu.cn (Z. Zeng), yiruwang@smail.nju.edu.cn (Y. Wang), bohanwang@smail.nju.edu.cn (B. Wang), 602025730020@smail.nju.edu.cn (T. Zhang), yl_zhang@smail.nju.edu.cn (Y. Zhang), zyixin@nju.edu.cn (Y. Zhang), nzou@nju.edu.cn (N. Zou).

https://doi.org/10.1016/j.measurement.2026.122144
Received 16 December 2025; Received in revised form 13 May 2026; Accepted 3 June 2026
Available online 4 June 2026
0263-2241/© 2026 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

relatively simple settings, such approaches often fail to capture complex nonlinear relationships, long-range temporal dependencies, and evolving correlations among variables. To address these limitations, deep learning methods have attracted increasing attention. Recurrent neural network (RNN) variants, such as LSTM-based encoder-decoder models and variational autoencoders (VAEs) [9,10], have shown strong potential by learning the normal reconstruction patterns of time-series and using reconstruction errors as anomaly scores. Representative studies include the LSTM encoder-decoder framework [10] of Malhotra et al. and OmniAnomaly [11,12], which combines stochastic latent variables with GRU units for unsupervised anomaly detection. More recent models, such as InterFusion [13] and MTAD-GAT [14], have further improved performance by jointly modelling temporal behaviour and inter-variable dependencies. Despite these advances, several challenges remain unresolved. An effective industrial anomaly detection framework should simultaneously capture long-term temporal patterns, exploit cross-sensor relationships, and maintain computational efficiency for long sequences and high-frequency data streams [15,16]. Transformer-based architectures have demonstrated strong representation capacity through attention mechanisms [3,17], yet their quadratic computational complexity with respect to sequence length may limit practical deployment in real-time industrial scenarios [4]. Hence, there is a clear need for a unified architecture that combines efficient long-sequence modelling with adaptive representation of inter-sensor dependencies.

This study focuses on anomaly detection in the chemical vapor deposition (CVD) stage of semiconductor manufacturing, where equipment or control failures can directly cause wafer defects. Each CVD run generates a high-dimensional multivariate time series: the uploaded raw file contains 1,596 recorded sensor/status variables, which are reduced to roughly 100 representative features after preprocessing, including chamber pressure, gas flow, temperature, and RF power. Typical anomalies include pressure spikes from valve failures, RF power interruptions caused by generator faults, and abnormal gas-flow patterns linked to mass-flow-controller errors. Although such faults are rare, their economic impact can be substantial. A single undetected failure may compromise an entire wafer lot, while small yield improvements can be economically meaningful in high-volume production [2,18]. At the industry level, the global defect inspection market exceeded USD 5 billion in 2024 and is projected to grow further by 2033, highlighting the increasing value of robust defect monitoring [19]. To address these challenges within the CVD monitoring setting studied here, we propose TideMamba, a hybrid architecture for unsupervised multivariate anomaly detection that integrates time-series decomposition with a Mamba-based selective state-space autoencoder. The proposed design is motivated by the complementary strengths of two model families. Selective SSMs provide efficient long-sequence modelling with linear time complexity and a strong capacity for capturing latent temporal dynamics [20]. In contrast, attention mechanisms are effective in identifying adaptive dependencies across variables and time steps [11,21]. TideMamba combines these two components in a dual-branch framework composed of an SSM branch and a gated self-attention branch. An adaptive gating unit dynamically balances temporal memory and contextual dependency modelling, allowing the framework to respond to different anomaly patterns. In addition, TideMamba incorporates a time-series decomposition step prior to representation learning. Industrial sensor data frequently contain slowly varying trends and periodic operational patterns that reflect normal process behaviour rather than faults, such as valve start-stop phases or recipe-driven oscillations. Modelling raw sequences directly may therefore blur the distinction between meaningful anomalies and benign drift. By separating trend and seasonal components and focusing the model on residual fluctuations, the proposed framework improves anomaly sensitivity while reducing interference from predictable structure. Fig. 2 shows the problem motivation and background of this paper.

The rest of the paper is organized as follows. Section 2 details the TideMamba architecture and methodology, including decomposition, dual-branch reconstruction, adaptive fusion, scoring, and thresholding. Section 3 presents the experimental results, including baseline comparisons, case studies, component-isolation ablations, and hyperparameter sensitivity. Section 4 discusses the findings, limitations, interpretability evidence, and deployment considerations. Section 5 concludes the paper and outlines future work.

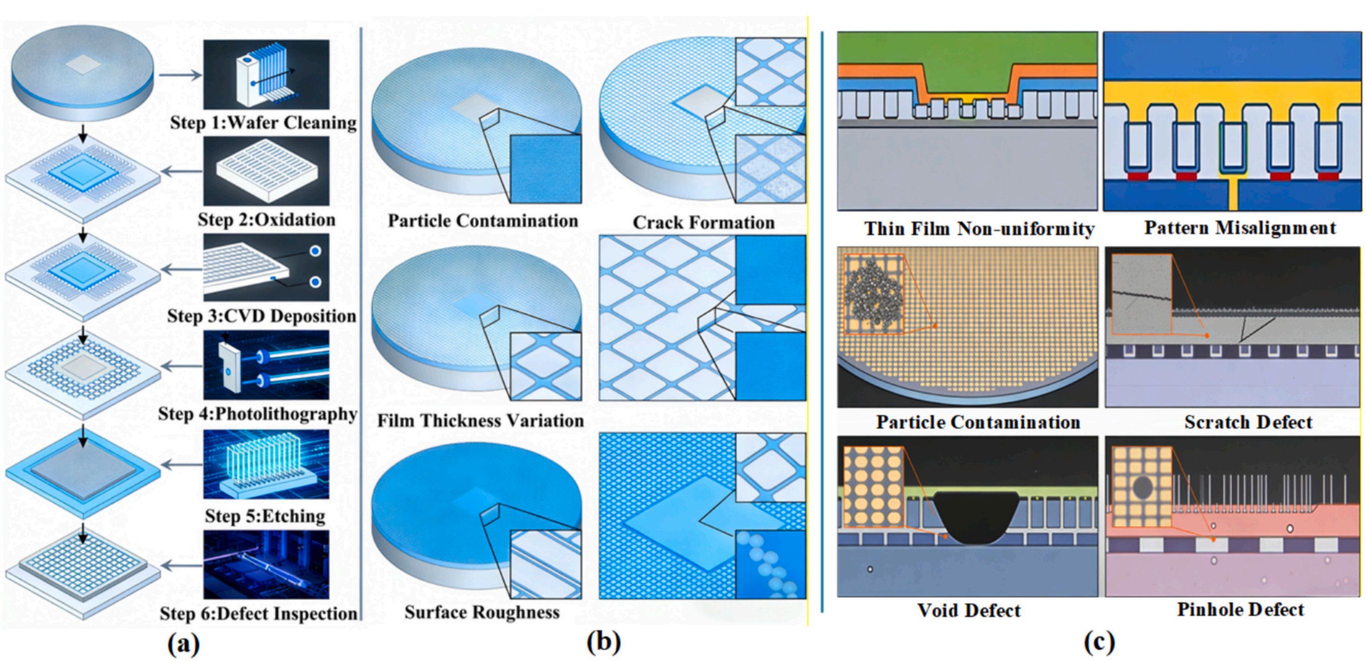

Fig. 1. Semiconductor manufacturing process flow and representative manufacturing anomalies. (a) The main stages of semiconductor manufacturing process. (b) and (c) Representative types of anomalies encountered during the semiconductor manufacturing process, including coating non-uniformity, contamination, misalignment, scratches, voids, and defects at the wafer and device levels caused by other processes.

2. Methods

TideMamba is an unsupervised, two-branch time-series autoencoder designed to model normal multivariate sensor behavior and detect anomalies through reconstruction error. Fig. 3 shows the architecture of the TideMamba model. It consists of three main components: (i) a preprocessing decomposition module for trend and seasonality removal; (ii) a Mamba autoencoder with parallel state-space and attention branches; and (iii) an adaptive gating and output module for generating the final reconstruction and anomaly score. The model takes as input the multivariate time-series data from the CVD process, namely a sequence of sensor readings at each time step across all retained sensors. In this study, the input is a vector x_t in $\mathbb{R}^D$ of D sensor values at time t, with D approximately equal to 100 after preprocessing. The output is a reconstructed sequence x_t, from which an anomaly score is derived. Different anomaly types, including pressure faults and RF power faults, cause different subsets of sensors to exhibit large reconstruction errors, which the detection logic flags as described in Section 2.6.

2.1. Trend and periodicity decomposition

Raw sensor signals from wafer semiconductor manufacturing processes often exhibit global trends (e.g., gradual drift due to sensor aging or cavity contamination, or artificial valve start-stop or warm-up phases) and periodic patterns (e.g., cyclic control oscillations or recipe-induced phases), which are not anomalous in themselves [2,3]. Incorporating these components into modelling may confuse the learning of normal and anomalous behaviour. Thus, TideMamba first performs a time-series decomposition to remove such components. For each sensor i (out of N sensors) at time t, we represent the measured value $x_i(t)$ as a sum of trend $T_i(t)$, periodic (seasonal) component $P_i(t)$, and residual $R_i(t)$ [22,23]:

$$x_i(t) = T_i(t) + P_i(t) + R_i(t), i = 1, 2, \cdots, N\# \quad (1)$$

The trend $T_i(t)$ can be obtained by a smoothing filter or moving average over a long window, and $P_i(t)$ can be estimated via seasonal decomposition (e.g., using domain knowledge of the process cycle period or frequency analysis) [24,25]. The residual $R_i(t)$ contains the high-frequency and irregular patterns—this is the signal component where anomalies are most likely to manifest. TideMamba retains only the residual $R_i(t)$ as the input for the subsequent autoencoder, effectively normalizing each time-series to be zero-mean and stationary. Let $\mathrm{x}(t) = [x_1(t), \cdots, x_N(t)]^\top$ be the vector of all sensor readings at time t, and similarly define $\mathrm{R}(t)$ for residuals. The decomposition can be expressed in vector form as [26,27]:

$$R(t) = x(t) - \widehat{T}(t) - \widehat{P}(t)\# \quad (2)$$

where $\widehat{T}(t)$ and $\widehat{P}(t)$ are estimates of the trend and periodic components. By subtracting these estimates, the detrended and seasonally adjusted series $R(t)$ is obtained. This preprocessing allows the model to focus more on sensor fluctuations that deviate from the expected baseline. In fact, this step significantly improves anomaly detection because the autoencoder no longer needs to expend a lot of energy modelling slow drifts or known periodic oscillations (which are common in CVD processes). Instead, it can focus on learning the typical shape of the residual pattern and detecting any irregular perturbations within it [3,28].

Additionally, we carefully choose the decomposition parameters so as not to remove any potential anomaly signatures: the trend T_t is computed with a window length (on the order of 1000 s in our experiments) much longer than the duration of any expected anomaly, ensuring that sudden anomalous shifts are not absorbed into the trend. The periodic component P_t is estimated based on the known recipe cycle (~60 s in our case) and removed only if a clear repetitive pattern is present. Mathematically, we implement the decomposition by a combination of median filtering and polynomial fitting for trend, and either a fixed-frequency removal or high-pass filter for seasonality. Specifically, for each sensor j, we compute $T_{i,t}$ as a low-frequency component via a rolling median (window ~ 201 points) followed by a smooth polynomial fit (degree 3) to capture any curvature. The residual after removing this trend is then high-pass filtered (using, e.g., a discrete Hankel or HP filter [16] if available) to eliminate any dominant periodicity if known. This yields $R_{i,t}$. (If no strong periodic pattern is known, we simply use $R_{i,t} = x_{i,t} - T_{i,t}$.) in Equation (2) is applied to all sensors. In our dataset, for example, one sensor had a ~ 60 s periodic injection cycle; we removed that known oscillation from P_t. By design, any anomaly that does not strictly follow the global trend or exact known periodic cycle will remain in R_t. The effectiveness of this step is evidenced by improved detection metrics when it is used (see ablation results in Section 4).

2.2. Mamba-gated dual-branch autoencoder

The core of TideMamba is a dual-branch autoencoder that integrates a SSM branch and an attention branch through an adaptive gating mechanism. This design leverages two complementary sequence-modelling paradigms: structured SSM for efficient long-sequence processing, and self-attention for flexible dependency modelling. The architecture draws inspiration from the Mamba SSM framework [16,29], which achieves linear time complexity over sequence length while

Table 1
Nomenclature.

Symbol/ Abbreviation	Description	Symbol/ Abbreviation	Description	Symbol/ Abbreviation	Description
CVD	Chemical Vapor Deposition	RNN	Recurrent Neural Network	ϕ_t	Detrended residual component at time t ($X_t - \tau_t$)
ROC	Receiver Operating Characteristic	CNN	Convolutional Neural Network	$\tilde{X}_t$	Detrended input at time t
AUC	Area Under the Curve	LSTM	Long Short-Term Memory	a_t(or z_t)	Latent state vector in the SSM branch at time t
AUC-ROC	Area Under the Receiver Operating Characteristic Curve	TPR	True Positive Rate (Recall)	b_t	Hidden representation vector from the attention branch at time t
PR-AUC	Precision-Recall Area Under the Curve	FPR	False Positive Rate	G_t	Gating weight (or vector) at time t for combining branches
SSM	State-Space Model	G-mean	Geometric mean of TPR and (1 – FPR)	M	State-space model latent dimension
MSE	Mean Squared Error	GRU	Gated Recurrent Unit (a type of RNN cell)	T	Length of input time-series window
PCA	Principal Component Analysis	X_t	Multivariate sensor reading vector at time t	d	Number of sensor channels (dimensions)
VAE	Variational Autoencoder	τ_t	Estimated trend component at time t	w	Sliding window length (time steps)

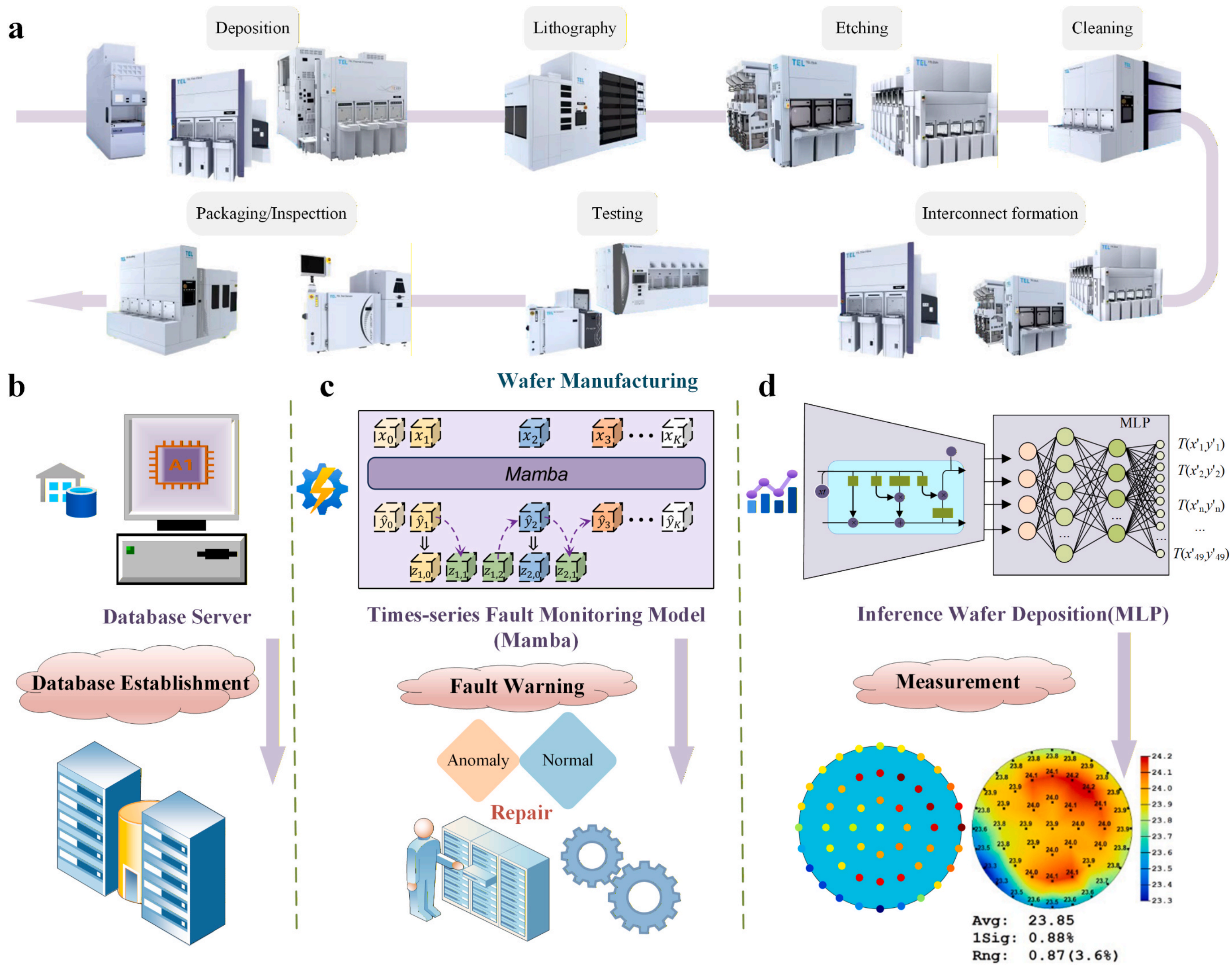

Fig. 2. Motivation and background. (a) This part illustrates how multiple sensors in the edge side CVD machine generate time-series data in the wafer manufacturing process (foundation). (b) Establishment of cloud based industrial process database. (c) Time series fault diagnosis and professional maintenance based on TideMamba (this work). (d) Prediction of backend film thickness and multi parameter material growth based on MLP (future work).

maintaining competitive modelling capacity. We adapt Mamba into an autoencoder by encoding the residual sequence $R(t)$ into latent representations and decoding these representations back to reconstructed inputs. A parallel attention path is then added to capture local and cross-sensor contextual relationships that may not be represented fully by the SSM state alone.

The first branch is a selective state-space encoder that processes the input sequence through a latent state updated by a state-space equation [16]. At each time t, the branch maintains an M-dimensional hidden state $h_t^{(s)} \in \mathbb{R}^M$. The discrete state dynamics are defined by a state transition matrix $A \in \mathbb{R}^{M\times M}$ and input matrix $B \in \mathbb{R}^{M\times N}$ that operate on the residual input $R(t)$, with the initial state $h_1^{(s)} = 0$ (or learned as a parameter). This formulation is analogous to a classic linear time-invariant SSM [16,30]:

$$h_{t+1}^{(s)} = Ah_t^{(s)} + BR(t), t = 1, 2, \cdots, T-1\# \tag{3}$$

where A captures how the latent process evolves and B determines how new sensor inputs drive state changes. We also define an output projection $C \in \mathbb{R}^{N\times M}$ to map the hidden state to an output vector (of the same dimension as a single input sample), which can be interpreted as the SSM branch's encoded representation of the system at time t. Equation (3) and Equation (4) correspond to the encoder part of the state-space branch [29,31]:

$$z_t^{(s)} = Ch_t^{(s)}\# \tag{4}$$

Intuitively, as new sensor readings arrive, the state h_t integrates information over time (much like the hidden state of an RNN, but here with a principled linear state evolution), enabling it to carry long-term context without the vanishing/exploding gradient issues of naive RNNs [32]. In Mamba, the matrices (A, B, C) are not fixed; they are learned during training (with structured parameterizations to ensure efficiency and stability [33]). Furthermore, selectivity in Mamba refers to making state updates partially dependent on the data through gating or other mechanisms [34]. In our implementation, this means that the effective state update can be modulated by learned patterns, giving the model nonlinear expressiveness while maintaining linear time complexity per step. The Mamba branch essentially serves as a powerful backbone for long sequence modelling – it can capture gradual drifts, long oscillatory modes, and other temporal patterns that span hundreds of time steps, all in an efficient manner. Our TideMamba encoder uses a variant of Mamba with M (state dimension) set to 64 (selected based on a balance of accuracy and model complexity. We also include techniques like normalization to ensure numerical stability for long sequences. The

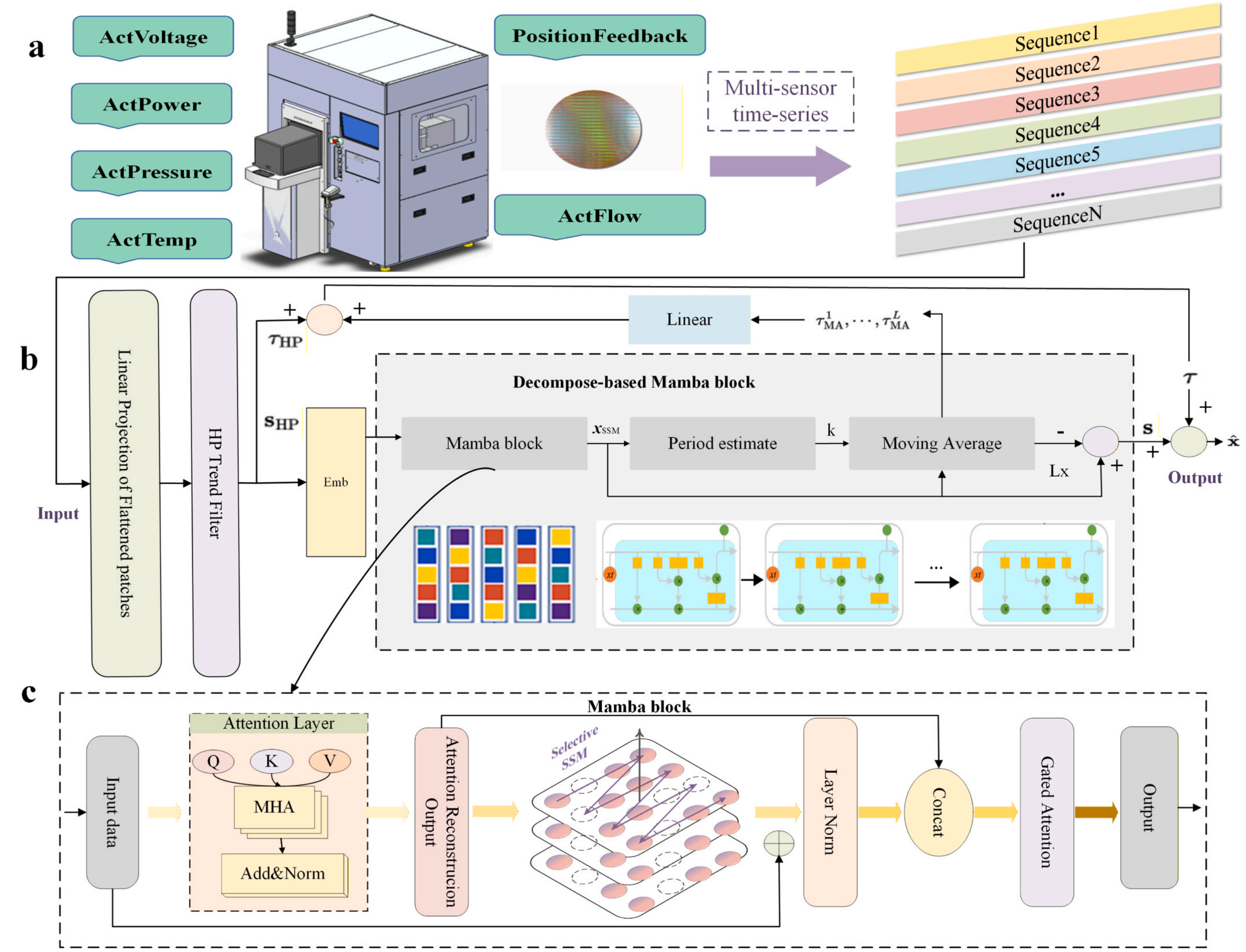

Fig. 3. TideMamba model architecture. (a) Time series obtained from multiple CVD sensors, including pressure, temperature, power, voltage, airflow, and status-related signals. (b) Each multivariate time series is decomposed into trend, periodic, and residual components using a trend filter. The residual, detrended and deseasonalized sequence is fed into a dual-branch autoencoder. One branch is the selective SSM branch, which maintains a latent state that evolves through efficient state transitions and captures long-term temporal dynamics. (c) The other branch is the attention branch, which applies a gated self-attention mechanism to capture inter-sensor relationships and contextual patterns. The adaptive gating unit computes context-dependent fusion weights to merge the branch outputs. The autoencoder is trained to reconstruct normal time-series patterns from the fused representation; high reconstruction errors indicate potential anomalies.

output $z_t^{(\mathrm{SSM})}$ provides one candidate reconstruction of the input at time t (when passed through the decoder, it should approximate R_t if the state-space branch fully explains the data).

2.3. Attention branch

The second branch is an attention-based encoder that operates in parallel to capture rich dependencies across sensors and time. We employ a self-attention mechanism akin to a simplified Transformer encoder operating on the sequence of residuals $R(1), R(2), \cdots, R(T)$ [14]. At a high level, the attention branch produces its own representation $z_t^{(a)}$ by relating each time step t to other time steps through learned similarity weights. For example, one form of single-head scaled dot-product attention can be written as [17,29]:

$$z_t^{(a)} = \sum_{\tau=1}^{T} \alpha_{t,\tau}\left(W_V \mathrm{R}(\tau)\right)\# \tag{5a}$$

$$\alpha_{t,\tau} = \frac{\exp\left(\left(W_Q R(t)\right)^\top \left(W_K R(\tau)\right)\right)}{\sum_{j=1}^{T} \exp\left(\left(W_Q R(t)\right)^\top \left(W_K R(j)\right)\right)}\# \tag{5b}$$

where W_Q, W_K, W_V are learned projection matrices for queries, keys, and values (dimensions omitted for brevity), and $\alpha_{t,\tau}$ is the attention weight that the t-th time step assigns to time step τ. This mechanism allows the model to adaptively focus on certain past sensor readings when encoding the current time step. For instance, if an anomaly at time t involves a combination of sensor spikes that mirror a prior pattern at time τ, the attention branch can directly connect t and τ (via a large weight $\alpha_{t,\tau}$) [32], bringing the relevant context into $z_t^{(a)}$. In contrast, a purely sequential model might struggle to retain that information if $|t-\tau|$ is large. We note that our attention branch includes gating in its internal layers (analogous to transformer feed-forward block gating [33,35]), which further helps modulate feature flow. The result $z_t^{(a)}$ can be seen as an alternative encoding of the input sequence: one that excels at capturing associations across sensors and time (including short-lived but significant correlations) rather than the persistent latent state. We note that to keep complexity manageable, our attention mechanism is not a full O (T^2) global attention on long sequences; rather, it is more localized and cross-channel, which is O (D^2) per time step (with $D \sim 100$, this is negligible).

2.4. Adaptive gating and fusion

A distinguishing feature of TideMamba is the adaptive gating unit that merges the SSM and attention branch outputs. Rather than simply concatenating or averaging the two representations, we compute a data-driven gating vector $g_t \in [0,1]^N$ at each time t (with dimension N, the sensor dimension, to gate each output feature) which determines the blend of the two branches for each sensor's reconstructed output [36]. The gating vector is produced by a small neural network that takes as input the current state or features (we use the concatenation of branch outputs) and outputs a sigmoid activation:

$$g_t = \sigma\left(W_g\left[z_t^{(s)}\cdot;\cdot z_t^{(a)}\right] + b_g\right)\# \tag{6}$$

where $[\cdot;\cdot]$ denotes vector concatenation, W_g and b_g are learnable parameters, and $\sigma(\cdot)$ is the elementwise sigmoid function (ensuring each $g_{t,i} \in (0,1)$). This gating vector then controls the fusion of the branch outputs:

$$z_t = g_t \odot z_t^{(s)} + (1 - g_t) \odot z_t^{(a)}\# \tag{7}$$

where $\odot$ denotes elementwise (Hadamard) product and 1 is an N-dimensional vector of ones. In essence, for each sensor (each component of z_t), the gating mechanism decides how much weight to give to the SSM branch's opinion vs. the attention branch's opinion [17]. If g is close to 1, the model trusts the state-space prediction for sensor i at time t; if it is near 0, the model relies more on the attention-based context for that sensor; intermediate values blend the two. This adaptive gating is particularly powerful in anomaly detection: under normal conditions, the state-space branch (encoding learned normal dynamics) may be sufficient, but when anomaly patterns occur and the state model cannot explain them, the gating can shift weights to the attention branch, which may capture the anomaly as a correlation or contextual bias [17]. In effect, the gating acts as an anomaly-sensitive mechanism—when an anomaly occurs, the contributions of the two branches conflict (the SSM branch, trained on normal data, may produce larger errors, while the attention branch can focus on local biases and label anomalies), and the gating network adjusts to highlight divergent signals.

2.5. Decoder and reconstruction

The autoencoder's decoder takes the fused representation z_t and maps it back to the original sensor space, producing a reconstruction $\widehat{x}(t)$ of the input $x(t)$. In TideMamba, we implement the decoder as a linear projection (essentially the transpose of the encoder's output projection) possibly followed by a non-linear activation. For simplicity, it can be written as [25]:

$$\widehat{x}(t) = W_o z_t + b_o\# \tag{8}$$

where $W_o \in \mathbb{R}^{N\times N}$ and $b_o \in \mathbb{R}^N$. In practice W_o can be tied to C from Equation (4) for parameter efficiency (since z_t lives in the same space as $x(t)$) [8]. The decoder operates at each time step, yielding a reconstructed sequence $\widehat{x}(1), \cdots, \widehat{x}(T)$. Training the autoencoder involves minimizing the reconstruction error between $\widehat{x}(t)$ and the original input $x(t)$ (or equivalently $R(t)$ since the trend and periodic parts are removed). We use a mean squared error (MSE) loss aggregated over all sensors and time steps [26]:

$$Loss_{\text{recon}} = \frac{1}{TN}\sum_{t-1}\|\widehat{x}(t) - x(t)\|_2^2\# \tag{9}$$

This loss function encourages accurate reconstruction of normal temporal patterns. To avoid simplistic solutions, the model is primarily trained on normal data (which is common in unsupervised anomaly detection [5,9]). Therefore, the state space branch, attention branch, and gating must work together to reproduce normal patterns on the training set, with each branch learning complementary aspects (one learns common dynamics, and the other learns typical relevance and context). We note that robustness can be improved through additional regularization (e.g., sparsity in gating or shrinkage penalties), but in our experiments, the basic loss described above was sufficient.

2.6. Anomaly scoring and detection

During inference, TideMamba calculates anomaly scores for each time step (or the entire sequence) based on the reconstruction error. Since the model minimizes errors on the normalized data during training, any significant increase in reconstruction error indicates that the input pattern deviates from the learned normalized manifold, thus likely indicating the presence of anomalies. We define the anomaly score at time t as the normalized reconstruction error across all sensors:

$$s(t) = \frac{1}{N}\sum_{i=1}^{N}(x_i(t) - \widehat{x}_i(t))^2, t = 1, 2, \cdots, T\# \tag{10}$$

If $s(t)$ exceeds a chosen threshold λ, the time step t (or the corresponding wafer/cycle) is flagged as anomalous: $y(t) = 1$ if $s(t) > \lambda$ (and $y(t) = 0$ otherwise). In practice, λ can be set by statistically estimating the error distribution on a validation set or by using a high percentile of reconstruction errors (e.g. 99th percentile) as the cutoff [6]. The scoring can also be integrated over a window if anomalies are defined at the sequence level. In our wafer case, each process run (cycle) can be assigned an aggregate anomaly score (e.g., the maximum or average of $s(t)$ over that run). TideMamba's gated dual-branch design tends to produce low error for normal samples and spiked error for abnormal patterns, yielding a clear separation in $s(t)$ [17]. Moreover, the model provides interpretability: by inspecting which sensor components contribute most to $s(t)$ (the terms in the sum of Equation (10) and the gating vector g_t, engineers can identify which signals and dynamics were unusual. For example, an abnormal wafer might show a large error in the pressure sensor reconstruction specifically, and gating values near 0 for that sensor (meaning the attention branch, not the state branch, dominated—signaling a context-specific anomaly). This aligns with domain needs in semiconductor manufacturing, where knowing which sensor and when an anomaly occurred is as important as detecting it [22].

Before moving to results, we summarize the computational characteristics: the TideMamba model has a time complexity of $O(T * (M + D))$ per sequence (since the SSM branch is $O(T)$ with state dim M, and the attention/gating is $O(T * D)$ roughly). With M and D on the order of 102, this is highly efficient. In our experiments, training 60 epochs on 1500 time points (with D100) took only a few minutes on a GPU, and inference per run on CPU is a few seconds (details in Section 3.3). This scalability means TideMamba can be deployed for real-time monitoring even as data rates or sensor counts grow.

3. Results

We evaluate TideMamba on a multivariate time-series dataset from industrial wafer CVD processes, collected during the deposition stage of semiconductor manufacturing. Table 2 presents the dataset containing

Table 2
Dataset information.

Dataset	Domain	Time Points	Sensors	Sampling Rate	Anomaly Count (%)
Industrial CVD wafer	Semiconductor manufacturing (CVD process)	2,245 (1,571/ 225/449 train/ val/test)	1,596 raw (~100 final)	~10 Hz	32 (1.43%)

2,245 timestamps and 1,596 raw sensor variables; the final model input contains approximately 100 independent sensor features after preprocessing. These features cover key process measurements such as temperature, chamber pressure, gas flow, RF power/voltage, and status signals. The sequence was sampled at approximately 10 Hz across production runs and concatenated after removing gaps. We used a strict chronology-aware 70/10/20 split to prevent temporal leakage: the first 1,571 points (70.0%) were used for model fitting, the next 225 points (10.0%) were used only for validation and threshold calibration, and the final 449 points (20.0%) were held out for testing. All 32 labelled anomalies appear in the held-out test segment, corresponding to 1.4% of the full sequence and 7.1% of the test segment. This setting reflects a practical deployment scenario in which models are fitted on earlier normal process history and evaluated on later abnormal events. Each anomaly is treated as a binary abnormal label for detection; where available, process tags are used only for qualitative case interpretation.

3.1. Baseline methods and evaluation metrics

To provide a rigorous and comprehensive assessment of TideMamba, we compared it with ten representative anomaly detection baselines, namely InterFusion, IsolationForest, LOF [36], LSTM_AE, MTAD-GAT, OmniAnomaly, OneClassSVM, PCA_Recon, USAD, and AnomalyTransformer [37], thereby covering both classical machine-learning methods and recent deep neural approaches. All baseline models were implemented in Python or reproduced from publicly available code, and their hyperparameters were carefully tuned on the validation set. Following a standard binary anomaly detection setting, evaluation was conducted at both the point level and the event level, where each time step was classified as normal or anomalous and all points within a contiguous anomalous segment were treated as anomalous. Precision, Recall, F1-score, and F2-score were used to quantify detection quality, with F2 further emphasizing recall in scenarios where missing anomalies is especially costly. We additionally reported ROC-AUC and PR-AUC as threshold-independent indicators across operating points, with PR-AUC being particularly informative under severe class imbalance, and G-Mean to reflect balanced sensitivity and specificity while discouraging excessive false alarms. Finally, execution time was recorded under identical hardware settings, including the average inference time per sequence or per time point as well as notable differences in training cost and computational demand, so as to evaluate the practical suitability of each method for real-time monitoring applications.

3.2. Preprocessing and feature selection

As described in Section 2.1, we detrended and deseasonalized each sensor time-series. Additionally, we performed the following: (i) any sensor with nearly constant readings (standard deviation $<$ 1e-8) over the training period was dropped (there were $\sim$ 1230 such signals, mostly static setpoints or binary flags that didn't change); (ii) among the remaining $\sim$ 366 signals, we removed highly redundant ones by dropping one of any pair with correlation $>$ 0.995, leaving $\sim$ 192 features; (iii) we then retained the most informative $\sim$ 100 features using a combination of domain knowledge and mutual information selection (this step mainly removed channels that were mostly noise). The final set of $\sim$ 100 sensor variables include all those relevant to the physical process (temperatures, pressures, flows, power, etc.). We normalized each feature to zero mean and unit variance (using training set statistics). These $\sim$ 100 features constitute the input vector x_t used for model training/inference. Data example shows a snippet of pre-processed sensor data including an anomaly interval, illustrating multiple sensor traces. We report the average detection runtime per sequence (or per time point) for each method, as well as any notable training time differences. This is measured on the same hardware (classical methods on an Intel Xeon CPU, deep methods on an NVIDIA V100 GPU for training; for fairness, we measure inference time on CPU for all methods since deployment in fabs would likely be on CPUs.

3.3. Performance comparison

Table 3 compares TideMamba with ten baseline methods on the wafer CVD test set using precision, recall, F1 score, F2 score, AUC-ROC, G-mean, PR-AUC, and runtime. Overall, TideMamba provides the strongest score-ranking performance on this dataset and achieves the highest precision among the neural methods, while remaining computationally efficient. Fig. 4 summarizes the comparative performance and runtime patterns.

In terms of ranking quality, TideMamba attains the highest AUC-ROC of 0.989 and the highest PR-AUC of 0.791, exceeding the strongest competing baselines on both criteria. The closest AUC-ROC is obtained by OmniAnomaly (0.986), whereas the closest PR-AUC is obtained by LSTM-AE (0.776). This result is important for imbalanced industrial anomaly detection, where the quality of anomaly ranking is often more informative than a single thresholder operating point. TideMamba also achieves the highest precision (0.840), tied with MTAD-GAT (0.840), indicating that its alarms are comparatively reliable and less prone to false triggering. At the same time, TideMamba maintains competitive point-wise detection performance. Its F1 score reaches 0.737, which is the best among all neural baselines. The highest F1 score in the table is 0.746, obtained by Isolation Forest, LOF, and One-Class SVM. However, these classical methods remain clearly below TideMamba in AUC-ROC and PR-AUC, suggesting that their final binary decisions are competitive on this split, while their anomaly score distributions are less robust overall. A similar pattern is observed for F2: methods with higher recall, such as OmniAnomaly (0.719) or the classical baselines (0.688 recall), obtain slightly higher recall-oriented F2 values, whereas TideMamba favors a more precision-preserving operating regime with F2 $=$ 0.686.

The baseline comparison further reveals distinct error profiles. InterFusion is highly conservative, with precision $=$ 0.833 but recall $=$ 0.469, implying that it misses a substantial portion of anomalies despite producing relatively few false alarms. OmniAnomaly exhibits the opposite tendency: it reaches the highest recall (0.719) but with markedly lower precision (0.676), indicating a higher false-positive rate. LSTM-AE provides a relatively balanced profile (precision $=$ 0.759, recall $=$ 0.688, F1 $=$ 0.721) and constitutes the strongest deep baseline overall, yet it still falls short of TideMamba on the most discriminative ranking metrics. Notably, the inclusion of the more recent Anomaly-Transformer baseline further strengthens the comparison; although it achieves AUC-ROC $=$ 0.975 and PR-AUC $=$ 0.746, its thresholder performance (precision $=$ 0.636, F1 $=$ 0.646) remains weaker than that of TideMamba. TideMamba is also favourable from the perspective of deployment efficiency. Its average runtime is 3.830 s per sequence, which is the lowest among the ten baseline methods. This result indicates that the performance gains of TideMamba are not obtained at the expense of higher computational cost; rather, the model offers a practically attractive balance between ranking accuracy, alarm reliability, and inference speed. This also fully reflects that Mamba benefits from the advantages of data interaction with hardware storage cache (hardware aware parallel algorithm, efficient parallelization), has relatively high memory efficiency, linear memory consumption, and can effectively process million level sequence data.

3.4. Latent representation and separation

To understand how TideMamba distinguishes normal vs. anomalous runs, we visualized the learned latent representations. Fig. 5 shows a t-SNE plot of test sequence embeddings and associated confusion matrices for different methods. TideMamba produces a tight cluster for normal runs and clearly separated points for anomalies in the t-SNE space, indicating that its encoding space discriminates the classes well. Furthermore, the Fig. 5 shows the highest overlap between TideMamba predicted points (green circles) and labelled points (orange crosses),

Table 3
Performance of TideMamba and baseline methods on the wafer CVD dataset. (Higher is better for all metrics except time. The best result in each column is bold.).

Algorithm	Precision	Recall	F1	F2	AUC_ROC	G-mean	PR_AUC	Runtime (s)
InterFusion	0.833	0.469	0.600	0.514	0.815	0.684	0.480	80.001
IsolationForest	0.815	0.688	**0.746**	**0.710**	0.911	0.828	0.702	5.142
LOF	0.815	0.688	0.746	0.710	0.972	0.828	0.726	15.267
LSTM_AE	0.759	0.688	0.721	0.701	0.984	0.828	0.776	30.538
MTAD-GAT	0.840	0.500	0.627	0.544	0.864	0.707	0.524	14.320
OmniAnomaly	0.676	**0.719**	0.697	0.710	0.986	**0.846**	0.771	12.202
OneClassSVM	0.815	0.688	0.746	0.710	0.972	0.828	0.727	59.585
PCA_Recon	0.810	0.531	0.642	0.570	0.815	0.728	0.490	114.008
USAD	0.833	0.625	0.714	0.658	0.975	0.790	0.687	18.721
AnomalyTransformer	0.636	0.656	0.646	0.652	0.975	0.808	0.746	8.350
TideMamba	**0.840**	0.656	0.737	0.686	**0.989**	0.809	**0.791**	**3.830**

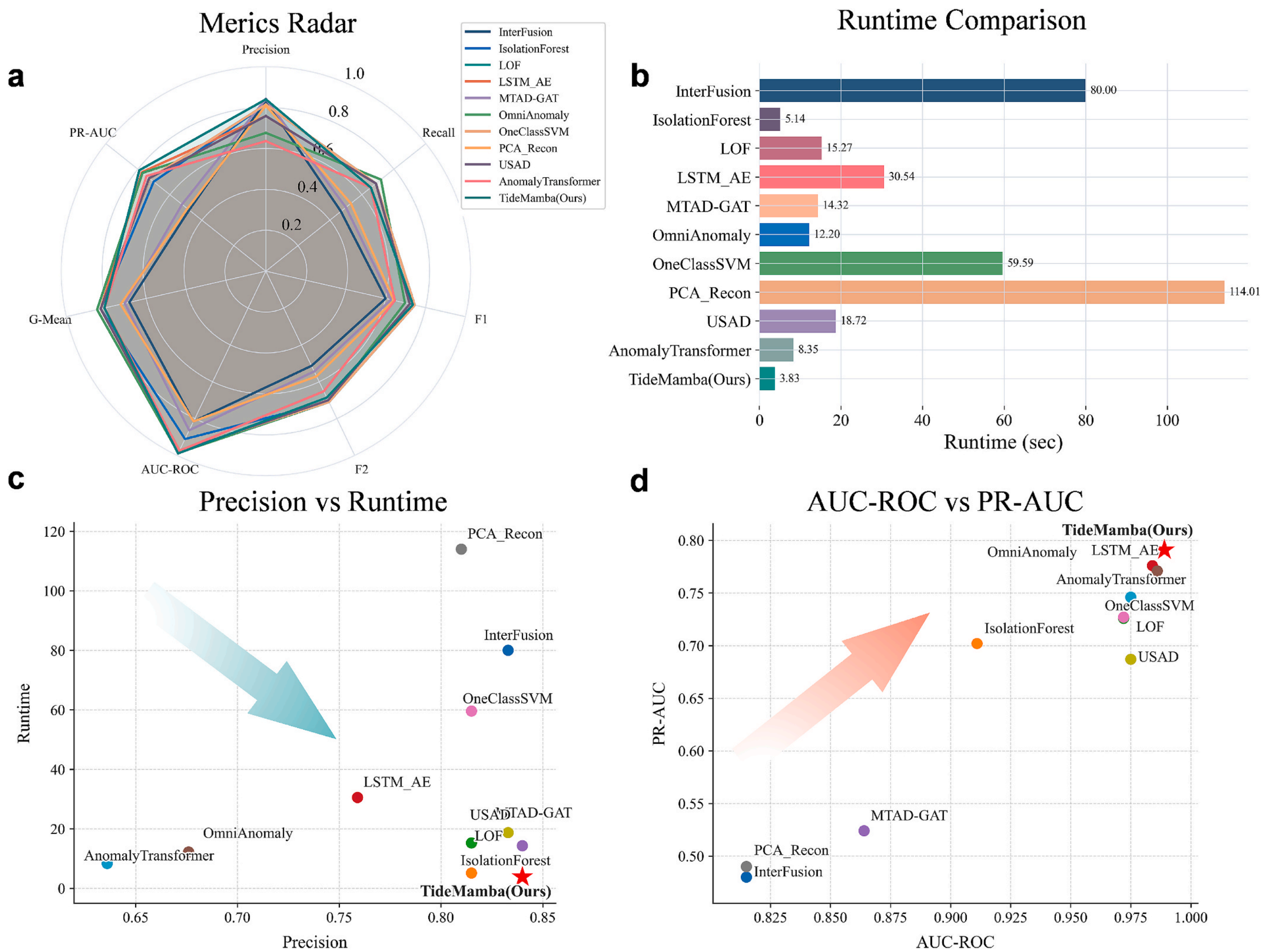

Fig. 4. Comprehensive performance comparison of TideMamba with 10 baseline algorithms. (a) Radar chart of seven key metrics (Precision, Recall, F1, F2, AUC-ROC, G-Mean, PR-AUC). A larger radius indicates better performance. TideMamba (ours) shows a well-balanced, outward-spreading polygon, indicating top-tier performance across most metrics. (b) Runtime comparison. (c) Two-dimensional projection of performance vs. cost: the left plot shows Precision vs. Runtime for each method (TideMamba sits in the high-Precision/low-Runtime corner / green arrow facing), and the right plot shows AUC-ROC vs. Runtime (TideMamba in the high-AUC-ROC/high-PR-ROC quadrant/ red arrow facing).

achieving the best trade-off compared to the missed detections of InterFusion and the excessive false positives of LSTM_AE. Consequently, its confusion matrix (Fig. 5, TideMamba panel) shows high true positives and true negatives with few misclassifications (the bright diagonal and minimal off-diagonals). In contrast, baseline methods like LSTM-AE or OmniAnomaly have more intermixed t-SNE embeddings—some anomalies embed near normals—leading to confusion matrices with more false negatives or false positives. For example, OmniAnomaly's stochastic encoding sometimes scatters normal and anomalous points, explaining its lower precision (some false alarms). InterFusion's latent space, while hierarchical, can mix some anomalies with normals if the anomaly doesn't strongly violate the learned inter-metric correlations, consistent with its lower recall. TideMamba's superior separation can be attributed to the gating mechanism: when an anomaly occurs, the

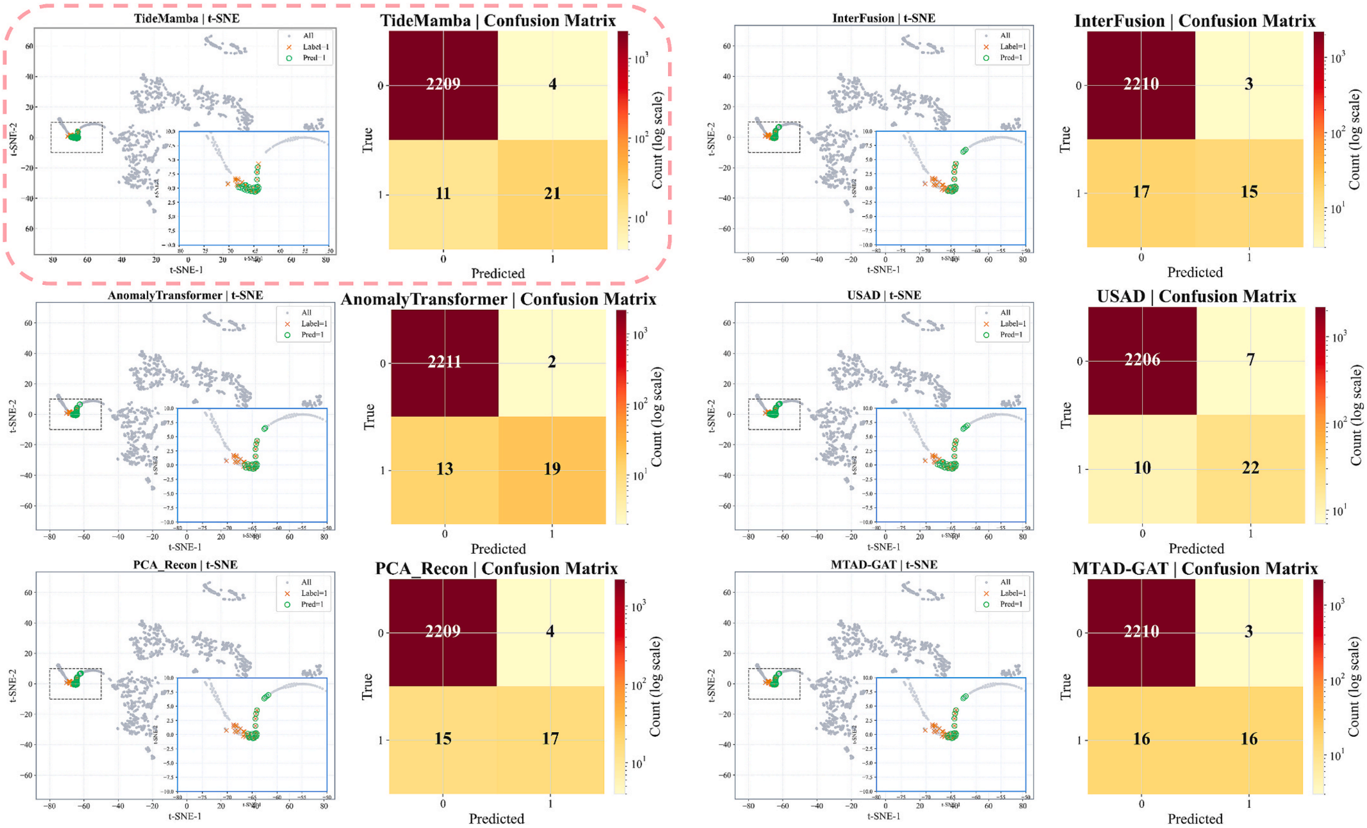

Fig. 5. T-SNE visualization of potential representations learned under different methods and confusion matrix for their classification. Each point in the t-SNE graph represents a timestamp of wafer sensing encoded by the model (color coded as normal or abnormal). TideMamba produces well separated clusters, while the baseline method shows more overlap between predicted anomalous time points (green) and anomalous label embeddings (orange red). The confusion matrix (on the right) quantifies classification performance: TideMamba's matrix shows higher true positive and true negative rates (dark cells on the diagonal) and lower false positive rates (fuzzy cells outside the diagonal), consistent with its higher F1 score. The methods such as InterFusion and OmniAnomaly showed high misclassification rates, manifested as brighter non diagonal cells, consistent with the lower accuracy/recall rates in Table 3.

attention branch output $z^{(a)}$ deviates significantly from the state branch output $z^{(s)}$ (which is still following learned normal dynamics), resulting in a fused representation z that cannot be mistaken for a normal pattern. The model essentially projects anomalies to a distinct region of latent space, easing downstream classification.

3.5. Temporal detection and case studies

Fig. 6 shows the anomaly detection traces of TideMamba on the sensor. Fig. 6a illustrates the reconstruction error components representing the sensor during the test run. During normal operation, the error remains low and stable. When anomalies occur (in shaded areas), the error of TideMamba on certain sensors will sharply increase, indicating that it cannot reconstruct these signals (indicating that their residual patterns deviate from the learned normal manifold). Fig. 6b shows the distribution of predicted abnormal labels and actual abnormal labels on the timeline: TideMamba's prediction is highly consistent with the actual abnormal range, with almost no false positives. Fig. 6c shows the variation of the anomaly score $s(t)$ over time; In normal times, the score approaches zero, but significantly increases in abnormal times, creating a lot of room for threshold setting. Field engineers have found that these outputs are interpretable: the error composition graph (Fig. 6a) helps identify which sensors contribute the most to anomalies, and the time score graph (Fig. 6c) shows the time when the process begins to lose control, which is crucial for real-time fault diagnosis and mitigation. Engineers can use these outputs as decision-support information: the sensor-level error chart suggests which process variables contributed most to the alarm, and the score trace indicates when the process began to depart from normal operation. These observations are practically suggestive but should not be interpreted as a systematic human-in-the-loop validation study. In Fig. 6d, we highlight three detected abnormal cases and magnify the sensor readings. In each case, TideMamba successfully identified anomalies: for example, a brief abnormal drop in the pressure sensor – TideMamba's attention branch treated it as a contextual anomaly (linking the pressure drop to simultaneous irregular gas flow readings), resulting in high scores and gating, highlighting the attention path in that interval. In another example, the temperature sensor deviated from the normal range at the end of operation; In this case, the prediction error of the state space branch steadily increases, even warning of anomalies before the process ends. These case studies demonstrate TideMamba's sensitivity to point anomalies (sudden peaks/drops) and pattern anomalies (shape or trend deviations) in a multivariate environment. By excluding global trends in advance, even gradual drift is considered an anomaly with a flat residual baseline.

3.6. Interpretability and sensor attribution

Fig. 7 summarizes the interpretability of TideMamba in industrial anomaly detection. Fig. 7a presents an error heatmap, where time and sensor metrics form the two axes and colour intensity denotes the magnitude of reconstruction error. Under normal operation, only weak background noise is observed, whereas anomalies emerge as bright red bars localized to specific sensors and time points, consistent with sparse sensor anomalies in the time domain. TideMamba’s residual-focused modelling enhances these patterns, making anomalous “spokes”

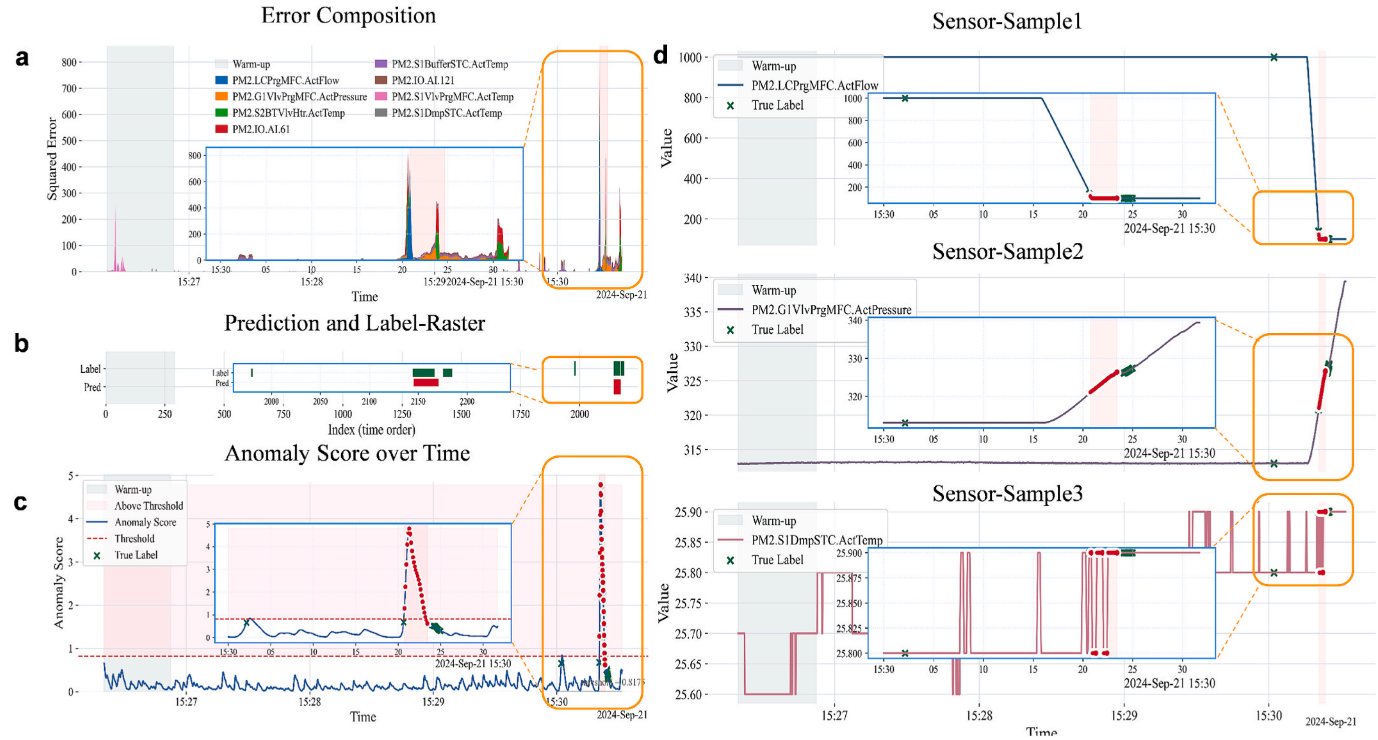

Fig. 6. Time-series results demonstrating TideMamba's anomaly detection on wafer sensor data. (a) Error components for key sensors over time: each sub-plot shows the reconstruction error $|x_i(t) - \hat{x}_i(t)|$ for a particular sensor. Errors remain low during normal operation and spike during an anomaly (red shaded region), identifying which sensor variables experience abnormal deviations. (b) Anomaly predictions vs. ground truth labels on a timeline (binary indicators): TideMamba's predictions closely match the true anomaly segments, with timely detection when the anomaly onset occurs. (c) Anomaly score $s(t)$ over time for the run: the score is near zero during normal periods and rises sharply during anomalies, providing a clear separation. (d) Three example anomaly cases (zoomed-in sensor plots): TideMamba captures each case – (d1) a short abnormal spike in one sensor, (d2) a context anomaly involving a combination of sensor readings deviating together, (d3) a gradual drift beyond normal bounds – reflecting the model's ability to detect diverse anomaly types.

visually salient. Fig. 7b shows the sensor correlation matrix based on the Pearson correlation coefficient, reflecting population-level covariance relationships. Although this matrix is not a direct output of TideMamba, it provides an important baseline for interpretation, because the attention branch effectively learns a dynamic, context-specific version of this correlation structure. Anomalies are often associated with moments when instantaneous correlations deviate from the static pattern within the range −1 to 1. Fig. 7c compares anomaly score distributions for normal and anomalous points. Anomaly scores are clearly shifted upward, whereas normal points cluster at low values, indicating that a simple threshold can separate the two with high confidence and that TideMamba scores can further rank anomalies by severity. Fig. 7d presents sensor contribution analysis, in which each sensor's contribution is approximated by summing reconstruction errors over detected anomalous intervals. In our experiments, chamber-pressure and gas-flow-related sensors contributed strongly, which is consistent with domain knowledge, while several auxiliary sensors also contributed, suggesting possible secondary effects [12]. The gating signal provides an additional diagnostic cue. Under the convention in Equation (7), a lower SSM gate value, or equivalently a higher attention weight 1-G(t), indicates that the fused reconstruction relies more on the attention branch. In anomaly cases, this shift can suggest that local/contextual patterns are not fully explained by the long-term SSM dynamics [16]. These outputs are useful for exploratory root-cause analysis, but they should be interpreted cautiously: they are model-derived diagnostic indicators rather than experimentally validated causal explanations.

The comparative behaviour of the baselines helps explain TideMamba's advantages and remaining limitations. InterFusion may overgeneralize normal patterns and underfit some anomalous features. OmniAnomaly achieves a comparable AUC-ROC (0.986), but its precision is lower, likely due to stochastic latent-space variability. LSTM-AE performs reasonably well (F1 = 0.721), yet it lacks explicit mechanisms for both long-range state-space dynamics and adaptive cross-sensor context. MTAD-GAT depends on learned graph attention that may be sensitive to graph quality, whereas TideMamba learns temporal and contextual dependencies jointly. Classical methods, including Isolation Forest, LOF, PCA, and One-Class SVM, do not model temporal dynamics explicitly; this can limit their ability to capture slow drift or context-dependent deviations [11,14]. TideMamba's time-domain modelling increases reconstruction error progressively during drift, which can support earlier warning in such cases. Moreover, this evidence is promising for practical interpretation, but it is not a formal validation of human-in-the-loop explainability. A systematic user study or prospective deployment study would be required to claim validated interpretability.

3.7. Ablation study

To evaluate the contribution of each architectural component in TideMamba, we conducted a component-isolation ablation study using the same test split and thresholding protocol as the full model. The evaluated variants include TideMamba_NoDetrend, which removes the trend and periodicity decomposition step and directly models the raw sensor sequences; TideMamba_NoGate, which replaces the adaptive gating module with a fixed fusion strategy; TideMamba_NoSSM, which removes the state-space modelling branch; TideMamba_NoAttention, which removes the attention branch and relies mainly on the SSM-based reconstruction path; and TideMamba_AttentionOnly, which isolates the attention-based reconstruction path. The complete results are shown in Table 4.

The ablation results demonstrate that the decomposition module is the most critical component. When trend and periodicity removal is

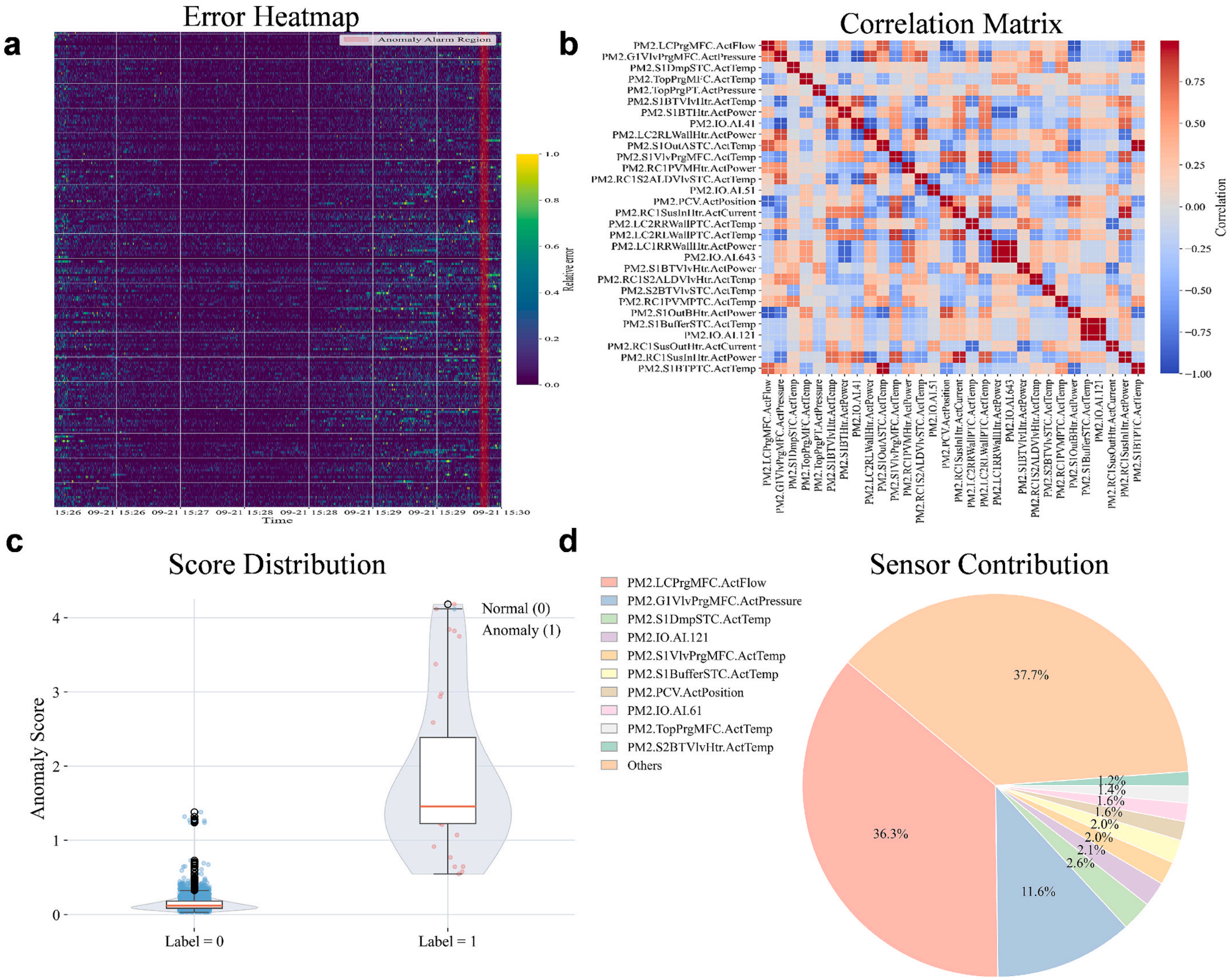

Fig. 7. Additional visualizations for interpretability and analysis of TideMamba's performance. (a) Error heatmap (time vs. sensor index) for a test run: normal operation yields low-error (blue/green) background, while anomaly intervals show clear red spikes on specific sensors, indicating when and where the model's reconstruction error was high. (b) Correlation matrix of sensor variables (based on normal data): color denotes Pearson correlation between sensor pairs (blue for negative, red for positive). This static correlation informs expected inter-sensor relationships. Anomalies often correspond to temporary violation of these correlations (e.g., a pair of sensors momentarily becoming uncorrelated despite a high normal correlation). (c) Score distribution for normal vs. anomalous samples: TideMamba's anomaly score $s(t)$ is significantly higher on anomalies (red density) than normals (blue density), allowing a threshold (dotted line) that cleanly separates the two with minimal overlap. (d) Sensor contribution chart: the proportion of total anomaly indication coming from each sensor, aggregated over all test cases (a larger slice means that sensor frequently had high error during anomalies). This highlights the most critical sensors that drive anomaly detection decisions, offering explainability by pinpointing which aspects of the process are most vulnerable.

Table 4
Comparison of ablation experiments.

Algorithm	Precision	Recall	F1	F2	AUC_ROC	G-mean	PR_AUC	Runtime(s)
TideMamba_NoDetrend	0.344	0.344	0.344	0.344	0.929	0.584	0.126	2.981
TideMamba_NoGate	0.808	0.656	0.724	0.682	0.982	0.809	0.766	2.915
TideMamba_NoSSM	0.808	0.656	0.724	0.682	0.980	0.809	0.759	0.762
TideMamba_NoAttention	0.636	0.656	0.646	0.652	0.980	0.808	0.751	2.382
TideMamba_AttentionOnly	0.840	0.656	0.737	0.686	0.985	0.809	0.760	3.604
TideMamba	0.840	0.656	0.737	0.686	0.989	0.809	0.791	3.830

disabled, TideMamba_NoDetrend shows a substantial performance degradation, with Precision, Recall, F1, and F2 all decreasing to 0.344. Its PR-AUC also drops sharply from 0.791 in the full TideMamba model to 0.126, while its G-mean decreases from 0.809 to 0.584. Although its AUC-ROC remains relatively high at 0.929, the very low PR-AUC indicates that the model without decomposition performs poorly under the highly imbalanced anomaly detection setting. This suggests that modelling raw non-stationary sensor sequences directly makes it difficult to isolate rare abnormal patterns, whereas removing low-frequency trend and periodic components helps expose residual fluctuations that are more informative for anomaly detection. The adaptive gating mechanism also contributes to the overall robustness of the model.

Without the gating module, TideMamba_NoGate obtains Precision = 0.808, Recall = 0.656, F1 = 0.724, F2 = 0.682, AUC-ROC = 0.982, and PR-AUC = 0.766. Compared with the full TideMamba model, removing the adaptive gate preserves recall but reduces precision, F1, AUC-ROC, and PR-AUC. This indicates that fixed fusion can still detect a similar number of anomalous points at the selected threshold, but adaptive fusion provides a more reliable score ranking and a better precision-recall trade-off.

The branch-isolation results further show that the SSM and attention branches provide complementary information. Removing the SSM branch gives TideMamba_NoSSM a Precision of 0.808, Recall of 0.656, F1 of 0.724, AUC-ROC of 0.980, and PR-AUC of 0.759. Removing the attention branch has a stronger negative effect on threshold-level performance: TideMamba_NoAttention obtains Precision = 0.636, Recall = 0.656, F1 = 0.646, F2 = 0.652, AUC-ROC = 0.980, and PR-AUC = 0.751. The unchanged recall but much lower precision suggests that the SSM-only variant can still identify part of the anomalous region, but it introduces more false positives at the chosen operating point. In contrast, TideMamba_AttentionOnly reaches the same Precision, Recall, F1, and F2 as the full model, namely 0.840, 0.656, 0.737, and 0.686, respectively. However, its AUC-ROC and PR-AUC are lower, at 0.985 and 0.760, compared with 0.989 and 0.791 for the full TideMamba model. This indicates that the attention branch alone can perform well at the selected threshold, but the combined architecture provides stronger threshold-independent ranking quality. Runtime results also reveal different computational characteristics among the ablated variants. TideMamba_NoSSM is the fastest variant, with a runtime of 0.762 s, followed by TideMamba_NoAttention at 2.382 s, TideMamba_NoGate at 2.915 s, and TideMamba_NoDetrend at 2.981 s. TideMamba_AttentionOnly requires 3.604 s, while the complete TideMamba model has the highest runtime among the ablation variants at 3.830 s. Despite this additional cost, the full model achieves the best AUC-ROC and PR-AUC, showing that the modest increase in runtime is accompanied by improved anomaly score quality.

Moreover, the decomposition stage is essential, as removing it causes the largest degradation across precision-recall metrics. The adaptive gating improves the reliability of branch fusion, especially in terms of precision and ranking performance. Although the attention-only variant matches the full model at the selected threshold, the complete TideMamba architecture achieves the strongest overall ranking performance, with the highest AUC-ROC (0.989) and PR-AUC (0.791). Therefore, the combination of residual decomposition, SSM-based temporal modelling, attention-based contextual modelling, and adaptive gating provides the most reliable configuration for the studied CVD anomaly detection task.

3.8. Hyperparameter sensitivity

To evaluate whether the performance of TideMamba depends strongly on specific architectural or temporal settings, we conducted a controlled hyperparameter sensitivity analysis on three key factors: the Mamba state dimension, the attention-branch configuration, and the input window length. In each group of experiments, only one factor was varied while the remaining training and detection protocol was kept unchanged. The default setting used in the main TideMamba design was defined as $d_{state} = 64$, hidden dimension = 256, one residual attention layer with eight heads, window length = 512, stride = 64, learning rate = 1×10^{-3}, batch size = 32, and a maximum of 60 epochs with early stopping. For the window-length experiments, the stride was adjusted proportionally as one-eighth of the window length, resulting in strides of 32, 64, and 128 for window sizes of 256, 512, and 1024, respectively.

The complete results are summarized in Table 5 and visualized in Fig. 8. Overall, the model showed stable threshold-level detection performance across a wide range of state dimensions. When d_{state} was varied from 16 to 128, precision, recall, F1, F2, and G-mean remained identical at the selected operating point, with precision = 0.840, recall = 0.656, F1 = 0.737, F2 = 0.686, and G-mean = 0.809 for all four settings. However, the threshold-independent ranking metrics were more sensitive. The default state dimension $d_{state} = 64$ achieved the highest AUC-ROC and PR-AUC in this group, with AUC-ROC = 0.990 and PR-AUC = 0.798. Smaller state dimensions produced lower PR-AUC values, namely 0.748 for $d_{state} = 16$ and 0.753 for $d_{state} = 32$, whereas increasing the state dimension to 128 reduced PR-AUC to 0.771. These results indicate that $d_{state} = 64$ provides the best ranking quality among the tested state dimensions, rather than merely matching the fixed-threshold detection results.

The attention-configuration analysis showed a clearer effect on the operating-point metrics. Removing the attention branch reduced precision to 0.636 and F1 to 0.646, although recall remained 0.656. Introducing a one-layer attention branch with four heads improved precision to 0.840 and F1 to 0.737, with a runtime of 3.532 s. The eight-head, one-layer reference configuration achieved precision = 0.808, recall = 0.656, F1 = 0.724, and PR-AUC = 0.751. Increasing the depth to two attention layers recovered F1 to 0.737, but runtime increased substantially to 6.223 s, indicating that deeper attention brought no clear detection advantage relative to its computational cost. The no-gate variant obtained a relatively high PR-AUC of 0.768 but a much lower fixed-threshold precision of 0.636 and F1 of 0.646. This discrepancy suggests that adaptive gating mainly improves the final alarm operating point, even when the underlying score ranking remains competitive. Thus, the attention branch and the gating mechanism are both important for converting anomaly scores into reliable binary alarms. Window length affected the precision–recall trade-off. The shorter window of 256 achieved the highest recall and F2 score, with recall = 0.688 and F2 = 0.705, and also had the lowest runtime of 2.414 s. However, its precision was lower at 0.786. The default window length of 512 achieved the best F1 and PR-AUC in the window group, with precision = 0.840, recall = 0.656, F1 = 0.737, and PR-AUC = 0.763. Increasing the window length to 1024 did not improve performance; instead, F1 decreased to 0.724 and PR-AUC decreased to 0.744. These results indicate that excessively long windows may add temporal context without improving anomaly discrimination on this dataset, whereas the 512-point window provides

Table 5
Hyperparameter sensitivity of TideMamba under the same validation/test protocol.

Sensitivity factor	Setting	Precision	Recall	F1	F2	AUC-ROC	G-mean	PR-AUC	Runtime (s)
State dimension	$d_{state} = 16$	0.840	0.656	0.737	0.686	0.979	0.809	0.748	4.917
State dimension	$d_{state} = 32$	0.840	0.656	0.737	0.686	0.979	0.809	0.753	3.680
State dimension	$d_{state} = 64$	0.840	0.656	0.737	0.686	0.990	0.809	0.798	3.959
State dimension	$d_{state} = 128$	0.840	0.656	0.737	0.686	0.985	0.809	0.771	3.894
Attention configuration	No attention	0.636	0.656	0.646	0.652	0.980	0.808	0.751	2.382
Attention configuration	4 heads / 1 layer	0.840	0.656	0.737	0.686	0.982	0.809	0.762	3.532
Attention configuration	8 heads / 1 layer	0.808	0.656	0.724	0.682	0.977	0.809	0.751	3.862
Attention configuration	8 heads / 2 layers	0.840	0.656	0.737	0.686	0.981	0.809	0.760	6.223
Attention configuration	8 heads / 1 layer, no gate	0.636	0.656	0.646	0.652	0.984	0.808	0.768	3.341
Window length	256	0.786	0.688	0.733	0.705	0.976	0.828	0.751	2.414
Window length	512	0.840	0.656	0.737	0.686	0.982	0.809	0.763	4.001
Window length	1024	0.808	0.656	0.724	0.682	0.978	0.809	0.744	3.645

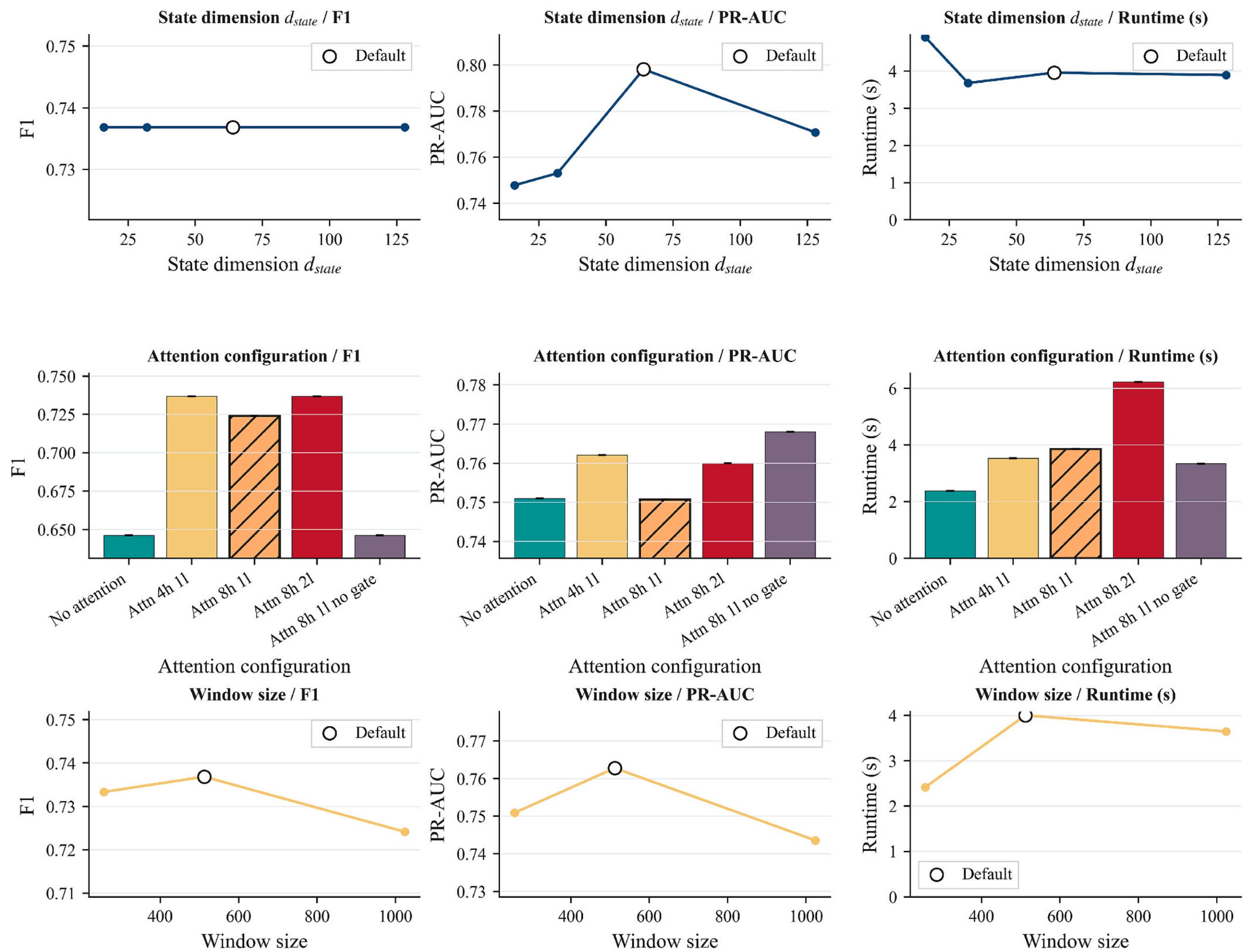

Fig. 8. Hyperparameter sensitivity analysis of TideMamba. Effects of the Mamba state dimension, attention configuration, and input window length on F1 score, PR-AUC, and runtime. The results show that $d_{state} = 64$ gives the strongest ranking performance among the tested state dimensions, attention improves fixed-threshold detection quality compared with the no-attention setting, and the 512-point window provides the best F1 and PR-AUC among the tested temporal windows.

a better balance between precision and recall.

Moreover, TideMamba is not dependent on a narrowly tuned state dimension; however, $d_{state} = 64$ gives the strongest ranking performance among the tested state sizes. Attention and gating are useful for improving the fixed-threshold alarm quality, while increasing attention depth from one to two layers mainly increases runtime. The 512-point window is a reasonable temporal scale for the studied CVD data because it achieves the strongest F1 and PR-AUC among the tested window lengths. Since each setting was evaluated once on the available labelled sequence, these results should be interpreted as empirical sensitivity evidence rather than a statistical significance test across multiple independent datasets.

4. Discussion

The results indicate that TideMamba satisfies the main practical requirements of multivariate time-series anomaly detection in the evaluated CVD setting: accurate ranking, efficient inference, and practically useful diagnostic outputs. The evaluation was performed on one industrial CVD dataset, where approximately 100 selected sensor features, including temperatures, pressures, gas flows, and RF powers, were monitored. In this setting, 32 labelled abnormal points were observed in the held-out test segment, spanning abrupt spikes, drops, and gradual process drifts. No baseline dominates TideMamba across all metrics. Instead, the comparisons show complementary weaknesses among existing methods: OmniAnomaly favors sensitivity but yields lower precision, whereas InterFusion is more conservative but exhibits low recall. TideMamba achieves a balanced precision-recall profile and the best ranking metrics under the reported protocol. These findings support the value of combining residual decomposition, selective SSM, attention-based context, and adaptive fusion for this industrial dataset, while they should not be interpreted as proof of universal superiority across all anomaly detection contexts.

An important observation is the model's ability to handle different anomaly regimes within the same dataset. The labelled wafer events include sudden point anomalies and longer-duration contextual anomalies. For point anomalies, such as a brief excursion in one sensor, the attention branch can emphasize the affected instant and its surrounding context, producing a strong reconstruction-error response. For contextual anomalies, both branches are involved: the SSM branch reflects mismatch with the learned temporal trajectory, while the attention branch highlights inconsistencies in sensor context, including correlated drifts across variables. This complementary interaction broadens the detectable anomaly space within the evaluated data. The error analysis also clarifies present limits. Missed detections are mainly associated with events that are extremely weak or transient and are therefore close to background noise. Lowering the threshold would likely recover some of these small events but would also increase false alarms. The operating point therefore remains application-dependent. Conversely, false positives tend to occur during transient process transitions, such as setpoint changes or startup/shutdown phases, that are under-represented in the fitting segment. This observation points to concept drift: when the definition of normal behavior evolves, a static detector may misclassify previously unseen but legitimate patterns. Long-term deployment would benefit from periodic retraining, fine-tuning on newly verified normal data, or online adaptation. Such extensions are feasible because the state-space formulation can be updated recursively and the training strategy does not require anomaly labels for new normal data.

Computationally, TideMamba is suitable for online or near-online monitoring at the scale evaluated here. Although memory consumption was not explicitly tabulated, the model is substantially lighter than many full Transformer-style alternatives. A full Transformer handling long windows has quadratic sequence-length complexity due to self-attention, whereas the Mamba SSM branch scales linearly with sequence length and the attention module is bounded within the reconstruction window. Supplemental window-length experiments summarized in Table 5 show that increasing the window length raises runtime approximately monotonically while leaving ranking metrics stable over the tested range. This pattern is consistent with the expected scalability of the architecture. The reported inference time of a few seconds per sequence on CPU further indicates that production deployment is realistic for similar data rates and feature counts. TideMamba also provides practically suggestive interpretability. The adaptive gate acts as a fusion signal, while sensor-wise reconstruction errors provide direct attribution to process variables. In one pressure-spike case, the relative attention contribution increased during the anomaly compared with nearby normal operation, which is consistent with the intuition that the event was localized and context-dependent. However, this should be framed as diagnostic evidence rather than validated causal explanation. Systematic operator studies and prospective deployment trials are needed before claiming confirmed industrial human-in-the-loop interpretability.

Although the present study focuses on semiconductor wafer fabrication, the architectural idea is not inherently restricted to this domain. Many industrial and cyber-physical systems show long-term drift, multivariate dependence, and abrupt faults, making applications such as power-plant monitoring, aircraft engine telemetry, and IT or network operations plausible future targets. Nevertheless, the current empirical evidence is limited to a single domain-specific dataset from one data source. Testing on multiple time batches of the same device supports within-source robustness but does not substitute for cross-dataset validation. Broader benchmarking on additional datasets, different equipment, and different fault taxonomies will be necessary to establish the generality of the observed gains in accuracy, efficiency, and interpretability.

5. Conclusion

In this work, we introduced TideMamba, a novel unsupervised anomaly detection framework that integrates a Mamba SSM branch and an attention-based branch within a dual-stream architecture. TideMamba addresses two core challenges in time-series anomaly detection: capturing long-range temporal dependencies and adapting to non-stationary trends and contextual variations while maintaining a low false-alarm rate. The Mamba SSM branch models and extrapolates smooth long-term system dynamics to generate robust predictions of normal behaviour, whereas the self-attention branch captures instantaneous multivariate dependencies and localized deviations, making it sensitive to sharp or context-dependent anomalies. Through an adaptive gating mechanism, the two branches are dynamically fused at each time step, allowing the model to emphasize anomaly-sensitive responses when the attention branch detects patterns beyond the predictive scope of the SSM, while relying on stable sequential forecasts under normal conditions. Experiments on a real semiconductor manufacturing dataset show that TideMamba outperforms ten state-of-the-art baselines in overall detection accuracy, achieving the best AUC-ROC and PR-AUC while maintaining strong computational efficiency and competitive precision for high-throughput industrial deployment. It also detects heterogeneous fault patterns effectively, including abrupt sensor spikes, equipment malfunctions, and subtle progressive drifts, demonstrating strong versatility in complex multivariate environments. A major strength of TideMamba is its interpretability. By analysing reconstruction errors and gating signals, the framework not only identifies anomalies but also reveals the sensors and latent factors most likely involved. Visualization tools such as error heatmaps, t-SNE embeddings, confusion matrices, and sensor contribution charts provide interpretable insight into anomaly timing, location, separability, and sensor-level attribution. This transparency is especially important in safety–critical domains such as semiconductor manufacturing, where black-box predictions alone are insufficient for operational trust. Future work may further enhance the framework. Incorporating an explicit external memory or feedback mechanism could help the model reference prototypical abnormal patterns and improve recall for borderline anomalies without sacrificing precision. Additional directions include online learning, broader cross-domain validation, and integration with root cause analysis modules for automated diagnosis. Overall, TideMamba represents a meaningful step toward reliable, efficient, and interpretable AI for time-series anomaly detection, with strong potential for smart manufacturing and wider IoT deployments.

CRediT authorship contribution statement

Mingwei Feng: Writing – original draft, Visualization, Validation, Supervision, Software, Methodology, Conceptualization. **Xuping Zhang:** Supervision, Funding acquisition. **Kangjian Di:** Formal analysis, Data curation. **Zikang Zeng:** Validation, Software, Investigation. **Yiru Wang:** Validation, Software, Investigation. **Bohan Wang:** Visualization, Validation, Investigation. **Tianrui Zhang:** Validation, Formal analysis. **Yali Zhang:** Validation, Investigation. **Yixin Zhang:** Writing – review & editing, Supervision, Funding acquisition. **Ningmu Zou:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant 62341408 and the Fundamental Research Funds for the Central Universities 483014380006.

Data availability

Data will be made available on request.

References

[1] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 15:1–15:58. 10.1145/1541880.1541882.

[2] G.A. Susto, M. Terzi, A. Beghi, Anomaly detection approaches for semiconductor manufacturing, Procedia Manuf. 11 (2017) 2018–2024, https://doi.org/10.1016/j.promfg.2017.07.353.

[3] S.-K.-S. Fan, C.-Y. Hsu, D.-M. Tsai, F. He, C.-C. Cheng, Data-driven approach for fault detection and diagnostic in semiconductor manufacturing, IEEE Trans. Autom. Sci. Eng. 17 (4) (2020) 1925–1936, https://doi.org/10.1109/TASE.2020.2983061.

[4] C.-Y. Chen, S.-C. Chang, D.-Y. Liao, Equipment anomaly detection for semiconductor manufacturing by exploiting unsupervised learning from sensory data, Sensors 20 (19) (2020) 5650, https://doi.org/10.3390/s20195650.

[5] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M.A. Zuluaga, Usad,, UnSupervised anomaly detection on multivariate time series, in: In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3395–3404, https://doi.org/10.1145/3394486.3403392.

[6] J.E. Jackson, G.S. Mudholkar, Control procedures for residuals associated with principal component analysis, Technometrics 21 (3) (1979) 341–349, https://doi.org/10.1080/00401706.1979.10489779.

[7] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, D. Pei, Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding, in: In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 3220–3230, https://doi.org/10.1145/3447548.3467075.

[8] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation Forest, in: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 413–422, https://doi.org/10.1109/ICDM.2008.17.

[9] A. Gu, C. Gulcehre, T. Paine, M. Hoffman, R. Pascanu, Improving the gating mechanism of recurrent neural networks, in: Proceedings of the 37th International Conference on Machine Learning, Proc. Mach. Learn. Res. 119 (2020) 3800–3809.

[10] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, LSTM-based encoder-decoder for multi-sensor anomaly detection, arXiv preprint arXiv: 1607.00148, 2016. 10.48550/arXiv.1607.00148.

[11] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828–2837, https://doi.org/10.1145/3292500.3330672.

[12] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, L. Chang, A novel anomaly detection scheme based on principal component classifier, in: Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in: In Conjunction with the Third IEEE International Conference on Data Mining, 2003, pp. 172–179.

[13] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (7) (2001) 1443–1471, https://doi.org/10.1162/089976601750264965.

[14] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, Q. Zhang, Multivariate time-series anomaly detection via graph attention network, in: In: Proceedings of the 2020 IEEE International Conference on Data Mining, 2020, pp. 841–850, https://doi.org/10.1109/ICDM50108.2020.00093.

[15] A. Gu, K. Goel, C. Ré, Efficiently modeling long sequences with structured state spaces. In: Proceedings of the 10th International Conference on Learning Representations, 2022.

[16] A. Gu, T. Dao, Mamba, Linear-time sequence modeling with selective state spaces, in. Proceedings of the First Conference on Language Modeling, 2024.

[17] T. Dao, A. Gu, Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality, in: Proceedings of the 41st International Conference on Machine Learning, Proc. Mach. Learn. Res. 235 (2024) 10041–10071.

[18] K. Cho, B. van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, in: Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014, pp. 103–111. 10.3115/v1/W14-4012.

[19] Y. Ren, Y. Yang, Y. Wang, Y. Liu, Dynamics of the global semiconductor trade and its dependencies, J. Geogr. Sci. 33 (6) (2023) 1141–1160, https://doi.org/10.1007/s11442-023-2123-9.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[21] R.E. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1) (1960) 35–45, https://doi.org/10.1115/1.3662552.

[22] C.-F. Chien, C.-Y. Hsu, P.-N. Chen, Semiconductor fault detection and classification for yield enhancement and manufacturing intelligence, Flex. Serv. Manuf. J. 25 (3) (2013) 367–388, https://doi.org/10.1007/s10696-012-9161-4.

[23] X. Wu, X. Qiu, Z. Li, Y. Wang, J. Hu, C. Guo, H. Xiong, B. Yang, Catch,, Channel-aware multivariate time series anomaly detection via frequency patching. In: Proceedings of the 13th International Conference on Learning Representations, 2025.

[24] Y. Nam, S. Yoon, Y. Shin, M. Bae, H. Song, J.-G. Lee, B.S. Lee, Breaking the time-frequency granularity discrepancy in time-series anomaly detection, in: In: Proceedings of the ACM Web Conference 2024, 2024, pp. 4204–4215, https://doi.org/10.1145/3589334.3645556.

[25] A. Borghesi, A. Bartolini, M. Lombardi, M. Milano, L. Benini, Anomaly detection using autoencoders in high performance computing systems, Proc. AAAI Conf. Artif. Intell. 33 (1) (2019) 9428–9433, https://doi.org/10.1609/aaai.v33i01.33019428.

[26] D.Y. Oh, I.D. Yun, Residual error based anomaly detection using auto-encoder in SMD machine sound, Sensors 18 (5) (2018) 1308, https://doi.org/10.3390/s18051308.

[27] Y.H. Park, I.D. Yun, Fast adaptive RNN encoder-decoder for anomaly detection in SMD assembly machine, Sensors 18 (10) (2018) 3573, https://doi.org/10.3390/s18103573.

[28] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.

[29] X. Zhu, Q. Ruan, S. Qian, M. Zhang, A hybrid model based on transformer and Mamba for enhanced sequence modeling, Sci. Rep. 15 (2025) 11428, https://doi.org/10.1038/s41598-025-87574-8.

[30] S. Liu, D. Yao, L. Fang, Z. Li, W. Li, K. Feng, X. Ji, J. Bi, AnomalyLLM: Few-shot anomaly edge detection for dynamic graphs using large language models, in: In: Proceedings of the 24th IEEE International Conference on Data Mining, 2024, pp. 785–790, https://doi.org/10.1109/ICDM59182.2024.00093.

[31] J. Liu, J. Guo, P.V. Orlik, M. Shibata, D. Nakahara, S. Mii, M. Takac, Anomaly detection in manufacturing systems using structured neural networks, in: In: Proceedings of the 13th World Congress on Intelligent Control and Automation, 2018, pp. 175–180, https://doi.org/10.1109/WCICA.2018.8630692.

[32] M. Tiezzi, M. Casoni, A. Betti, M. Gori, S. Melacci, State-space modeling in long sequence processing: a survey on recurrence in the transformer era, Neural Netw. 193 (2026) 108039, https://doi.org/10.1016/j.neunet.2025.108039.

[33] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: beyond efficient transformer for long sequence time-series forecasting, Proc. AAAI Conf. Artif. Intell. 35 (12) (2021) 11106–11115, https://doi.org/10.1609/aaai.v35i12.17325.

[34] D.R. Cote, S.V. Nguyen, A.K. Stamper, D.S. Armbrust, D. Tobben, R.A. Conti, G. Y. Lee, Plasma-assisted chemical vapor deposition of dielectric thin films for ULSI semiconductor circuits, IBM J. Res. Dev. 43 (1–2) (1999) 5–38, https://doi.org/10.1147/rd.431.0005.

[35] H. Rostami, J. Blue, C. Yugma, in: Equipment Condition Diagnosis and Fault Fingerprint Extraction in Semiconductor Manufacturing, in, 2016, pp. 534–539, https://doi.org/10.1109/ICMLA.2016.0094.

[36] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, Lof,, Identifying density-based local outliers, in, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104, https://doi.org/10.1145/335191.335388.

[37] J. Xu, H. Wu, J. Wang, M. Long, Anomaly Transformer: Time series anomaly detection with association discrepancy. In: Proceedings of the 10th International Conference on Learning Representations, 2022.