

ASTNet: Asynchronous Spatio-Temporal Network for Large-Scale Chemical Sensor Forecasting

Shihao Tu
Zhejiang University
shihao.tu@zju.edu.cn

Yang Yang*
Zhejiang University
yangya@zju.edu.cn

Wenyue Ding
SUPCON Technology
Co., Ltd.
dingwenyue@supcon.com

Yicheng Lu
Zhejiang University
yicheng.lu@zju.edu.cn

Qingkai Ren
Zhejiang University
qingkairen@zju.edu.cn

Yupeng Zhang
Zhejiang University
yuppzhang@zju.edu.cn

Yin Zhang
Zhejiang University
yinzhang@zju.edu.cn

Abstract

The chemical industry is faced with the urgent challenge of effectively harnessing the vast amounts of time-series data generated by thousands of sensors, which is essential for forecasting chemical states, achieving accurate real-time control of production processes. Traditional forecasting methods suffer from high computational latency and struggle with the complexity of spatiotemporal dependencies. As a result, modeling this data becomes challenging. This paper introduces a novel approach, referred to as *ASTNet*, designed to address these challenges. *ASTNet* integrates an asynchronous spatiotemporal modeling framework that combines temporal and spatial encoders, enabling concurrent learning of temporal and spatial dependencies while reducing computational latency. Additionally, it introduces a gated graph fusion mechanism that adaptively combines static (meta) and evolving (dynamic) sensor graphs, enhancing the handling of heterogeneous sensor data and spatial correlations. Extensive experiments on three real-world chemical sensor datasets demonstrate that *ASTNet* outperforms SOTA methods in terms of both prediction accuracy and computational efficiency, making *ASTNet* successfully deployed in chemical engineering industrial scenarios.

CCS Concepts

• **Computing methodologies** → **Temporal reasoning**; *Spatial and physical reasoning*.

Keywords

multivariate time series; forecasting; chemical process; deep learning

*Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
KDD '25, Toronto, ON, Canada

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-1454-2/2025/08
<https://doi.org/10.1145/3711896.3737194>

ACM Reference Format:

Shihao Tu, Yang Yang, Wenyue Ding, Yicheng Lu, Qingkai Ren, Yupeng Zhang, and Yin Zhang. 2025. *ASTNet: Asynchronous Spatio-Temporal Network for Large-Scale Chemical Sensor Forecasting*. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25)*, August 3–7, 2025, Toronto, ON, Canada. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3711896.3737194>

1 Introduction

Background. The development of industrial intelligence has enabled chemical factories to collect large-scale time series data through sensors, which facilitates the forecasting of chemical states in the production process. This capability is crucial for ensuring production safety, improving product quality, and achieving energy conservation and emission reduction in chemical engineering. The chemical reaction processes in production are lengthy and dynamic, exhibiting complex correlation relationships among the sensors, known as *spatial dependency*, as well as *long-term temporal dependency* of recorded time series, totally called spatiotemporal dependency. Manual forecasting requires significant expertise, and its imprecision can lead to product defects and production disruptions [35]. Moreover, chemical factories with thousands of sensors face high latency in manual predictions, making real-time forecasting impractical. Therefore, there is an urgent need to develop automated real-time spatiotemporal forecasting methods for chemical engineering.

Currently, spatiotemporal forecasting has evolved from statistical models to data-driven approaches, with early methods like ARIMA [22] and RNN [17] capturing temporal dependencies but facing efficiency limitations. Transformer-based models, such as Informer [45] and PatchTST [30], improve efficiency through attention mechanisms but overlook spatial correlations. To address this, methods like AGCRN [41], MTGNN [38], and StemGNN [42] learn time-invariant graphs, while MegaCRN [18], HimNet [8], PatchSTG [10], DUET [31], and Crossformer [44] focus on time-varying graphs. However, RNN-based models (AGCRN, MegaCRN, HimNet) suffer from inefficiency, and point-wise tokenization in StemGNN and MTGNN limits scalability. Patch-wise tokenization in Crossformer, PatchSTG and DUET enhances efficiency but also incurs quadratic complexity. Recent research focuses on reducing computational costs, such as STID [33], which improve computational efficiency but lack explicit graph modeling, while BigST [14]

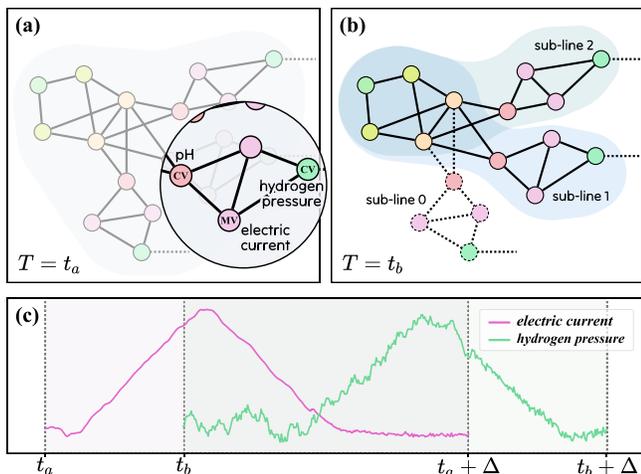


Figure 1: (a) The chemical production pipeline involves large-scale heterogeneous sensor variables (e.g., pH, temperature, etc.), which are further categorized into manipulated variables (MVs) and controlled variables (CVs). (b) The state of the production pipeline is time-varying. At time t_b , sub-line 0 is in a shutdown maintenance state. (c) Sensors exhibit time delay effects. An increase in the electric current (MV) at time t_a results in an increase in hydrogen production (CV) after a certain period.

and STWave [11] model predefined graphs via convolutional and spectral attention mechanisms.

However, when applied to real-world chemical sensor data, these methods often fail to perform effectively and efficiently due to several key factors: (1) The large number of sensors in chemical engineering pipelines makes modeling challenging, and there is strong heterogeneity between sensors, leading to significant physical differences in the recorded data, such as pH sensors and temperature sensors. (2) Sensor systems have potential sensor graph due to spatial dependency, including time-invariant (*meta*) graph and time-varying (*dynamic*) graph. Time-invariant graph reflects stable factors like physical locations and causal relationships, while time-varying graph evolve over time due to changing spatial dependencies. Both increase the complexity of modeling. (3) Due to the time-lag effects [32] in the time series of different sensors, long-term dependencies must be taken into account when modeling the data.

Challenges. To tackle these factors, an in-depth analysis of chemical production processes identifies two key challenges:

How to reduce computational latency in modeling spatiotemporal dependency?

First, traditional spatiotemporal modeling follows a sequential paradigm—first modeling temporal dependencies, then spatial dependencies—resulting in significant computational latency when meet large-scale sensors that hinders real-time prediction. This delay arises from two main factors: (1) The inefficiency of the sequential execution structure. (2) The high-dimensional representations from temporal modeling, which increase computational complexity in spatial modeling.

Second, industrial processes exhibit long-term temporal dependencies due to inherent lag effects and cumulative control feedback. As illustrated in Figure 1(c), increasing the electric current leads to a delayed change in hydrogen pressure. Accurately capturing such dependencies requires a long lookback window, significantly increasing computational costs, especially for attention-based models.

How to model complex spatial dependency among large-scale sensors?

First, in chemical engineering, the pipeline system can be modeled as a sensor graph, which encompasses both a meta graph and a dynamic graph. Meta graph is static and time-invariant. Manual construction of a meta graph is clearly unacceptable due to the large scale of sensors in a pipeline, and such graph are often biased or inaccurate [36]. Dynamic graph are time-varying due to adjustments in production tasks, equipment changes, or maintenance during the production process, which introduce dynamic spatial dependency between sensors. As shown in Figure 1(a,b), the topology of the sensor graph changes between time points t_a and t_b ; sub-line 0 stops operating, and some edges and nodes also disappear. Therefore, an important challenge is how to automatically model both the time-invariant and time-varying topologies of the sensor graph and effectively integrate them.

Second, the heterogeneity of data between sensors further complicates the modeling of relationships. For instance, in the chlor-alkali industry, sensors record values such as equipment electric current, reactant temperature, pH level, feed flow rate, and more, each with different measurement scales. As a result, enabling the model to recognize the sensor-specific properties of these varying sensors remains a challenge.

Solutions. To tackle the above challenges, in this paper, we propose a novel Asynchronous SpatioTemporal Network for large-scale chemical sensor forecasting, referred to as *ASTNet*.

For the first challenge, we aim to design a paradigm that enables temporal and spatial modeling to be computed as concurrently as possible. Additionally, the model needs to efficiently capture long-term temporal dependency. Traditional spatiotemporal models [19, 41, 44] typically follow a sequential paradigm, where temporal and spatial dependencies are modeled in sequence, leading to significant computational latency. This is because the temporal encoder (or spatial encoder) generates high-dimensional, lengthy representations, which are then used to model spatial (or temporal) dependencies. In contrast, *ASTNet* adopts an asynchronous spatiotemporal modeling paradigm. Specifically, sensor data is asynchronously fed into both the temporal and spatial encoders to model dependencies separately. After synchronization, the learned spatial dependencies are then used to enhance the temporal dependencies. This approach allows parallel computation of temporal and spatial modeling, significantly reducing latency. For efficient long-term temporal dependency modeling, *ASTNet* adopts patch-wise tokenization instead of point-wise tokenization, reducing computational overhead. Specifically, *ASTNet* uses different patch lengths for temporal and spatial modeling. Temporal modeling focuses on capturing subtle changes in the time series, thus using a shorter patch, while spatial modeling captures the coarser-grained trends in the time series, requiring longer patch. This approach effectively balances model capacity and computational latency. These

designs meet the low-latency needs of chemical engineering sensor predictions, providing timely and efficient feedback for real-time decisions.

For the second challenge, our goal is to ensure that the model effectively captures both the time-invariant sensor graph (*meta graph*) and the time-varying sensor graph (*dynamic graph*), and adaptively integrates them into a unified graph to enhance model robustness. Additionally, the model accounts for sensor-specific properties when modeling spatiotemporal dependencies. To accomplish this, *ASTNet* utilizes sensor-specific indicators to enrich both temporal and spatial representations, and employs a dynamic gating mechanism to adaptively integrate the time-invariant and time-varying sensor graphs. For the time-invariant sensor graph, *ASTNet* derives by modeling the relationships between sensor-specific indicators, which capture the inherent and static relationships among sensors based on their latent properties. For the time-varying sensor graph, *ASTNet* dynamically learns the spatial dependencies among sensors over time, reflecting the evolving relationships due to changes in process conditions, equipment states, and control adjustments. By combining these two sensor graphs through a gating mechanism, *ASTNet* can adaptively adjust the importance of each graph based on the current state of the sensors, ensuring that the model remains robust to both static and dynamic changes in the sensor graph.

In summary, the main contributions are as follows:

- We are the first to propose an asynchronous spatiotemporal modeling strategy for large-scale chemical sensor forecasting, addressing the critical challenge of computational latency in traditional sequential frameworks by enabling parallel temporal and spatial dependency learning, which is essential for real-time decision-making in chemical production.
- We design *ASTNet*, a novel dynamic graph fusion framework that integrates time-invariant meta graphs and time-varying dynamic graphs through a gated mechanism. This approach adaptively balances heterogeneous sensor correlations while reducing erroneous spatial dependencies, significantly enhancing robustness in complex industrial environments.
- We conduct extensive experiments on real-world chemical sensor datasets involving thousands of heterogeneous sensors. Quantitative results demonstrate *ASTNet*'s superior prediction accuracy and efficiency over SOTA baselines, with the MAE improved by 7.4% and the MAPE improved by 7.0% compared to the best baseline. *ASTNet* has been deployed in reality for prediction and management of sensor data in chemical plants.

2 Related works

2.1 General Time Series Modeling

Time series modeling has recently become a key research area in both industry and academia. Early statistical methods, such as ARIMA [22] and ETS [13], have limited modeling ability. To address these shortcomings, numerous deep neural network models have been proposed. RNN [17] and LSTM [16] capture long/short-term temporal dependencies but suffer from high computational latency due to sequential execution. PatchTST [30] reduces computation latency and captures temporal semantics through patch-wise tokenization. These models achieve promising results in long-term

forecasting but struggle to capture spatial dependencies in large-scale sensor data. To model it, Informer [45] and Autoformer [37] integrate data from diverse variables during the extraction of time series embedding representations. iTransformer [28] employs a self-attention module across variables, treating independent time series as tokens and leveraging the self-attention mechanism to capture inter-variate correlations. In contrast, TSMixer [6] utilizes an MLP module for inter-variable interactions, capturing intricate variable correlations through multi-level features extracted via fully connected layers. Crossformer [44] utilizes a cross-attention mechanism that captures both local and global temporal patterns while modeling spatial dependencies simultaneously. CCM [5] deploys a cluster-aware feedforward mechanism, enabling customized management and processing for each individual cluster. Similarly, DUET [31] uses a dual-encoder framework to separately capture spatiotemporal dependency. DWLR [24] proposed a method to improve the temporal generalization ability in the case of label shift.

2.2 Spatiotemporal Modeling

Spatiotemporal forecasting has been extensive research as it plays a key role in many real-world applications [8, 25, 39, 41]. Graph Neural Networks (GNNs) [4, 9, 21, 40, 43] address this gap by modeling spatial structures as graphs, offering a novel paradigm for spatiotemporal analysis. In recent years, the first category employs predefined graph structures combined with sequential modeling, exemplified by DCRNN [25], GWNet [39], and DGCRN [23], which leverage a prior graph to establish static spatial dependencies. The second paradigm introduces automated time-invariant graph learning, where methods like MTGNN [38], AGCRN [41], and StemGNN [3] automatically infer time-invariant graph from data. The third category addresses dynamic spatial dependency through modeling time-varying graph in models such as HimNet [8], MegaCRN [19], and DMSTGCN [26]. However, these methods often incur computational overhead. Recent efficiency-oriented approaches adopt two strategies: linear/low-rank approximations (Lastjomer [12], BigST [15], HIEST [29]) that trade spatial expressiveness for speed, and linear-based architectures (STID [33], SimST [27]) that preserve spatial dependency through learnable parameters. Moreover, PatchSTG [10] advances efficiency through irregular spatial patching.

However, these methods, mainly designed for traffic, weather, and electricity forecasting, often underperform with real-world chemical sensor data. They suffer from high computational latency when handling large-scale sensors and long-term dependencies. Additionally, spatiotemporal methods typically rely on short look-back windows for short-term predictions (e.g., 12 \rightarrow 12), which cannot capture the long-term temporal dependencies in chemical data. Furthermore, chemical sensors are highly heterogeneous, and the sensor graph complexity exceeds that of public datasets.

3 Problem Formulation

Representation of Spatiotemporal Data. The spatiotemporal data can be denoted as a tensor $\mathbf{X} \in \mathbb{R}^{C \times T}$, where C represents the number of sensors and T denotes the timestamps. Each entry $X_{c,t}$ of the tensor \mathbf{X} corresponds to the c -th sensor at the t -th timestamp.

For instance, in the context of chemical sensors prediction, the tensor \mathbf{X} might store data such as pH, temperature, current magnitude, etc., recorded within fixed time intervals (e.g., every 5 seconds).

Spatiotemporal Forecasting. Spatiotemporal prediction focuses on estimating future states $\mathbf{X}_{t_{K+1}:t_{K+H}}$ from historical observations $\mathbf{X}_{t_{K-L+1}:t_K}$. To simplify the notation, we refer to the predicted future states as $\mathbf{y}_{horizon}$ and the historical input data as $\mathbf{x}_{lookback}$, both of them are timeslices. In this context, the goal is to predict H future timestamps by analyzing a sequence of L past timestamps. As such, this process can be formally expressed as $\mathbf{y}_{horizon} = z(\mathbf{x}_{lookback})$, where $z(\cdot)$ represents the predictive function.

4 Methodology

The *ASTNet* framework (as shown in Figure 2) is specifically designed to efficiently comprehensively model the complex spatiotemporal dependencies in large-scale chemical sensor data.

Overview. To simplify the pipeline, we define a pair of lookback and horizon windows (two timeslices): $\mathbf{x} \in \mathbb{R}^{C \times L}$ and $\mathbf{y} \in \mathbb{R}^{C \times H}$. Here, L and H denote the number of timestamps in the lookback and horizon windows, respectively, and C represents the number of sensors. The operations described for this pair are consistently applied to all other window pairs in the dataset.

ASTNet begins with the spatiotemporal embedding process, where the time series data \mathbf{x} , with two different patch lengths ($P_t < P_s$), undergoes re-normalization, tokenization, and context enrichment. This step generates two latent representations: fine-grained temporal embedding $\mathbf{h}_t \in \mathbb{R}^{C \times N_t \times d}$ and coarse-grained spatial embedding $\mathbf{h}_s \in \mathbb{R}^{C \times N_s \times d}$, where $N_t > N_s$. Next, the spatial embedding \mathbf{h}_s is passed through a lightweight temporal encoder to produce an improved spatial representation $\tilde{\mathbf{h}}_s$. Both \mathbf{h}_t and $\tilde{\mathbf{h}}_s$ are then asynchronously fed into the temporal and spatial encoders, respectively. Importantly, these encoders can operate in parallel. Once synchronized, the spatial encoder generates a dynamic graph \mathbf{A}^l and a refined spatial representation $\tilde{\mathbf{h}}_s^{l+1}$. The dynamic graph \mathbf{A}^l is then fused with the meta graph \mathbf{A}_{meta} (using a sensor-specific indicator) via a gating mechanism, producing a unified graph \mathbf{A}^l . Simultaneously, the temporal encoder outputs an updated temporal representation $\tilde{\mathbf{h}}_t^l$, which is further refined by spatial dependencies through the unified graph \mathbf{A}^l . The resulting embeddings \mathbf{h}_t^{l+1} and $\tilde{\mathbf{h}}_s^{l+1}$ are then passed to the next layer of the asynchronous encoder. Finally, a projection head maps the output of the last layer, \mathbf{h}_t^{L-1} , to $\mathbf{y}_{horizon}$, and the objective function $|\mathbf{y}_{horizon} - \hat{\mathbf{y}}_{horizon}|$ is used to optimize the model.

In the following sections, we will delve into the details of each component, starting with the spatiotemporal embedding process, followed by the temporal and spatial modeling strategies, and concluding with the asynchronous fusion that integrates these elements into a cohesive framework.

4.1 Spatiotemporal Embedding

Re-Normalization. Sensor heterogeneity and time series non-stationarity give rise to the distribution shift problem, which manifests as instability in the mean and variance of data over time [2]. To address this issue, each time series instance $\mathbf{x} \in \mathbb{R}^{C \times L}$ is normalized to \mathbf{x}_{norm} [20]. After generating predictions, a re-normalization step

reintroduces the original mean and standard deviation to restore the non-stationary components.

Tokenization. Tokenizing time series at a point-wise level not only fails to effectively capture meaningful patterns [34] but also significantly increases computational complexity [30]. To address this, this paper adopts a patch-wise tokenization strategy [30]. Specifically, we partition \mathbf{x}_{norm} into multiple patches of length P with stride S (the gap between adjacent patches). The resulting patch sequence $\mathbf{x}_{norm}^p \in \mathbb{R}^{C \times N \times P}$, where $N = \lfloor \frac{L-P}{S} \rfloor + 2$. To ensure sequence completeness, the original series is padded (last value repeated S times) before partitioning. Subsequently, a linear projection then maps each patch to its latent representation:

$$\mathbf{z} = \text{Projection}(\mathbf{x}_{norm}^p) \in \mathbb{R}^{C \times N \times d} \quad (1)$$

where d is the dimension of token embedding.

Context Incorporation. Heterogeneous large-scale sensors, such as pH and current sensors, are common in chemical production. By embedding sensor-specific indicators (learnable parameters) into the spatiotemporal representation, their heterogeneity is effectively captured. Specifically, a learnable parameter $\mathbf{E}_{tag} \in \mathbb{R}^{C \times d}$ is assigned to each sensor, and positional encodings $\mathbf{E}_{pos} \in \mathbb{R}^{N \times d}$ are incorporated. Finally, the enhanced latent representation is obtained as follows:

$$\mathbf{h} = \text{Concatenate}(\mathbf{E}_{tag}, \mathbf{z} + \mathbf{E}_{pos}) \in \mathbb{R}^{C \times (N+1) \times d} \quad (2)$$

4.2 Transformer Backbone

Given token embeddings $\mathbf{h} \in \mathbb{R}^{C \times N \times d}$, we adopt a standard Transformer Encoder to obtain per-token representations aggregated from all tokens. We apply LayerNorm [1] after attention and feedforward layers to improve training stability. The self-attention mechanism is defined as follows:

$$\mathcal{A}_{ij} = \mathbf{h}_i^T \mathbf{W}_q \mathbf{W}_k^T \mathbf{h}_j \quad (3)$$

$$\text{Attention}(\mathbf{h}) = \text{Softmax}\left(\frac{\mathcal{A}}{\sqrt{d}}\right) \mathbf{h} \mathbf{W}_v \quad (4)$$

where $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v \in \mathbb{R}^{d \times d}$ project token embeddings \mathbf{h} into d -dimensional queries, keys, and values. The Transformer backbone consists of multiple layers of this attention mechanism followed by a feedforward network (FFN), which enhances token-wise representations before passing them to the next layer. By permuting \mathbf{h} , the attention mechanism can be applied separately in the temporal and spatial dimensions.

4.3 Temporal Modeling

The temporal characteristics of time series data in chemical engineering involve long-term dependencies and nonlinear dynamics. Long-term dependencies arise from lag effects and the cumulative impact of control feedback loops, where past operations influence key variables' dynamics. Nonlinear dynamics are reflected in periodic fluctuations, non-stationary behavior, and complex trends throughout different stages. To model this, a Transformer is employed. Specifically, a small P_t is used in the spatiotemporal embedding process to obtain a fine-grained embedding $\mathbf{h}_t^l \in \mathbb{R}^{C \times N_t \times d}$. Then, a Transformer encoder is applied to generate more informative embeddings. The operation for the l -th layer is formally

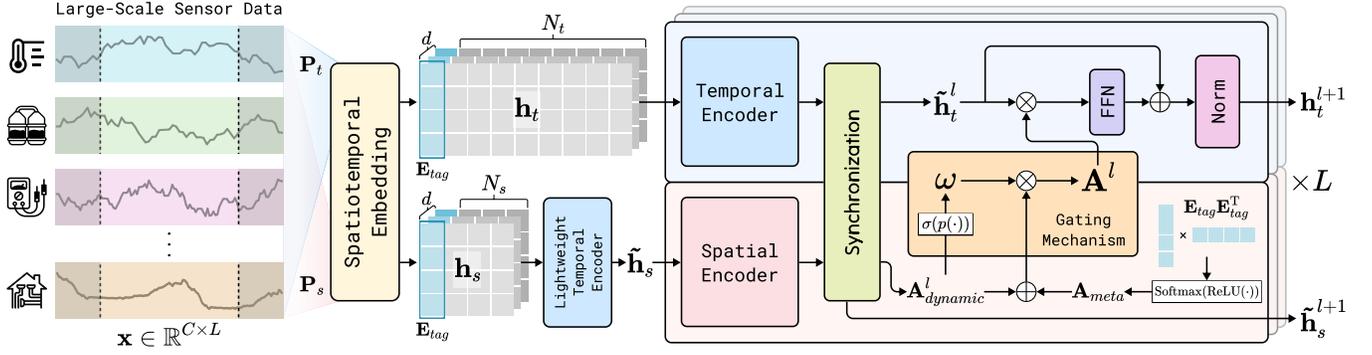


Figure 2: ASTNet Framework

defined as follows:

$$\tilde{h}_t^l = \text{TransformerEncoder}(h_t^l) \quad (5)$$

4.4 Spatial Modeling

In chemical engineering, pipeline systems are modeled using a sensor network based on static, time-invariant physical equipment and sensor relationships. However, manually constructing a prior graph for each plant is impractical due to the large number of sensors, and predefined graphs can be biased or inaccurate. Dynamic changes in production tasks, equipment, or maintenance introduce topological variations, making it crucial to automatically model both time-invariant and time-varying topologies.

To address this, *ASTNet* proposes a spatial modeling approach that incorporates both time-invariant and time-varying sensor graphs. Specifically, the time-invariant sensor graph, referred to as the meta graph, is derived from E_{tag} . The dynamic graph between sensors is modeled using an attention mechanism across the spatial dimension. Finally, the latent sensor graph is obtained by applying gated graph fusion to combine the meta graph and the dynamic graph. The detailed components are outlined as follows:

Lightweight Temporal Encoder. To reduce computational overhead, *ASTNet* introduces a lightweight temporal encoder to efficiently model the temporal data of sensors. Specifically, we use a larger P_s in the spatiotemporal embedding process to obtain the coarse-grained temporal embedding $h_s \in \mathbb{R}^{C \times N_s \times d}$, we model this input using another Transformer:

$$\tilde{h}_s = \text{TransformerEncoder}(h_s) \quad (6)$$

Meta Graph. The meta graph constructed from sensor embeddings E_{tag} . This meta graph is derived by following formulations:

$$A_{meta} = \text{Softmax}(\text{ReLU}(E_{tag} E_{tag}^T)) \quad (7)$$

The meta graph A_{meta} is computed by taking the dot product of E_{tag} , applying the $\text{ReLU}(\cdot)$ function, and then normalizing it with a $\text{Softmax}(\cdot)$ operation. Each element (i, j) in A_{meta} represents the correlation strength between the i -th and j -th sensor embeddings.

Dynamic Graph. Spatiotemporal dependencies between sensors evolve due to process condition adjustments, equipment state fluctuations, and control strategy interventions. To model this dynamic, a Transformer adaptively learns the correlation strengths across the spatial dimension, transforming \tilde{h}_s^l into \tilde{h}_s^{l+1} , which is fed into

the next layer as input. This process constructs a dynamic adjacency matrix, $A_{dynamic}^l$, by aggregating the attention weights. The mechanism is formally defined as:

$$A_{dynamic}^l, \tilde{h}_s^{l+1} = \text{TransformerEncoder}(\tilde{h}_s^l) \quad (8)$$

Gated Graph Fusion. In the operation of chemical engineering pipelines, at certain moments, some sensors may not exhibit significant correlations. However, in existing modeling approaches, both the static correlation matrix A_{meta} and the dynamic correlation matrix $A_{dynamic}$ contain non-zero values, which may introduce erroneous correlations when capturing spatial dependencies, thereby reducing prediction accuracy.

To address this issue, this work proposes a gating-mechanism-based method for correcting spatial dependencies. Specifically, we first utilize $A_{dynamic}$ to determine the state of each sensor and generate a gating matrix ω to adaptively adjust the final spatial dependency matrix A^l . This method is implemented through the following formulas:

$$\omega = \text{Sigmoid}(p(A_{dynamic}^l)) \in \mathbb{R}^C \quad (9)$$

$$A^l = \omega \cdot (A_{meta} + A_{dynamic}^l) \quad (10)$$

Here, $p(\cdot)$ is a learnable linear mapping function that extracts sensor state information from $A_{dynamic}$, the $\text{Sigmoid}(\cdot)$ function maps this state information to the interval $(0, 1)$, generating gating weights. The element-wise multiplication of ω with $A_{meta} + A_{dynamic}$ effectively suppresses irrelevant sensor correlations, thereby enhancing the accuracy of spatial dependency modeling.

4.5 Asynchronous Fusion

Traditional fusion techniques often employ a sequential paradigm, prioritizing temporal dependency modeling followed by spatial dependency [10, 41, 44]. Such a paradigm tends to be inefficient, especially in cases with numerous sensors or stringent real-time forecasting demands, which cannot align with the application scenario.

To address this, an asynchronous fusion paradigm is proposed to concurrently integrate temporal and spatial features, enabling parallel computation and reducing computational latency, making it suitable for real-time forecasting in large-scale chemical sensors.

Table 1: Dataset statistics.

| Datasets | #Sensors | #Timestamps | #TimeSlices | Timespan |
|----------|----------|-------------|-------------|-------------------|
| A | 1113 | 7102078 | 284083 | 20230601-20240715 |
| B | 1557 | 20165630 | 161325 | 20230114-20240103 |
| C | 2377 | 19178805 | 153430 | 20240107-20240822 |

Specifically, we design a multi-layer composite encoder $z(\cdot)$, comprising a temporal encoder $f(\cdot)$ and a spatial encoder $g(\cdot)$. The model asynchronously yet cooperatively extracts temporal and spatial features, balancing accuracy and efficiency. At each layer l , $f(\cdot)$ and $g(\cdot)$ synchronize with each other during parallel computation. Afterward, they generate updated representations for the next layer. The total number of layers is denoted by L . The operations are defined as follows:

$$\mathbf{h}_t^{l+1}, \tilde{\mathbf{h}}_s^{l+1} = z(f(\mathbf{h}_t^l), g(\tilde{\mathbf{h}}_s^l)) \quad (11)$$

$$\tilde{\mathbf{h}}_t^l = f(\mathbf{h}_t^l) \quad (12)$$

$$\tilde{\mathbf{h}}_s^{l+1}, \mathbf{A}^l = g(\tilde{\mathbf{h}}_s^l) \quad (13)$$

$$\mathbf{h}_t^{l+1} = \text{Norm}(\text{FFN}(\mathbf{A}^l \tilde{\mathbf{h}}_t^l) + \tilde{\mathbf{h}}_t^l) \quad (14)$$

The outputs \mathbf{h}_t^{l+1} and $\tilde{\mathbf{h}}_s^{l+1}$ from each layer serve as inputs to the next, progressively refining spatiotemporal representations.

5 Experiments

The goal of this section is to address the following four pivotal research questions by conducting comprehensive experiments on three large-scale chemical sensor data.

- **RQ1:** How does *ASTNet* perform when compared to current approaches in large-scale chemical sensor forecasting?
- **RQ2:** What contributions do the main components of *ASTNet*?
- **RQ3:** How efficient is *ASTNet* in large-scale datasets?
- **RQ4:** How do the essential hyper-parameters impact *ASTNet*?

6 Datasets

Data Collection. To validate the effectiveness of *ASTNet*, we conducted extensive experiments on sensor data from three large-scale chemical engineering production lines, which originate from real industrial production environments. The datasets are provided by SUPCON¹, a leading industrial automation and control technology company, with permission for usage granted by the collaborators. These three datasets are from typical scenarios in the chemical industry, including chlor-alkali, petroleum, and coal chemical industries. Each chemical plant has over 1000 sensors, with a data sampling frequency of 5 seconds, which ensures adequate capture of dynamic changes in the chemical processes. Table 1 presents the final statistical information of these three datasets.

Preprocessing. The data preprocessing involves handling missing and anomalous values. For example, missing values caused by sub-production line suspensions or equipment maintenance were addressed by zero-filling, as this approach was deemed appropriate after expert evaluation, aligning with the physical meaning and effectively avoiding any negative impact on model training. On the other hand, for missing values due to sensor failures or data

transmission interruptions, linear interpolation was used for imputation. Moreover, SUPCON’s professional data team screened for anomalous sequences in the data based on statistical methods and domain knowledge, eliminating outliers caused by equipment failures or measurement errors, and therefore ensuring data quality and validity. Finally, sensor data in chemical processes often have different units and ranges (e.g., temperature in °C and pressure in MPa), so the data was standardized to prevent certain parameters from overly influencing model training.

Dataset Split. To ensure the rigor of the experiments, each dataset was strictly divided in chronological order, with a 6:2:2 ratio for training, validation, and test sets. Specifically, the first 60% of the data was used for training, the next 20% for validation, and the final 20% for testing. This division effectively prevents model evaluation bias due to data leakage. Given that chemical processes involve sparse sequential data and that model training is costly, a sliding window method with a window size of 25 steps was used to obtain samples from each time slice, enabling the model to process a certain length of historical data. The lookback window length for each time slice was set to 256, which helps the model capture long-term dependencies.

6.1 Experiment Setup

Baselines. In this paper, we perform a comprehensive comparison of 11 SOTAs with our proposed model, *ASTNet*. These baselines are systematically categorized into three distinct groups based on their underlying modeling approaches: **(a)** Non-spatial modeling-based methods: This group includes models that exclusively focus on modeling temporal dependencies for forecasting, namely PatchTST [30] and PDF [7]. **(b)** Time-invariant spatial-based methods: These models automatically learn time-invariant sensor graphs and incorporate temporal modeling to address spatiotemporal forecasting tasks. Specifically, we evaluate STID [33], AGCRN [41], MTGNN [38], and StemGNN [3]. **(c)** Time-varying spatial-based methods: Models in this category dynamically capture spatial dependencies across varying time periods for spatiotemporal forecasting. The baselines considered include MegaCRN [19], HimNet [8], PatchSTG [10], Crossformer [44], and DUET [31]. See Appendix B for a detailed description of these baselines.

Evaluation Metrics. To evaluate the model comprehensively, we consider both performance and efficiency metrics. In terms of performance, we use three common indicators: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics help assess the model’s accuracy while accounting for potential sensor noise and abrupt fluctuations in the chemical sensor data. Please refer to Appendix C for the detailed formula of the above evaluation metrics.

Implementation Details. During training, *ASTNet* is optimized using the Adam optimizer with a learning rate of 0.0003, and the training epochs is set to 40 for all datasets, with early stopping patience set to 5. The cosine learning rate scheduler is used. The batch size is set to 8 for Dataset A and to 4 for Datasets B and C due to the large-scale sensors. Each experiment is repeated 5 times, with metric average results reported. The default hyper-parameters of *ASTNet* for datasets A, B, and C are as follows: lookback window

¹<https://global.supcon.com/>

Table 2: Large-scale chemical sensor forecasting performance comparison of our *ASTNet* and baselines.

| Datasets | Methods | Horizon 60 | | | Horizon 120 | | | Horizon 360 | | | Average | | |
|---------------|--------------|--------------|---------------|---------------|--------------|---------------|---------------|---------------|---------------|---------------|--------------|---------------|---------------|
| | | MAE | RMSE | MAPE (%) | MAE | RMSE | MAPE (%) | MAE | RMSE | MAPE (%) | MAE | RMSE | MAPE (%) |
| A | PatchTST | 0.256 | 8.882 | 57.215 | 0.264 | 11.750 | 51.277 | 0.322 | 12.639 | 65.049 | 0.281 | 11.090 | 57.847 |
| | PDF | 0.243 | 9.143 | 47.903 | 0.271 | 11.092 | 51.257 | 0.352 | 13.603 | 67.232 | 0.289 | 11.280 | 55.464 |
| | STID | 0.215 | 9.111 | 41.730 | 0.247 | 8.971 | 45.290 | 0.331 | 11.346 | 54.650 | 0.264 | 9.809 | 47.223 |
| | AGCRN | 0.232 | 8.295 | 44.910 | 0.260 | 9.499 | 49.130 | 0.313 | 11.962 | 56.770 | 0.268 | 9.919 | 50.270 |
| | MTGNN | 0.237 | 8.564 | 47.490 | 0.261 | 9.467 | 50.740 | 0.319 | 12.106 | 59.210 | 0.272 | 10.046 | 52.480 |
| | StemGNN | 0.203 | 4.092 | 41.350 | 0.230 | 4.913 | 46.100 | 0.311 | 10.731 | 55.540 | 0.248 | 6.578 | 47.663 |
| | MegaCRN | 0.182 | 6.960 | 38.380 | 0.218 | 8.249 | 43.910 | 0.315 | 11.958 | 56.500 | 0.238 | 9.056 | 46.263 |
| | HimNet | 0.202 | 8.046 | 38.790 | 0.220 | 8.372 | 40.250 | 0.294 | 11.748 | 56.728 | 0.238 | 9.389 | 45.256 |
| | Crossformer | <u>0.162</u> | 7.678 | <u>31.460</u> | <u>0.191</u> | 8.683 | <u>35.500</u> | <u>0.276</u> | 11.189 | <u>48.290</u> | <u>0.210</u> | 9.184 | <u>38.417</u> |
| | DUET | 0.191 | 7.462 | 38.710 | 0.222 | 8.138 | 42.000 | 0.287 | <u>10.853</u> | 50.690 | 0.233 | 8.817 | 43.800 |
| | PatchSTG | 0.184 | 7.762 | 33.170 | 0.217 | 8.307 | 37.920 | 0.297 | 11.092 | 49.220 | 0.233 | 9.053 | 40.103 |
| <i>ASTNet</i> | 0.150 | 6.603 | 30.260 | 0.181 | 8.003 | 34.710 | 0.257 | 11.175 | 46.010 | 0.196 | 8.594 | 36.993 | |
| B | PatchTST | 0.182 | 2.446 | 78.886 | 0.274 | 13.771 | 115.773 | 0.342 | 19.905 | 92.995 | 0.266 | 12.041 | 95.885 |
| | PDF | 0.193 | <u>2.125</u> | 73.159 | 0.227 | 10.750 | 90.045 | 0.315 | 16.539 | 92.054 | 0.245 | <u>9.805</u> | 85.086 |
| | STID | 0.178 | 1.957 | 70.930 | 0.211 | 9.746 | 75.300 | 0.245 | 13.891 | 84.270 | 0.212 | 8.531 | 76.833 |
| | AGCRN | 0.197 | 5.767 | 105.430 | 0.216 | 11.401 | 108.910 | OOM | OOM | OOM | - | - | - |
| | MTGNN | 0.192 | 9.472 | 81.770 | 0.210 | 13.753 | 85.350 | OOM | OOM | OOM | - | - | - |
| | StemGNN | 0.169 | 8.659 | 67.900 | 0.194 | 13.241 | 74.100 | OOM | OOM | OOM | - | - | - |
| | MegaCRN | 0.161 | 11.883 | 38.350 | 0.194 | 9.600 | 74.330 | OOM | OOM | OOM | - | - | - |
| | HimNet | 0.172 | 9.048 | 67.640 | 0.191 | 11.638 | 71.850 | 0.229 | 14.404 | 81.430 | 0.197 | 11.697 | 73.640 |
| | Crossformer | <u>0.160</u> | 6.979 | 64.070 | <u>0.179</u> | <u>9.699</u> | <u>68.280</u> | <u>0.221</u> | 13.144 | <u>75.050</u> | <u>0.187</u> | 9.941 | <u>69.133</u> |
| | DUET | 0.168 | 7.739 | 68.830 | 0.187 | 11.202 | 72.110 | 0.227 | <u>13.485</u> | 80.210 | 0.194 | 10.809 | 73.717 |
| | PatchSTG | 0.175 | 7.599 | 68.720 | 0.198 | 10.814 | 72.990 | 0.236 | 14.399 | 80.790 | 0.203 | 10.937 | 74.167 |
| <i>ASTNet</i> | 0.153 | 7.446 | <u>57.810</u> | 0.176 | 10.068 | 62.850 | 0.221 | 13.690 | 72.460 | 0.183 | 10.401 | 64.373 | |
| C | PatchTST | 0.095 | <u>4.622</u> | 32.039 | 0.141 | <u>5.536</u> | 42.203 | 0.284 | 32.390 | 50.638 | 0.173 | 14.183 | 41.627 |
| | PDF | 0.095 | 4.975 | 30.662 | 0.120 | 5.860 | 36.822 | 0.306 | 31.708 | 60.702 | 0.174 | <u>14.181</u> | 42.729 |
| | STID | 0.093 | 4.355 | 28.940 | 0.119 | 5.465 | 32.600 | 0.275 | 29.066 | 50.590 | 0.162 | 12.962 | 37.377 |
| | AGCRN | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | - | - | - |
| | MTGNN | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | - | - | - |
| | StemGNN | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | - | - | - |
| | MegaCRN | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | OOM | - | - | - |
| | HimNet | 0.082 | 12.807 | 26.000 | <u>0.101</u> | 19.107 | 29.740 | 0.212 | 35.721 | 39.780 | 0.132 | 22.545 | 31.840 |
| | Crossformer | <u>0.070</u> | 9.174 | <u>23.230</u> | 0.115 | 20.563 | 26.910 | 0.198 | 32.003 | 35.540 | <u>0.128</u> | 20.580 | <u>28.560</u> |
| | DUET | 0.079 | 10.243 | 26.890 | 0.111 | 17.758 | 30.800 | 0.208 | 31.126 | 39.370 | 0.132 | 19.709 | 32.353 |
| | PatchSTG | 0.086 | 12.639 | 27.340 | 0.119 | 19.212 | 31.490 | 0.210 | 31.307 | 40.140 | 0.139 | 21.053 | 32.990 |
| <i>ASTNet</i> | 0.056 | 8.073 | 20.520 | 0.092 | 17.044 | 24.080 | 0.182 | <u>30.109</u> | 32.080 | 0.110 | 18.408 | 25.560 | |

length, embedding dimension (d), number of heads, and feedforward network dimension are set to 256, 128, 4, and 512, respectively. Additionally, $P_t = 16$ and $P_s = 64$ are used for all datasets, and the number of spatiotemporal layers is set to 2. All experiments are implemented in PyTorch using 8 NVIDIA RTX 3090 24GB GPUs.

6.2 Performance Comparisons (RQ1)

Table 2 presents the MAE, RMSE, and MAPE for chemical sensor forecasting across all methods on three real-world, large-scale chemical sensor datasets. The performance is reported for horizons of 60 (5 min), 120 (10 min), and 360 (30 min), as well as the average performance across all horizons. **Bold text** indicates the optimal metric, while underlined text denotes the second-best metric. "OOM" means the method ran out of memory, and the corresponding empty average result is denoted as "-". From the results in Table 2, our framework *ASTNet* outperforms the baseline methods, having an average 7.4% MAE and 7.0% MAPE improvement over

the best-competing methods on all datasets and various horizons, which indicates the superiority of *ASTNet*. The key reason behind the performance improvement can be attributed following reasons: **(a)** STID and HimNet effectively utilize sensor-specific indicators to address sensor heterogeneity, surpassing models like PatchTST and PDF that do not use this strategy. *ASTNet* introduces sensor indicators to handle sensor heterogeneity, aiming to simultaneously enhance the modeling of both temporal and spatial dependencies. **(b)** MTGNN and AGCRN only consider time-invariant graph, while MegaCRN and HimNet incorporate time-varying graph, achieving better results. *ASTNet* models both time-invariant and dynamic graphs through modeling meta graph and dynamic graph, effectively capturing the dynamic relationships between sensors in chemical engineering processes, while also demonstrating better generalization ability to capture invariant relationships. **(c)** Chemical engineering pipelines are mostly in a stable operation state, where sensor interference is typically absent. *ASTNet* learns a meta

graph using sensor-specific indicators, models dynamic graphs using attention mechanisms, and finally uses a gating mechanism to integrate graph structures. This allows the model to adaptively choose whether or not to consider the sensor graph structure, which is other models cannot achieve.

Table 3: Average MAE of the ablated variants of *ASTNet*.

| Model | A ↓ | B ↓ | C ↓ |
|-------------------|---------------|---------------|---------------|
| w/o A^l | 0.2382 | 0.2582 | 0.1334 |
| w/o ω | 0.2228 | 0.2138 | 0.1204 |
| w/o A_{meta} | 0.2089 | 0.2094 | 0.1174 |
| w/o $A_{dynamic}$ | 0.2306 | 0.2345 | 0.1353 |
| w/o E_{tag} | 0.2134 | 0.1956 | 0.1282 |
| <i>ASTNet</i> | 0.1957 | 0.1833 | 0.1101 |

6.3 Ablation Study (RQ2)

This section presents ablation studies to validate the key components of the proposed approach by analyzing five model variants: (1) **w/o A^l** : removing spatial dependency modeling to assess its impact. (2) **w/o ω** : replacing the gated mechanism with a static vector to test adaptive graph integration. (3) **w/o A_{meta}** : removing the meta graph to evaluate time-invariant spatial dependencies. (4) **w/o $A_{dynamic}$** : removing the dynamic graph and gated fusion to test time-varying spatial dependencies. (5) **w/o E_{tag}** : removing the learnable sensor-specific indicator, limiting the model’s ability to capture sensor heterogeneity.

As shown in Table 3, removing spatial modeling (**w/o A^l**) leads to a significant drop in prediction accuracy, highlighting the importance of capturing spatial dependencies. The **w/o ω** variant shows moderate performance degradation, indicating that static graph integration fails to adapt effectively to dynamic sensor conditions. The absence of the meta graph (**w/o A_{meta}**) results in further performance decline, suggesting that time-invariant spatial relationships play a crucial role in maintaining robustness. The **w/o $A_{dynamic}$** variant performs the worst, confirming that time-varying spatial dependencies are essential in dynamically evolving chemical production processes. Finally, the removal of sensor-specific indicators (**w/o E_{tag}**) degrades model performance, emphasizing their necessity in addressing sensor heterogeneity.

Table 4: Efficiency Comparison.

| Model | #Params | Cost Time | Mem Usage |
|-------------------------|------------|----------------|-----------|
| STID | 72.98K | 6.92ms | 1209.84MB |
| AGCRN | 762.76K | - | OOM |
| MegaCRN | 420.48K | 3261.41ms | 1211.17MB |
| HimNet | 1232.90K | 1951.72ms | 1502.26MB |
| StemGNN | 482870.90K | 274.24ms | 3054.38MB |
| MTGNN | 49008.04K | 624.83ms | 1398.85MB |
| Crossformer | 16127.52K | 104.16ms | 1286.50MB |
| PatchSTG | 4506.27K | 153.85ms | 1402.52MB |
| DUET | 7571.80K | 121.49ms | 1270.95MB |
| <i>ASTNet</i> w/o Async | 1604.55K | 37.59ms | 1248.43MB |
| <i>ASTNet</i> | 1604.55K | 26.49ms | 1248.43MB |

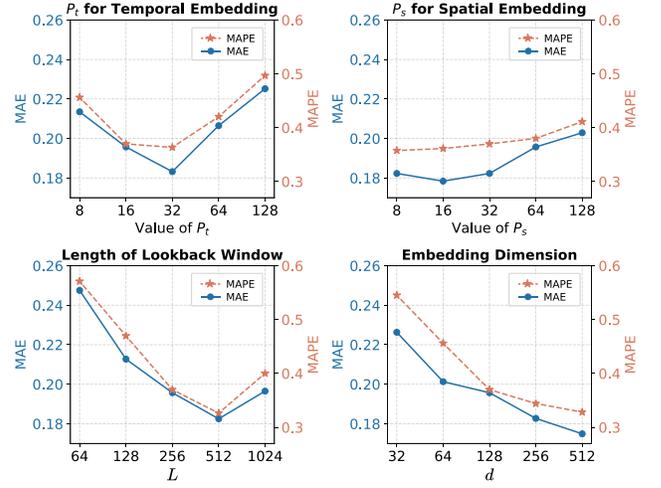


Figure 3: Hyperparameter Study of *ASTNet*

6.4 Efficiency Comparisons (RQ3)

The computational latency of *ASTNet* is evaluated against nine spatiotemporal baseline models in a large-scale sensor scenario with 1,000 sensors and a 256-length lookback window. To ensure accurate measurement, GPUs are pre-warmed using complex pre-computation tasks (e.g., large-scale matrix multiplication) before running each model and each test is repeated 5 times, with average results reported. Table 4 provides a comprehensive analysis by reporting the number of parameters (#Params), per-timeslice runtime (t_{cost}), and GPU memory usage (Mem usage) per timeslice. As shown in the table, STID performs well due to the high computational efficiency of its linear projection mechanism. In contrast, models like AGCRN, MegaCRN, and HimNet, which rely on the GCRU architecture, face significant efficiency gaps due to the iterative nature of RNNs, leading to high latency. Notably, AGCRN experiences an out-of-memory issue in this setting. Similarly, models with complex architectures and point-wise tokenization, such as StemGNN and MTGNN, suffer from high GPU memory usage and latency due to their large number of learnable parameters. On the other hand, models utilizing patch-wise tokenization and parallel computation, such as Crossformer, PatchSTG, and DUET, achieve competitive performance with lower computational latency and memory usage. However, PatchSTG incurs extra overhead due to its complex irregular spatial partitioning, while DUET faces additional costs from its dual clustering mechanism. Both models also rely on a sequential paradigm for spatiotemporal dependency modeling. In contrast, *ASTNet* adopts an asynchronous spatiotemporal modeling paradigm and uses different patch lengths for temporal and spatial modeling, offering superior computational efficiency. The variant *ASTNet* w/o Async, which removes asynchronous computation, still maintains competitive performance in both latency and GPU memory usage.

7 Hyper-parameters Study (RQ4)

This section examines the impact of key hyperparameters on the performance of our model, evaluated using two metrics: Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). The results are discussed across four critical hyperparameters: P_t (Temporal Embedding Patch Length), P_s (Spatial Embedding Patch Length), L (Length of Lookback Window), and d (Embedding Dimension). Figure 3 draws the impact of hyper-parameters on the representative dataset A.

Temporal Embedding Patch Length P_t . The impact of varying the temporal embedding patch length, P_t , is shown in the top-left plot. The results demonstrate that the model’s MAE consistently improves as P_t increases, reaching a minimum at $P_t = 32$, beyond which performance slightly deteriorates. The MAPE metric follows a similar trend, with the best performance observed at $P_t = 32$. This suggests that a moderate patch length captures the temporal dependencies effectively without introducing excessive complexity.

Spatial Embedding Patch Length P_s . The spatial embedding patch length, P_s , influences the performance of the model, as seen in the top-right plot. Unlike the temporal patch length, the model performs best at $P_s = 8$ in terms of both MAE and MAPE, with slight performance degrading as P_s increases. This indicates that modeling spatial dependencies does not require very fine-grained features, as opposed to modeling temporal dependencies.

Length of Lookback Window L . The effect of the lookback window length, L , is illustrated in the bottom-left plot. MAE decreases as L increases from 64 to 128, but performance plateaus and even worsens slightly for longer lookback windows (e.g., at $L = 1024$). MAPE exhibits a similar trend. This suggests that the model benefits from a moderate lookback window size, which provides a good balance between capturing sufficient historical context and avoiding overfitting with excessive data, which could introduce more noise and redundant information that negatively impacts forecasting accuracy.

Embedding Dimension d . Finally, the embedding dimension d is evaluated in the bottom-right plot. Both MAE and MAPE improve significantly as d increases from 32 to 128. However, further increases in d lead to diminishing returns, and performance begins to stabilize. This implies that an embedding dimension around 128 strikes a balance between representational capacity and model complexity, avoiding both underfitting and overfitting.

8 Deployment Scenario

In practical applications, the *ASTNet*, integrated with the intelligent monitoring and management platform developed by SUPCON Technology Co., Ltd., enables efficient and accurate prediction and management of sensor data in chemical plants. This real-time prediction system has been successfully deployed and is operational in three major chemical plants in China. The system collects real-time sensor data from chemical plants, transmits it to *ASTNet* for prediction, and relays the results to the management platform. When *ASTNet* predicts potential deviation trends in sensor data, the system generates immediate early warning reports, enabling technical personnel to make remote equipment adjustments efficiently. In addition, the system is designed with multiple functional modules

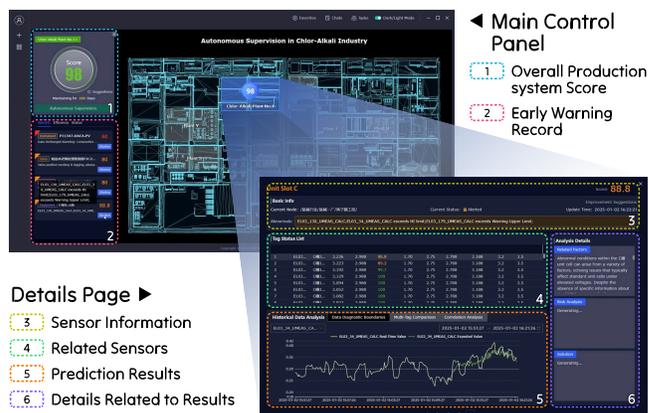


Figure 4: Demonstration of the deployment scenario. The main control panel provides a comprehensive overview of the production system, and the details page displays related information and prediction results about individual sensors.

to support technical personnel in real-time scheduling and monitoring of the production process. As shown in Figure 4, through flexible filtering conditions (such as equipment, area, time, etc.), operators can comprehensively grasp the overall operational status of the factory on the main control panel and delve into the real-time data of each sensor and the corresponding prediction results on the details page, enabling efficient equipment assessment and traceability analysis. The displayed prediction results and real data are presented in line chart form, allowing operator to switch between different time intervals and sensors. The pale yellow line represents real-time value, while the green line represents predicted value by *ASTNet*, showing a close match between predictions and the actual future trends. The results indicate that *ASTNet* has achieved excellent prediction accuracy in the real data. Please refer to Appendix A for a detailed description of the system.

9 CONCLUSIONS

ASTNet proposes a novel approach for real-time spatiotemporal forecasting in chemical sensor networks, addressing computational latency and complex spatial dependencies. It features asynchronous modeling to reduce latency and dynamic graph fusion to enhance robustness. Experiments show superior accuracy and efficiency over state-of-the-art methods, making it suitable for large-scale industrial applications and improving decision-making in chemical production.

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. In *Proceedings of the 33rd International Conference on Machine Learning (ICML 2016)*.
- [2] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. 2015. *Time series analysis: forecasting and control*. John Wiley and Sons.
- [3] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Spectral Temporal Graph Neural Network for Multivariate Time-series Forecasting. In *Advances in Neural Information Processing Systems*.
- [4] Jiacheng Cen, Anyi Li, Ning Lin, Yuxiang Ren, Zihe Wang, and Wenbing Huang. 2024. Are High-Degree Representations Really Unnecessary in Equivariant Graph Neural Networks?. In *Advances in Neural Information Processing Systems*.
- [5] Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassioulas, Jure Leskovec, and Rex Ying. 2024. From Similarity to Superiority: Channel Clustering for Time Series Forecasting.
- [6] Si-An Chen, Chun-Liang Li, Sercan O. Arik, Nathanael Christian Yoder, and Tomas Pfister. 2023. TSMixer: An All-MLP Architecture for Time Series Forecasting. *Transactions on Machine Learning Research* (2023).
- [7] Tao Dai, Beiliang Wu, Peiyuan Liu, Naiqi Li, Jigang Bao, Yong Jiang, and Shu-Tao Xia. 2024. Periodicity Decoupling Framework for Long-term Series Forecasting. *International Conference on Learning Representations* (2024).
- [8] Zheng Dong, Renhe Jiang, Haotian Gao, Hangchen Liu, Jinliang Deng, Qingsong Wen, and Xuan Song. 2024. Heterogeneity-Informed Meta-Parameter Learning for Spatiotemporal Time Series Forecasting. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
- [9] Taoran Fang, Tianhong Gao, Chunping Wang, YihaoShang, Wei Chow, Lei Chen, and Yang Yang. 2025. KAA: Kolmogorov-Arnold Attention for Enhancing Attentive Graph Neural Networks. In *International Conference on Learning Representations*.
- [10] Yuchen Fang, Yuxuan Liang, Bo Hui, Zezhi Shao, Liwei Deng, Xu Liu, Xinke Jiang, and Kai Zheng. 2025. Efficient Large-Scale Traffic Forecasting with Transformers: A Spatial Data Management Perspective. In *SIGKDD*.
- [11] Yuchen Fang, Yanjun Qin, Haiyong Luo, Fang Zhao, Bingbing Xu, Liang Zeng, and Chenxing Wang. 2023. When spatio-temporal meet wavelets: Disentangled traffic forecasting via efficient spectral graph attention networks. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*.
- [12] Yuchen Fang, Fang Zhao, Yanjun Qin, Haiyong Luo, and Chenxing Wang. 2022. Learning All Dynamics: Traffic Forecasting via Locality-Aware Spatio-Temporal Joint Transformer. *IEEE Transactions on Intelligent Transportation Systems* (2022), 23433–23446.
- [13] Everette S Gardner Jr. 1985. Exponential smoothing: The state of the art. *Journal of forecasting* (1985).
- [14] Jindong Han, Weijia Zhang, Hao Liu, Tao Tao, Naiqiang Tan, and Hui Xiong. 2024. BigST: Linear Complexity Spatio-Temporal Graph Neural Network for Traffic Forecasting on Large-Scale Road Networks. *Proceedings of the VLDB Endowment* (2024).
- [15] Jindong Han, Weijia Zhang, Hao Liu, Tao Tao, Naiqiang Tan, and Hui Xiong. 2024. BigST: Linear Complexity Spatio-Temporal Graph Neural Network for Traffic Forecasting on Large-Scale Road Networks. *Proceedings of the VLDB Endowment* (2024).
- [16] S Hochreiter. 1997. Long Short-term Memory. *Neural Computation* MIT-Press (1997).
- [17] John J. Hopfield. 1982. Time Series Prediction Using a Recurrent Neural Network. In *Proceedings of the IEEE International Conference on Neural Networks*.
- [18] Renhe Jiang, Zhaonan Wang, Yudong Tao, Chuang Yang, Xuan Song, Ryosuke Shibasaki, Shu-Ching Chen, and Mei-Ling Shyu. 2023. Learning Social Meta-knowledge for Nowcasting Human Mobility in Disaster. In *Proceedings of the ACM Web Conference 2023*.
- [19] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. 2023. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- [20] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. 2021. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *International Conference on Learning Representations*.
- [21] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.
- [22] Vasantha Kumar and Lelitha Vanajakshi. 2015. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. In *Proceedings of the 22nd International Symposium on Transportation and Traffic Theory (ISTTT 2015)*. 1–9.
- [23] Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. 2023. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data* (2023).
- [24] Juren Li, Yang Yang, Youmin Chen, Jianfeng Zhang, Zeyu Lai, and Lujia Pan. 2024. DWLR: Domain Adaptation under Label Shift for Wearable Sensor. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*.
- [25] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations (ICLR '18)*.
- [26] Zhehan Li, Yanjie Zhu, Junbo Zhang, and Zhenhui Li. 2021. Dynamic and Multi-faceted Spatio-temporal Deep Learning for Traffic Speed Forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.
- [27] Xu Liu, Yuxuan Liang, Chao Huang, Hengchang Hu, Yushi Cao, Bryan Hooi, and Roger Zimmermann. 2024. Reinventing Node-Centric Traffic Forecasting for Improved Accuracy and Efficiency. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*.
- [28] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations (ICLR '23)*.
- [29] Qian Ma, Zijian Zhang, Xiangyu Zhao, Haoliang Li, Hongwei Zhao, Yiqi Wang, Zitao Liu, and Wanyu Wang. 2023. Rethinking Sensors Modeling: Hierarchical Information Enhanced Traffic Forecasting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM)*.
- [30] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *Proceedings of ICLR*.
- [31] Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. 2025. DUET: Dual Clustering Enhanced Multivariate Time Series Forecasting. In *SIGKDD*.
- [32] Gustavo HT Ribeiro, Paulo SG de M Neto, George DC Cavalcanti, and Ren Tsang. 2011. Lag selection for time series forecasting using particle swarm optimization. In *The 2011 International Joint Conference on Neural Networks*.
- [33] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. 2022. Spatial-Temporal Identity: A Simple yet Effective Baseline for Multivariate Time Series Forecasting. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*.
- [34] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. 2022. Pre-training Enhanced Spatial-temporal Graph Neural Network for Multivariate Time Series Forecasting. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*.
- [35] SM Tauseef, Tasneem Abbasi, R Suganya, and SA Abbasi. 2017. A critical assessment of available software for forecasting the impact of accidents in chemical process industry. *International Journal of Engineering, Science and Mathematics* (2017).
- [36] Xiaoyang Wang, Yao Ma, Yiqi Wang, Wei Jin, Xin Wang, Jiliang Tang, Caiyan Jia, and Jian Yu. 2020. Traffic flow prediction via spatial temporal graph neural network. In *Proceedings of the web conference 2020*.
- [37] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*.
- [38] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- [39] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*.
- [40] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. HOW POWERFUL ARE GRAPH NEURAL NETWORKS? (2019).
- [41] Liang Yao, Xianfeng Tang, Huaiyu Wu, Junbo Zhang, and Yu Zheng. 2020. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*.
- [42] X. Zhang, Y. Liu, Z. Li, and Z. Liu. 2020. StemGNN: Spatio-Temporal Graph Neural Network for Traffic Forecasting. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*.
- [43] Yuelin Zhang, Jiacheng Cen, Jiaqi Han, Zhiqiang Zhang, Jun Zhou, and Wenbing Huang. 2024. Improving equivariant graph neural networks on large geometric graphs via virtual nodes learning. In *Forty-first International Conference on Machine Learning*.
- [44] Yunhao Zhang and Junchi Yan. 2023. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*.
- [45] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*.

A Detail of Deployment Scenario

In this section, we elaborate on the deployment details. Figure 5 shows the intelligent management system’s two main pages based on ASTNet real-time prediction: the main control panel and the details page. The main functional modules in the interface are identified, numbered with dashed boxes, and their information is shown in the legend.

Main Control Panel. On this page, Module 1 records the production system’s overall score, considering factors like sensor operating time, production efficiency, and warning records to help technical personnel understand the factory’s overall operation status. Module 2 shows early warning records. As noted in Section 8, when *AST-Net* predicts a sensor may have future trend anomalies, the system promptly generates warning reports to aid technical personnel in remote device control. Warning records are color-coded by urgency: orange for normal warnings and red for significant predicted trend deviations. The main control panel’s right side visualizes the factory’s architecture to show production units’ layouts and statuses; clicking a unit navigates to its detail page for more info.

Detail Page. In more detail, Module 3 displays the basic information of the currently selected sensor, including its hierarchical structure, warning records, sensor score, etc. Technical personnel can view information on various sensors through flexible filtering conditions. Furthermore, the list in Module 4 includes all sensors related to the current sensor, facilitating status analysis and interface navigation. Module 5 is the main interface for viewing sensor data, supporting the viewing of prediction results and actual data through time interval filtering. In the graph, the light yellow line represents the actual values, while the green line represents predicted values. Additionally, it supports multi-sensor comparison and correlation analysis, greatly aiding technical personnel in equipment diagnostics and traceability analysis. Finally, Module 6 will display more detailed result analysis (If pre-setting is done) and supports editing.

B Baselines

In this section, we present the details of baseline methods based on three distinct models.

Non-spatial Modeling-based Methods . PatchTST [30]:This is a Transformer-based model for long-term time series forecasting, where the input series is divided into fixed-length patches treated as tokens, enabling efficient modeling of temporal dependencies through a patch-based attention mechanism. PDF [7]:This is a periodicity decoupling framework for long-term time series forecasting, which separates periodic and non-periodic components of the series to improve forecasting accuracy by modeling them independently through specialized modules.

Time-invariant Spatial-based Methods . STID [33]:This is a spatial-temporal identity framework for multivariate time series forecasting, which leverages simple yet effective embeddings to capture spatial and temporal dependencies without complex architectural designs. AGCRN [41]:This is an adaptive graph convolutional recurrent network for traffic forecasting, which integrates graph convolutional networks (GCN) and recurrent neural networks (RNN) to dynamically capture spatial dependencies and temporal patterns in traffic data. MTGNN [38]:This is a multivariate time series forecasting framework that leverages graph neural

networks (GNN) to model inter-series dependencies and temporal patterns jointly, enabling effective representation learning for complex multivariate forecasting tasks. StemGNN [42]:This is a spectral temporal graph neural network for multivariate time series forecasting, which combines graph Fourier transform (GFT) and temporal convolution to capture both spatial dependencies and temporal dynamics in the spectral domain.

Time-varying Spatial-based Methods . MegaCRN [19]:This is a spatio-temporal meta-graph learning framework for traffic forecasting, which employs meta-graphs and graph convolutional recurrent networks (GCRN) to adaptively model complex spatial and temporal dependencies in traffic data. HimNet [8]:This is a heterogeneity-informed meta-parameter learning framework for spatiotemporal time series forecasting, which adaptively learns task-specific parameters to address heterogeneous patterns across spatial and temporal dimensions through meta-learning. PatchSTG [10]:This is a Transformer-based framework for large-scale traffic forecasting, which introduces a patch-based approach to manage spatial data efficiently, enabling scalable modeling of spatiotemporal dependencies with reduced computational complexity. Crossformer [44]:This is a Transformer-based model for multivariate time series forecasting, which explicitly leverages cross-dimension dependencies by integrating inter-series and intra-series relationships through a novel attention mechanism. DUET [31]:This is a dual clustering-enhanced framework for multivariate time series forecasting, which integrates clustering mechanisms to capture both global and local patterns, improving the modeling of complex dependencies across multiple time series.

C Evaluation Metrics.

To comprehensively evaluate the model’s performance, we utilized three common forecasting evaluation metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

MAE measures the average absolute difference between predicted values and actual values. The formula for MAE is defined as follows:

$$\text{MAE} = \frac{1}{H} \sum_{i=1}^H |y_{\text{horizon}}^{(i)} - \hat{y}_{\text{horizon}}^{(i)}| \quad (15)$$

Here, y_{horizon} represents the actual value, \hat{y}_{horizon} represents the predicted value, and H denotes the number of future timestamps to be predicted, the same applies to the following equations. MAE demonstrates robustness against anomalous fluctuations and noise, effectively matching the real sensor data. By not squaring errors, MAE reduces the impact of anomalous values caused by measurement accuracy issues in sensor data, enhancing the evaluative reference.

RMSE quantifies the square root of the average of the squared differences between predicted and actual values, calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{H} \sum_{i=1}^H (y_{\text{horizon}}^{(i)} - \hat{y}_{\text{horizon}}^{(i)})^2} \quad (16)$$

RMSE is the square root of Mean Squared Error ($\frac{1}{H} \sum_{i=1}^H (y_{\text{horizon}} - \hat{y}_{\text{horizon}})^2$). It is sensitive to large errors and helps assess model

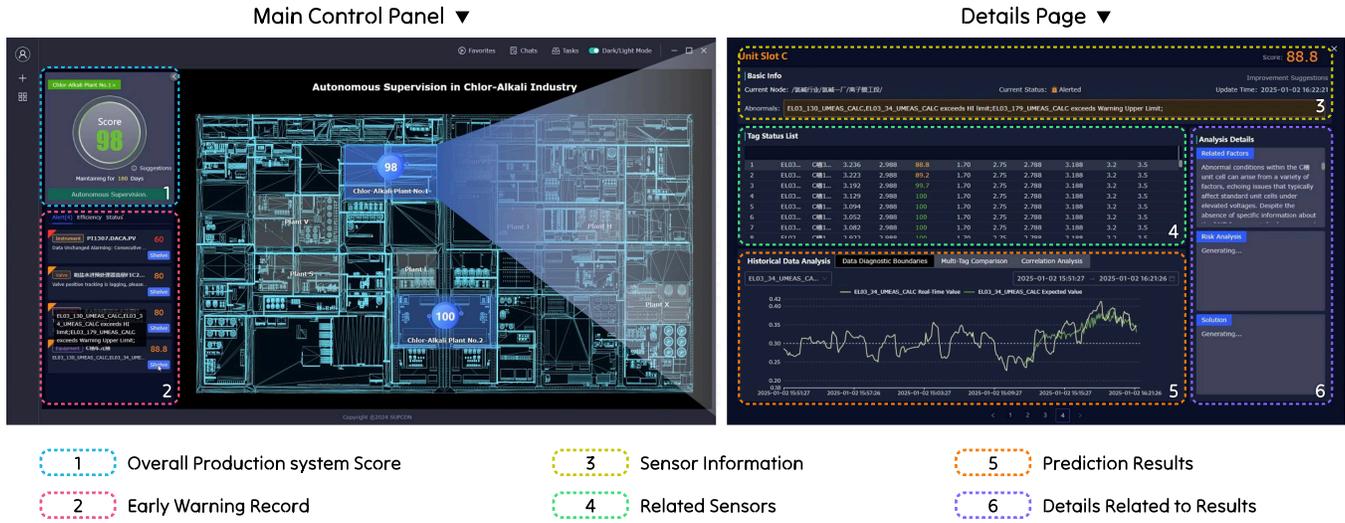


Figure 5: Detail of Deployment Scenario

| Variable Relationship | Maximum | Minimum | Median | Mean | Variance |
|---|---------|----------|--------|------|----------|
| Current->Brine pH | 1.00 | 0.000081 | 0.42 | 0.45 | 0.08 |
| Current->Chlorine Pressure | 0.99 | 0.000668 | 0.49 | 0.48 | 0.09 |
| Current->Hydrogen Pressure | 0.98 | 0.000006 | 0.43 | 0.44 | 0.08 |
| Hydrochloric Acid Flow->Brine pH | 1.00 | 0.000000 | 0.00 | 0.10 | 0.05 |
| Hydrochloric Acid Flow->Chlorine Pressure | 1.00 | 0.001383 | 0.41 | 0.44 | 0.09 |
| Hydrochloric Acid Flow->Hydrogen Pressure | 1.00 | 0.000009 | 0.40 | 0.43 | 0.09 |
| Caustic Flow->Brine pH | 0.99 | 0.000053 | 0.51 | 0.50 | 0.08 |
| Caustic Flow->Chlorine Pressure | 1.00 | 0.002038 | 0.47 | 0.48 | 0.08 |
| Caustic Flow->Hydrogen Pressure | 1.00 | 0.001726 | 0.46 | 0.47 | 0.08 |

Table 5: Variable Causal Analysis

accuracy. By taking the square root of MSE, RMSE solves the problem of unit inconsistency in MSE, providing an error measure in the same units as the original data to better describe losses.

MAPE reflects the percentage of prediction errors relative to actual values, offering a perspective on relative errors. The formula for MAPE is articulated as:

$$MAPE = \frac{100\%}{H} \sum_{i=1}^H \left| \frac{y_{horizon}^{(i)} - \hat{y}_{horizon}^{(i)}}{y_{horizon}^{(i)}} \right| \quad (17)$$

In percentage form, MAPE enables comparisons across different data magnitudes, providing consistent error assessments for data of varying scales and units, aiding in comparing model performances across different datasets and sensor nodes.

These metrics evaluate model performance from multiple perspectives. Comprehensive consideration of them enables thorough assessment of the model on real sensor data.

D Data Analysis on Motivation

A causal analysis was conducted on Dataset A from the chlor-alkali industry, focusing on the relationships among key variables. Following preprocessing, 10,000 time slices (each 5,000 data points in length) were sampled for each variable. The Granger causality test with various lags was then applied to these samples. For each causal direction $x \rightarrow y$, 10,000 p-values were obtained for statistical analysis, with each p-value representing the causal strength at a specific time point.

The statistic information is depicted in Fig. 5. Results indicate that causal strengths exhibit temporal variability. The majority of relationships showed no significant causal link (characterized by high mean p-values), whereas some demonstrated significant associations (low minimum p-values). These findings underscore the dynamic nature of causal relationships in chemical processes and will be incorporated into the revised manuscript.