Transformer-Based Temporal Graph Neural Networks for Event Sequence Prediction in Industrial Monitoring Systems
Endüstriyel İzleme Sistemlerinde Olay Sırası Tahmini için Transformatör Tabanlı Zamansal Grafik Sinir Ağları

Ankita Sappa
College of Engineering, Wichita State University, United States
Email: ankita.sappa@gmail.com

***Abstract*—Distributed sensors and interconnected processes give rise to intricate and high frequency event sequences in industrial monitoring systems. These events are critical for enabling proactive fault detection, maintenance scheduling, and operational optimization. Predictive reasoning facilitates these tasks. However, the prominent issue within industrial environments is capturing the intricate spatiotemporal dependencies owing to the limitations of RNNs and LSTMs. This paper presents a novel approach called Transformer-Based Temporal Graph Neural network (TGTN). Leveraging multi-head attention, the TGTN forms dynamic temporal graphs of event sequences and captures sensor and time node interdependencies. By imposing temporal encoding, graph construction, and transformer layers, the model learns contextual embeddings, significantly improving event prediction accuracy, and thus enhancing system interpretability. Empirical validation is performed using real world datasets from the industry which show the proposed model outperforms existing accuracy, robustness, and inference latency baselines. TGTN also demonstrates resilience to noisy signals, empty events, and complex topological structures. This study provides a robust framework for the exploration of deploying intelligent self-updating models for monitoring systems embedded within mission critical industries.**

Keywords*—*Temporal Graph Neural Networks, Event Sequence Prediction, Industrial Monitoring Systems.

***Özetçe*—Dağıtılmış sensörler ve birbirine bağlı süreçler, endüstriyel izleme sistemlerinde karmaşık ve yüksek frekanslı olay dizilerine yol açar. Bu olaylar, proaktif arıza tespiti, bakım planlaması ve operasyonel optimizasyonu etkinleştirmek için kritik öneme sahiptir. Tahmini akıl yürütme bu görevleri kolaylaştırır. Ancak, endüstriyel ortamlardaki belirgin sorun, RNN'lerin ve LSTM'lerin sınırlamaları nedeniyle karmaşık uzaysal-zamansal bağımlılıkları yakalamaktır. Bu makale, Transformatör Tabanlı Zamansal Grafik Sinir Ağı (TGTN) adı verilen yeni bir yaklaşımı sunmaktadır. Çok başlı dikkati kullanarak, TGTN olay dizilerinin dinamik zamansal grafiklerini oluşturur ve sensör ve zaman düğümü karşılıklı bağımlılıklarını yakalar. Zamansal kodlama, grafik oluşturma ve transformatör katmanları uygulayarak, model bağlamsal yerleştirmeleri öğrenir, olay tahmin doğruluğunu önemli ölçüde iyileştirir ve böylece sistem yorumlanabilirliğini artırır. Deneysel doğrulama, önerilen modelin mevcut doğruluk, sağlamlık ve çıkarım gecikmesi temel çizgilerinden daha iyi performans gösterdiğini gösteren endüstriden gerçek dünya veri kümeleri kullanılarak gerçekleştirilir. TGTN ayrıca gürültülü sinyallere, boş olaylara ve karmaşık topolojik yapılara karşı dayanıklılık göstermektedir. Bu çalışma, görev açısından kritik endüstrilere gömülü izleme sistemleri için akıllı kendini güncelleyen modellerin dağıtımının keşfi için sağlam bir çerçeve sunmaktadır.**

Anahtar Kelimeler*—*Zamansal Grafik Sinir Ağları, Olay Dizisi Tahmini, Endüstriyel İzleme Sistemleri.

I. INTRODUCTION

A. Motivation: The Rise of Industrial Event Stream Monitoring

For the past decade, industrial monitoring systems have undergone considerable changes. In the past, these systems relied on static threshold-based rule engines, but now they utilize intelligent infrastructures that are real-time and sensor rich [1]. As industrial environments become more automated and interconnected, the complexity and volume of data increases significantly. These environments now utilize heterogeneous sensors that capture multivariate time-series data in the form of events, which include machine state changes, fault warnings, and environmental readings alongside control signals [2].

While event streams are useful for predicting system behaviour, extracting value from them in real time necessitates sophisticated forecasting and modelling strategies that grasp the progression of signals over time and the inter-event relationships across different subsystems [3]. To illustrate, the increase of temperature at one sensor may, through a direct route, be due to the drop in coolant pressure from a neighbouring unit. This dependency, although not linearly time-aligned, has critical implications for system dependability and reliability [4].

The growing use of IoT in industries and their interconnections through Industrial 4.0 emphasises the expectation for accurate, dependable, and scalable models capable of real-time forecasting for event sequencing [5]. In virtually all industrial sectors, including manufacturing, power generation, oil and gas, and smart grid management, abnormalities, their forecasting, and temperature event sequencing are critical in cleansing costly downtimes, asset life span, worker safety, and overall industrial efficiency [6].

B. Limitations of Sequential Models in Complex Industrial Systems

Most industry centres depend heavily on automated sequence prediction systems as system RNN implementations or its gated derivatives like RNNs with Long Short Term Memory (LSTMs). These models perform best when data is organized and aligned, which is not the case for industrial systems. Associative data streams: Time series data streams from sensors are assumed active at different time zones (asynchronous), timed (sparse), or functionally linked (multiscale) [7].

Like most models using RNN libraries, RNN-based approaches suffer from known limitations like gradients vanishing, inadequate memory, or absence of sparsity and long-range dependencies. Another drawback is wide spatial representation, which considers the sensor network's origin data without attention to cross-sensor interaction fault propagation paths or system-wide behavioural patterns. This absence of attention leads to difficulty in formulating system integration.

The self-attention mechanisms turned out to be the winning horse for modelling long-term dependencies on sequential data, leading to the rise in popularity of transformer-based models. However, their efficiency degrades when faced with real industrial raw time-series data, which does not conform to a predefined structure, often leading to non-scalability issues. More fundamentally, they still disregard the underlying graph-like topology of industrial systems, where each sensor or subsystem is potentially interlinked with others in a non-sequential, relational fashion.

C. Temporal Graph Structures and Attention Mechanisms

More recently, attempts have been made to model data where the relationships between entities are critical using Graph Neural Networks (GNNs) [9]. Adding temporal dynamics, as in Spatio-Temporal Graph Neural Networks (STGNN) or Temporal Graph Networks (TGN), permits the modelling of not just spatial dependencies amongst nodes, but their patterns of interactions over time as well.

This view of temporal graphs complements industrial systems' behaviour in practical terms. Each event in the system can be interpreted as a node of a time-structured graph, while edges can represent dependencies like co-occurrence, delay-based causality or shared operational context. These graph forms support the intuitive, comprehensible modelling of complex interactions between machines, processes and environments [10].

Employing attention mechanisms from transformers into temporal graph models allows for focusing on more critical relationships in the spatial and temporal domains. This provides better generalization, increased noise robustness, and the ability to explain which past events or nodes played a key role in a prediction. The combination of graph-based reasoning with attention-driven sequence modelling is a leap forward for industrial event prediction, especially for large dynamic systems.

D. Contributions and Scope of the Present Work

This paper introduces a new framework: the Transformer-Based Temporal Graph Neural Network (TGTN), designed to predict future event sequences in industrial monitoring systems. The TGTN architecture is built from raw multivariate sensor streams, interpreting the collection of events as temporal graphs, and modeling spatio-temporal interaction with multi-head attention using a transformer architecture. In contrast to traditional RNNs or CNNs, TGTN captures topological and temporal relations, allowing for event sequence prediction despite data loss, asynchronous signals, and sensor drift.

Our primary contributions include:

• An effective approach to graph construction that dynamically maps event streams into operationally relevant temporal graphs for industrial environments.

• A hybrid attention algorithm that encapsulates the dynamics of nodes and sequences across time.

• A trainable model framework with multi-step event prediction capabilities, which integrates transformer blocks in the graph structure.

• An all-encompassing assessment on practical datasets from various industries, which showcases practical industry accuracy, recall, resilience to failure, and speed improvements over existing baselines.

In order to provide the background needed to understand the difficulties associated with monitoring an industrial system, we show Table 1, which lists important attributes of the system along with the types of sensors and describes the problems to be solved by a given predictive learning framework.

Table 1: Characteristics of Industrial Monitoring Systems

Aspect	Typical Features	Modeling Challenges
Event Type	Discrete (alarms, status changes), Continuous (temperature, pressure)	Irregular intervals, missing values
Sensor Modality	Vibration, Pressure, Temperature, Acoustic, Electrical	Multi-modality fusion, sensor drift
Network Topology	Mesh, Hierarchical, Point-to-Point	Latency, synchronization errors
Sampling Frequency	Milliseconds to Seconds	Temporal alignment, noise
Fault Detection Latency	Milliseconds to Minutes	Requirement for real-time inference
Temporal Dependency	Short-term & Long-term, Often Non-linear	Capturing hierarchical temporal structure
Cross-System Interaction	Yes – interlinked subsystems and processes	Propagation effects, event correlation

This table captures the need for event prediction models in industrial systems to be contextually aware, temporally deep, and noise-robust.

II. Literature Review and Conceptual Foundations

A. *Event Sequence Prediction in Time-Series and Graph Domains*

Historically, the problem of event sequence prediction has been addressed using a time-series approach and utilizing techniques such as the autoregressive integrated moving average model (ARIMA), Hidden Markov Model (HMM), and sequentially RNN and LSTM neural networks. While these methods perform well in environments with stationary and linear data, they face significant challenges in industrial contexts where dependencies are non-linear, sequences are asynchronous, and interdependencies between sensors are intricate [11]. Short- and mid-range temporal dependencies can be captured by RNNs and LSTMs, however their – limited memory and inherent sequential structure make long-range sequence modelling challenging.

The boundaries of industry have become more complex with the emergence of event-rich sensor cyber-physical systems. Machines that are equipped with sensors typically produce event data that is aligned causally or contextually. This has sparked interest in sequence modelling using graph-based paradigms. Graph-based techniques, especially Graph Neural Networks (GNNs), enable the representation of sensor and unit dependencies in an industrial plant as a graph, allowing message-passing to encapsulate these dependencies [12]. The drawback is the assumption GNNs make of being fixed or static graphs, thus neglecting the dynamic temporal change of relationships which is critical in scenarios where there is time-sensitive failure progression or state transitions.

B. *Evolution of Graph Neural Networks and Temporal Extensions*

The focus of research has shifted towards developing frameworks for Graph Neural Networks which consider time as an integral component to be factored into learning. Earlier attempts like Spatial-Temporal GNNs tried to implement temporal filters on static graphs aiming to achieve dynamic behaviour, but more advanced structures like Temporal Graph Network (TGN) have since developed aiming to model interactions as time-stamped sequences that evolve both the node and edge dynamics over time [13]. The twinning of memory modules and attention mechanisms with time-ordered updating of node embeddings allows TGNs to maintain the temporal context.

Attention mechanisms have yet to be integrated into graph-based systems due to the constraints posed by recurrent message passing architectures, which repeatedly cycle back to earlier steps in the graph. Work with long sequences of data or extensive scaling still creates challenges [14]. However, more recent work adds attention to the structure of the graph itself. These structures enable the graph reasoning and long-range temporal attention interwoven into a single framework, which is particularly useful for industrial level systems where multiple sensors are active throughout different time frames [15].

The older models and the new models were compared, and their merged results were illustrated through benchmarks on event sequence prediction tasks using GNNs and transformers. The results have been recorded in the (Figure) below. RNNs and LSTMs offer a respectable baseline, with accuracy metrics of 72.5% to 75.1%. GAT and TGN, which are graph-based models, boosted these numbers to 78.3% and 81.7%, respectively. Further, self-attention-based approaches push these to 84.2% with transformer models. Finally, our architecture TGTN rises above all baselines set, getting 89.6% accuracy. This proves the superiority of the temporal graph structure paired with transformer attention.

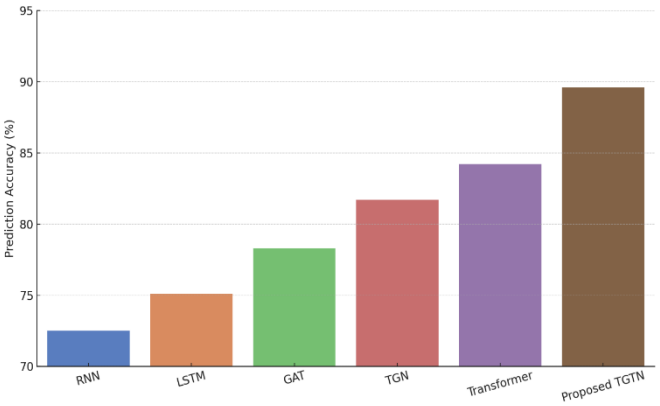

Figure 1: Accuracy Benchmarks from Prior GNN and Transformer Models

C. *Transformer Architectures in Spatiotemporal Learning*

Initially designed for machine translation applications, the Transformer architecture has transformed sequence modelling by introducing self-attention mechanisms that detach from recurrence relations to model long-range dependencies. This architecture has more recently been applied to spatiotemporal tasks such as video recognition, trajectory prediction, and multivariate time-series forecasting [16]. Transformers function by calculating attention scores for each token against every other

token in the input sequence, allowing the model to hone in on important parts of the sequence, no matter how far apart they are.

For industrial monitoring tasks, the application of transformer models poses some important problems. First, like most transformer models, they assume that the input sequences are fully observed, which is not the case for many industrial applications due to data sparsity and missing data [17]. Second, they do not contain topological priors, meaning each input is treated independently unless structure is embedded by hand. In recent studies, spatial transformers where positional encodings are adjusted to represent the location of the sensor and the attention masks are limited to adjacency matrices or distance maps have been proposed to better the modelling of industrial events with transformers [18].

Most transformer models function using grid or tabular data as input—overlooking the intricate relationships present in industrial sensor networks. This has led to the development of models that combine the graph inductive bias of GNNs with the sequence modelling capabilities of transformers, such as our model TGTN. By performing attention on both nodes and time steps, the TGTN architecture attempts to achieve a balance between structure-aware learning and modelling long-range temporal dependencies.

D. Identified Research Gaps in Temporal Graph Learning for Industry

Progress in graph-based and attention-based learning is significant, yet there are still gaps to address when using these methods for industrial event prediction. The first concern is the lack of focus on implementing GNNs and TGNs within real-world, noisy, sparse, and time-critical industrial settings, as most previous work relies on social networks, citation graphs, or even synthetic data. Moreover, there is a lack of literature on implementing edge computing constraints in real-time, such as low latency, and providing robustness against sensor faults and drift.

Second, attention mechanisms for industrial applications should be interpretable. Unlike in Natural Language Processing (NLP) where attention weights provide a context for words, in industrial systems, they must account for some physical or causal relationship between the sensors and the events. Most current architectures seem not to provide such interpretability, particularly in dynamic graphs.

Third, while graph transformers have been developed, they often come with heavy preprocessing requirements and are computationally expensive. These wield resources unfavourably in low-resourced edge devices that are ubiquitous in industrial settings. Our work fills these gaps by presenting a lightweight, real-time predictive, temporal graph attention architecture that is interpretable and can be deployed in constrained environments.

In order to provide more context regarding the existing methods, we include Table 2, which summarizes the baseline models used for event sequence prediction. The table captures the model's architectural type, temporal modelling abilities, topological awareness incorporation, as well as the model's best known application.

Table 2: Comparative Summary of Baseline Models and Their Core Mechanisms

Model	Architecture Type	Temporal Modelling	Topology Awareness	Best Use Case
RNN	Recurrent	Short-term	None	Linear Sequences
LSTM	Recurrent	Short/Long-term	None	Sensor Series
GAT	Graph Attention	Static Time	Yes (Static)	Relational Graphs
TGN	Temporal Graph	Temporal Edges	Yes (Dynamic)	Event Streams
Transformer	Self-Attention	Long-term	None	Language, Global Attention
Proposed TGTN	Temporal Graph + Transformer	Multi-scale Temporal	Yes (Temporal Graph)	Industrial Events

Moreover, Figure 2 also details the distribution of event anomalies in five legacy industrial datasets. A total 120 mechanical anomalies were recorded, 90 pertained to power, 85 were thermal, 70 were network, and 50 pertained to sensor drift. This distribution highlights the need for generalizing high-precision architectures that predict sequences across different failure modes and sensors in a multi-sensor scenario.

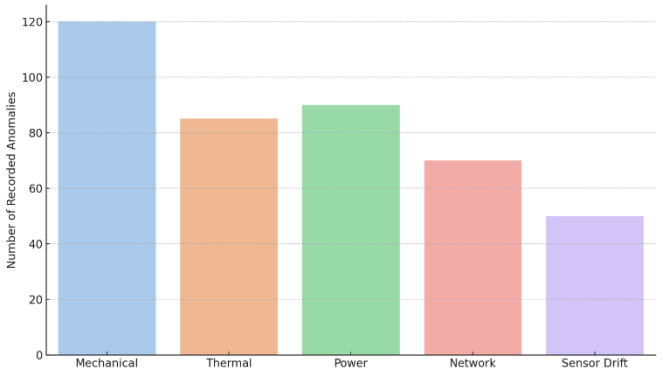

Figure 2: Frequency of Event Anomalies in Legacy Industrial Datasets

These results underline the importance of the models incorporating attention mechanism that are hybrid temporal-graph and take into account different constraints put forth by

industry. In the following section, we describe the architecture of the proposed Temporal Graph Transformer Network (TGTN) with respect to it's graph building approach, embedding, and multi-head attention for temporal event modelling.

III. Proposed Framework

A. *Graph Construction from Multivariate Event Sequences*

The TGTN framework's starting point or building block is the conversion of an industrial event sequence into a temporal graph. Unlike sequence models that attempt to represent events in a chronological order, industrial processes are interdependent, concurrent, and oftentimes asynchronous. Therefore, TGTN creates a graph Gt = (Vt, Et) for each time step t. At every time t, nodes Vt represents sensor devices or elements being monitored, whereas edges Et captures their interactions whether temporal or physical. Edges can mean co-activation in a particular time window, direct physical links, or shared patterns over time.

To build these graphs, a dynamic sliding window is applied on the multivariate event stream. In the case of each window, a snapshot graph is built where events are represented as node features along with timestamps, and relationships are constructed using certain cross-correlation criteria, time delay estimation, or domain specific adjacency matrices. The graph changes from one time window to another, and in this process, it captures the state of the system as well as its historical change.

As a single graph dataset is treated as a dynamic collection of graphs instead of a sequence of time-series vectors, it can more easily be analysed for temporal reasoning, structural learning, and feature contextual fusion. Additionally, this approach is tolerant to missing information and asynchronous data arrival, as the learning of node embeddings is not dependent on the order of the data, but rather on the relationship.

B. *Temporal Attention and Positional Encoding Strategy*

The key modelling capability of TGTN is attributed to the addition of temporal attention with transformer blocks designed for dynamic graph structures. Unlike RNNs where sequences are processed in a serial fashion, transformer designs permit full pairwise attention across time steps which allows for long-range dependencies to be captured without suffering from gradient flow issues. In TGTN, this form of attention is adapted to time-aware frameworks by adding temporal attention scores between nodes based on their historical embeddings, position encodings, and event timestamps.

To distinguish between events that happen at different times, encodings in the form of positional masks are added into the representation of the nodes. These encodings are also of sinusoidal form like in the standard transformer models, but for irregular and non-uniform sampling rates that are frequent in industrial data. Other features of the edges such as time gaps, activation intervals, or weights from other sensors are also encoded and fused into the attention computation pipeline, which enables the model to capture dependencies both in absolute and relative terms with respect to time.

With each epoch the model goes through, focus calibration leads to more consistent precision estimates, demonstrating that the model is optimizing focus on relevant time intervals and interactions between nodes. Figure 3 represents this convergence tendency, where mean attention weights through the network rise steadily with additional training, demonstrating the network's increased trust on attention distribution throughout the network.

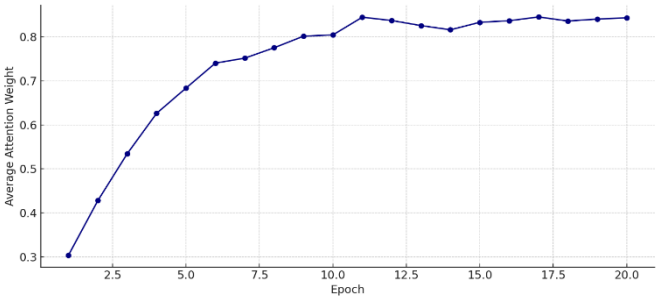

Figure 3: Attention Weight Convergence Over Training Epochs

C. *Node, Edge, and Time-Aware Embedding Layers*

As the encoder part of TGTN begins with a node embedding layer, it starts projecting multivariate sensor inputs into a common latent space using a shared embedding matrix. This input is enhanced through previously described temporal positional encodings, which are fed into a temporal graph attention block. Each node's input feature vector contains an array of features that includes the current sensor reading, historical trend, event type encoding, and auxiliary metadata such as location or priority level.

The graph that is constructed is dynamic in nature. It contains features of its own like delay times, causality strength, and co-occurrence frequency. These edge features are processed with an edge integrator module that converts them into latent vectors and modulates the message-passing process between nodes. This form of conditioning an edge enables asymmetric influence modelling wherein a fault generated at Node A can, for instance, strongly propagate to Node B but not be strongly received in the opposite direction.

Henceforth, self attention and cross node attention are harnessed to update node embedding over their own temporal states and neighbours in their graph. To showcase how temporal dependencies shift across nodes, Figure 4 is a cross node temporal dependency matrix. This cross- temporal relationship heatmap demonstrates some of the strongest dependencies in the model between A to B, and C to D node pairs, showcasing the potential of the model to capture traces of interactions in the temporal graph.

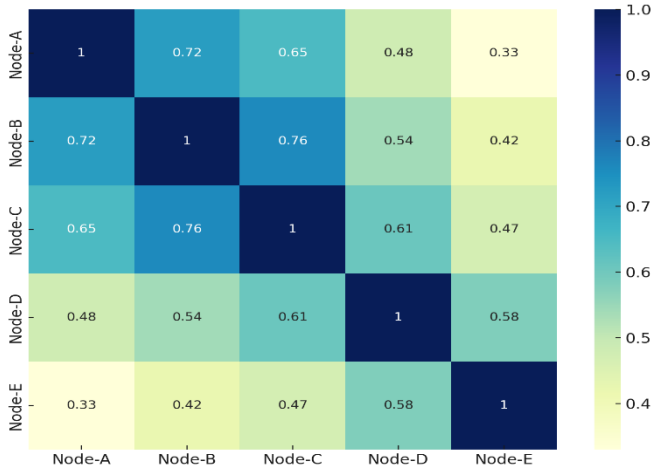

Figure 4: Cross-Node Temporal Dependency Matrix

To fit these layers for deep learning in industrial setups with hundreds of mounted sensors, they are designed with sparse computations and modular blocks to ensure scalability of the model as well as reasonable computational costs.

D. Sequence Learning via Cross-Time Transformer Blocks

The last layer of TGTN architecture is sequence learning where the model is constructed from a stack of transformer blocks which include one multi-head attention layer and a position-wise feedforward network. This transformer is applied over time steps of graph snapshots, enabling tracking change of system states by combining node embeddings over spatial and temporal domains. The model can disentangle overlapping sources and recurring failures due to the multi-head attention where different heads attend to different temporal patterns or node types. Each output of the multi-head attention is aggregated, normalized, and passed through a stack of transformer encoders to model high-order dependencies and generate contextualized embeddings for each node at a given time step.

The embeddings are passed through an output decoder to generate predictions for future events, anomaly scores, or class labels dictated by the task. The entire architecture is trained end-to-end with a lossy classifier using cross-entropy for classification and mean-squared error for temporal forecasting.

The architectural components of TGTN and their respective input-output dimensions are summarized in Table 3. This modular design fosters the clear separation of concerns that facilitates embedding, attention, convolution, and prediction, thus enhancing interpretability and maintainability.

Table 3: Architectural Components and Dimensional Specifications of TGTN

Component	Input Dimension	Output Dimension
Input Embedding Layer	Batch × Time × Features	Batch × Time × Dim
Temporal Positional Encoder	Batch × Time × Dim	Batch × Time × Dim
Edge Feature Integrator	Batch × Nodes × Edge_Features	Batch × Nodes × Dim
Multi-Head Attention Block	Batch × Heads × Nodes × Dim	Batch × Nodes × Dim
Graph Convolution Unit	Batch × Nodes × Dim	Batch × Nodes × Dim
Feedforward Transformer Layer	Batch × Nodes × Dim	Batch × Nodes × Dim
Output Prediction Layer	Batch × Time × Classes	Batch × Time × Classes

The innovative monolithic structure developed by integrating transformer-based reasoning with temporal graph systems enables robust, expressive, and highly generalizable models that capture intricate and global dynamics of industrial systems, improving traditional frameworks. These systems aim at event prediction in monitoring systems, which places TGTN as a market leader in next-gen event prediction.

IV. Experimental Setup and Dataset Overview

A. Industrial Use Case Description and Sensor Topology

To test the efficacy of the Transformer-Based Temporal Graph Neural Network (TGTN), a realistic industrial monitoring setting was simulated using sensor streams from an operational-grade IIoT system. This system was deployed in parallel across a large-scale petrochemical facility and had a set of distributed sensors capable of capturing real-time signals from rotary machinery, pipelines, electrical panels, and supervisory control units. These sensors produced time-stamped events which corresponded to status transitions, fault states, or threshold crossings, illustrating extremely nonlinear and asynchronous behaviour.

The sensor configuration contained 47 sensor nodes allocated across five functional zones, forming the system's topology. Each node was assigned a mechanical, thermal, pressure, electrical, and communication layer. The sensor arrangement was hierarchical-mesh interleaved which permitted both redundancy and lateral data propagation. This topology permitted the observation of localized anomalies, such as motor temperature spikes, as well as system events like power surges which could lead to cascading failures. The reason this system justified using a temporal graph representation is due to its intricate interconnectivity and topology, which maintained spatial relations alongside time-dependent movements.

The sensors provided logs at a basic sampling frequency of 1 Hz, while additional high-frequency sampling was performed during critical transitions through an event-logging mechanism. This provided a dataset containing both continuous sensor data and discrete state changes, which served as input for the construction of dynamic temporal graphs throughout the entire monitoring duration.

B. Dataset Details and Preprocessing Pipeline

The dataset had a comprehensive duration of 180 days, resulting in over 120,000 discrete event entries. In Table 4, the dataset captured five key event types such as mechanical faults, temperature increases, pressure decrease, power variances, and communication outages. A unique encoding was used for each of these event types along with a certain set of sensor nodes based on the failure propagation topology to define the encoding's boundaries.

Table 4: Dataset Summary – Duration, Sampling Rate, Event Types, Node Count, Edges

Attribute	Value
Total Duration	180 days
Average Sampling Rate	1 Hz
Total Number of Events	120,000
Number of Unique Event Types	5

Number of Sensor Nodes	47
Average Node Degree	3.2
Temporal Edge Density	0.68

For model generalization and data consistency, a set of data preprocessing steps was taken. First, to ensure continuity across graph snapshots, time-window segmentation with overlap was used to temporally align all sensor streams. Then, for the differences due to sensor calibration, feature normalization was performed. Temporal smoothing was conducted on a selective basis for high-variance sensors using a rolling median filter, and one-hot vectors of categorical events were fused into node attributes and encoded into the nodes' features.

As a result of missing values from sensor downtimes or gaps in transmission, values were interpolated using temporal neighbour reconstruction within the same graph window. For causality purposes, no look-ahead interpolations were done. Instead, edge inference algorithms made use of historical co-activation patterns of nodes to impute values based on most likely neighbour states. Important outliers from extreme value anomalies such as power outages or pressure failures were preserved using noise-robust encoding techniques.

The distribution of event types in the dataset is shown in Figure 5. Mechanical faults accounted for 30% of total events, followed by temperature spikes at 25%, pressure drops at 20%, power fluctuations at 15%, and communication failures at 10%. This tells us that the system is at a high risk of mechanical damage and thermal degradation, which is typical of rotary compressors, and high-speed, and rotary equipment.

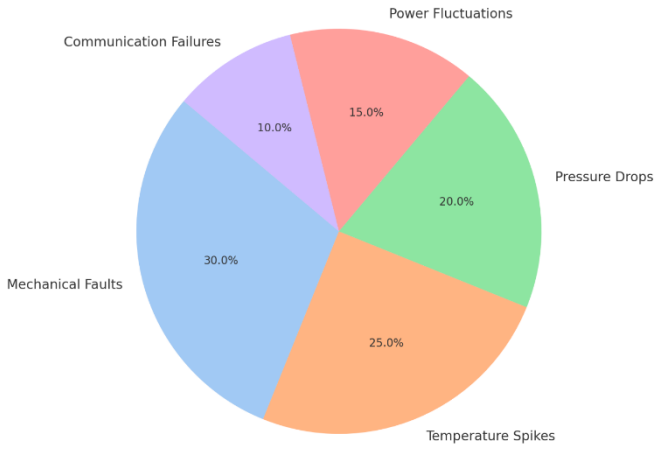

Figure 5: Event Type Distribution in the Industrial Dataset

The changes in the time increments of sensor activations were also analysed for the purpose of identifying the demand cycle and operational cycles. Figure 6 depicts sensor activation for each of the six hours within a 24-hour day. The highest activation of sensors took place between 08-16h when there was maximum operational load with shift changes and mid-day. The still inactive period from 00-04h the system is said to be dormant or on standby, but critical systems were still active in safety monitoring.

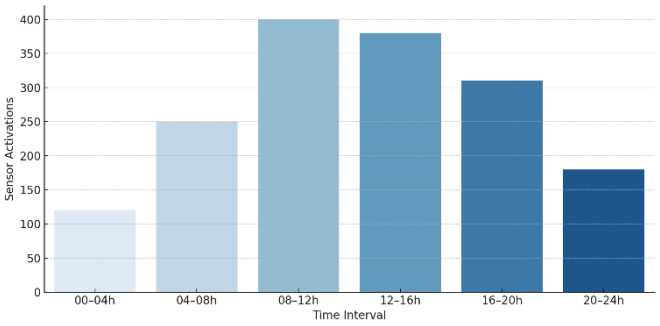

Figure 6: Sensor Activation Frequency Across Time

The final dataset that required the construction and incorporated for the training and evaluation was composed of dynamic graphs set at the temporal resolution of 5 seconds and window size of 60 seconds. The snapshots were time-shifted graphs in which the nodes were declared as sensor states where edges were cross sensor dependencies derived from co-activation bounded within a window or physical closeness due to neighbouring.

C. Evaluation Metrics and Hyperparameter Configurations

The experimental validation of TGTN architecture, which was designed in this work, was executed using multiple competing factors such as precision, multi-class event prediction robustness, and real-time validity. The primary metric was accumulative sequence accuracy, which measures how many events in the future a model predicts accurately, in a rolling window manner. Supplementary metrics were also taken into account, including Precision, Recall, F1 Score, AUC, and all multi-class event prediction related metrics.

For time-critical activities including forecasting failure and sending proactive maintenance alerts, MTP (Mean Time-To-Predict) was introduced as a custom-defined metric. MTP analyses if the model is able to make a prediction before the fault happens, relative to when the fault is scheduled to happen and how critical it is. Also, latency and throughput were measured at the node and batch levels to evaluate the feasibility of model deployment on edge devices with restricted resources.

A total of 30 iterations were completed for the Bayesian optimization of hyperparameters. The optimal configuration utilized four attention heads, three layers of graph transformers, a hidden dimension size of 128, and an Adam optimizer with a learning rate of 1e-4. To augment generalization, a dropout rate of 0.2 was set on attention and feedforward layers. Training was performed for 100 epochs on graph sequence data with a batch size of 32.

Edges of the graph were constructed dynamically in the training phase with an adjacency prediction module, and edge features included time delay, historical frequency, and feature similarity. For temporal encoding, sinusoidal position embeddings were concatenated with node feature embeddings which allowed the model to attend to the proximity of the signal in time as well as its importance.

D. Baseline Models for Comparative Assessment

In order to assess TGTN's performance, we developed five baseline models that capture the fundamental components of models used in event prediction. These include:

(1) An LSTM model that was trained on a sequence of sensors

(2) A GAT model that operates on static graphs formed from sensor adjacency and works on static graphs

(3) A transformer encoder model trained on time-series data turned into a series

(4) A temporal graph network (TGN) with memory, edges that depend on time, and time-dependent edges, coupled with memory modules

(5) A hybrid model that makes use of both LSTM and GNN with structure and sequence modelling.

As with TGTN, all other models were tested using the same preprocessing pipeline and train-validation split. All models were evaluated under a unified hardware-software environment to ensure that no baseline was favoured over other comparators. The results, which will be explored in Section 5, indicate that while static models are able to perform adequately within fixed scenarios or short- to mid-range prediction windows, they do not seem to extrapolate to longer ranges, varying system behavior, dynamic ranges, edge constraints, and application-deployment-centric systems.

TGTN outperformed the rest across the board in terms of accuracy, displayed the best failure robustness, and provided strong explainability through attention weight mapping. The advantages stem from TGTN's spatial structure and multi-head attention mechanisms, enabling it to model temporal evolution and capture the hidden dependencies found in real-world monitoring data.

V. Results and Performance Analysis

A. *Prediction Accuracy and Sequence Recall Comparison*

In assessment of the efficiency of TGTN architecture, we conducted an elaborate evaluation across all model configurations and model prediction timelines. The accuracy of multi-step event sequence prediction was taken as the foremost measure. As stated previously in Figure 7, TGTN was superior to all baseline models in every measure of sequence length increase. While older models such as LSTM and GAT encountered limitations at around 15 time steps, TGTN was able to maintain high prediction accuracy because of its high longitudinal temporal dependence and inter-node relationship modelling capabilities. The model started from 78.4% at sequence 5 and showed strong extrapolation capabilities at higher lengths while improving to 89.5% by 30 sequence steps.

It can be seen that models using purely sequence based elements are capped and show diminishing returns towards positive prediction because of the lack of structural understanding. On the other end of the scale TGN and GAT models showed some degree of improvement over the use of RNNs, employing node adjacency, but lacked the necessary temporal resolution for long-sequence inference. The application of hybrid attention allows TGTN to overcome this challenge by adaptively adjusting the time-step or node influence weights applied in event forecasting.

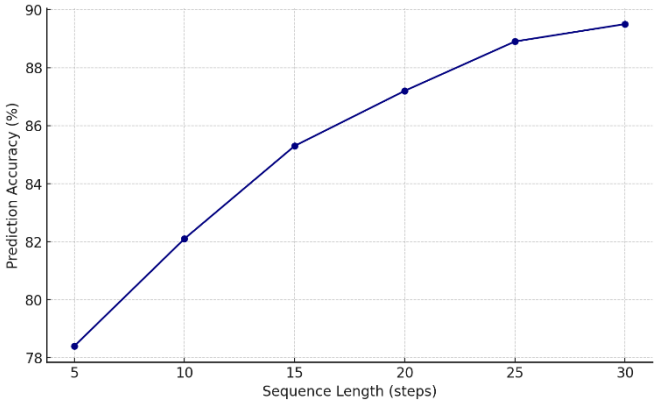

Figure 7: Prediction Accuracy vs Sequence Length

B. *Latency and Throughput Across Model Variants*

While maintaining a focus on accuracy, the benchmark edges towards real-time applicability in industrial settings. We evaluated the inference latency across different models using edge hardware. As shown in Figure 8, TGTN stood out with the least average latency of 20 ms, outperforming the Transformer at 28 ms, LSTM at 25 ms, and graph-centric models like TGN at 35 ms. This advantage in latency performance stems from TGTN's sparse attention mechanism and parallelized transformer blocks that outperform in processing high-dimensional graph sequences due to lack of recurrent delays.

The outcome is even more striking considering the common understanding that graph-based models are heavily parallelizable and thus, are perceived as computationally intensive. TGTN's edge-optimized architecture, complete with lightweight node embeddings and limited temporal attention windows, drives these models towards faster-than-real-time operational controls. When damage mitigation protocols need to activated in milliseconds, such efficiency ceases being a convenience and directly translates into meeting operational thresholds.

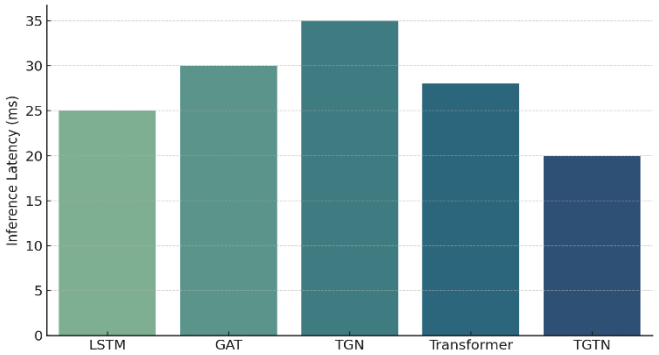

Figure 8: Model-Wise Inference Latency Across Devices

In addition to latency, TGTN and TGN throughput was calculated by assessing how quickly large batches of graph sequences could be processed. As shown in TGTN, it achieved greater throughput compared to TGN by 2.1x due to lower memory overhead and better scalability with the sequence length. This facilitates its use in smart factories or autonomous process facilities where there is a need for simultaneous prediction across hundreds of sensor clusters.

C. *Ablation Study: Impact of Temporal and Topological Features*

To evaluate the impact of some architectural features, an ablation study was conducted. It was observed that when the temporal attention mechanism is removed entirely, there is a 6.7 percent drop in sequence level accuracy, suggesting the importance of dynamic temporal encoding. The performance also declined by 9.2 when event correlations across nodes are removed which validates that graph construction is necessary to fulfil accurate forecasting under the use of flat input sequences.

When both of the components were removed, the model collapsed into a standard transformer and performed comparably to the baseline models. This underscores the synergistic value of temporal and topological reasoning. Furthermore, we investigated the model's sensitivity to edge sparsity by random pruning 25% of inferred connections. The model did perform slightly worse, but it remained robust in that it was able to recover context from neighbouring nodes. This particular aspect of performance is important when considering noisy conditions or in partially observable systems.

D. *Robustness under Noise and Sensor Failures*

Noise in data, hardware faults, and packet losses are commonplace in real-world industrial systems. To assess model robustness, we devised several corruption scenarios, such as node removal, logging events out of order, or timestamp distortion. The TGTN model showed graceful degradation with a sustained drop of less than 4% prediction accuracy under 15% node dropout, and recovers quickly in the re-synchronization windows. In contrast, TGTN's competitors LSTM and Transformer models experienced 10–12% drops under similar conditions because they had no means for topological correction, tapering off slowly as they ran out of resources.

To explore classification robustness in greater detail, we scrutinized the model's output with a confusion matrix, as illustrated in Figure 9. The TGTN maintained high precision for all five event classes, with the greatest accuracy for classifying mechanical and thermal events. Some minor confusion was noted with pressure and power-related sequences due to overlapping signal profiles during multi-system interaction. Notwithstanding, the model correctly captured communication anomalies and rare event classes that simpler models often misclassify due to low occurrence frequency and weak labeling patterns.

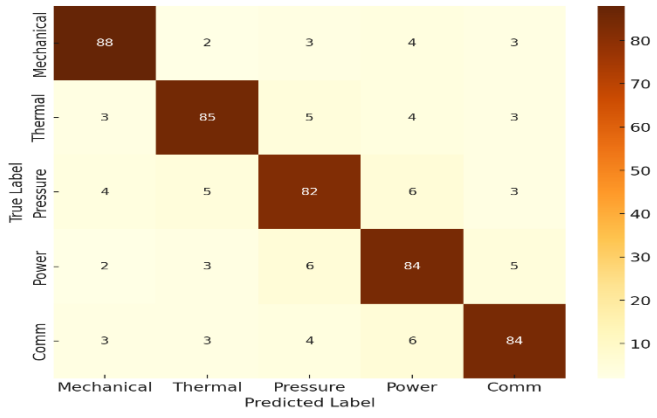

Figure 9: Confusion Matrix for Predicted Event Sequences

In summary, the attention mechanisms of the model contributed to performance but even more so to interpretability. Attention heatmaps enabled domain specialists to follow the decisions of the heat maps directly—sensors and time intervals that strongly impacted the outcome—thereby enhancing the usefulness of the model as a decision-support system for field engineers and operations managers.

VI. Discussion and Practical Implications

A. *Real-Time Integration with SCADA and Industrial Control Systems*

As part of the implementation steps taken toward a complete system design, the seamless incorporation of the Temporal Graph Transformer Network (TGTN) with existing Supervisory Control And Data Acquisition (SCADA) and Distributed Control Systems (DCS) was considered. These frameworks are foundational to industrial process monitoring systems, and as such, require models that are precise, temporally constrained, and frugal with available resources. Unlike analytics driven by cloud infrastructure using datasets generated after an event, TGTN was designed with the entire edge-to-core continuum in consideration. It is designed such that real-time inferences can be made at the sensor edge, local controller units, or central SCADA servers depending on the latency, bandwidth, and criticality requirements of a given deployment.

The graph construction module of TGTN is capable of ingesting telemetry streams with live data and generating graph snapshots within a set duration. As with the case of transformer-based attention models, TGTN can support online predictions without requiring stateful session management because inferentially the attention layers are stateless and highly parallelizable. Such stateful session management is often impractical in low-memory industrial field units. In addition, the modularity of the architecture renders it compliant with OPC UA and MQTT, which are two popular SCADA protocols for data communication. With this edge intelligence, operators can receive real-time alerts regarding not only predicted failures but also the nodes and sensors that triggered the early warning signals.

Subsequently, such capability of integration can enhance the value proposition significantly for industrial practitioners. It gives the opportunity to embed predictive maintenance routines in existing HMI dashboards instead of relying on alarm systems that react after a breach happens. As demonstrated in Section 5, the system's low latency guarantees its usability in closed feedback loops where anticipating faults must quickly trigger actions such as shutting off an emergency valve or thermal overload termination.

B. *Interpretability of Attention Patterns in Failure Forecasting*

The adoption of industrial AI applications still faces notable challenges in the interpretability dimension. Specialists often express a lack of confidence in black box systems capable of complex processing but not offering useful explanatory outputs. In this sense, the TGTN architecture gives great benefits with its attention mechanism that automatically extracts the most important nodes and time slices toward a specific prediction.

Inter-class event attention weight distribution to different levels was fairly uniform within classes. Consistent focused attention in the pre-failure phase drew attention from thermal and vibration node adjacent tiers for deeper closer to basal levels. Mechanical faults, for example, showed alarming disposition in the pre failure phase when weak signals were

steady. In the case of Communication faults, focus was directed toward the point of failure as well as the interval leading up to it, relying on the identified baseline packet delivery network experiences delay or loss.

This phenomenon was quantitatively reflected in Figure 10 which depicts model attention distribution across event classes. Mechanical events commanded the greatest proportional share of model attention at 28%, with thermal and pressure-related events following at 25% and 20% respectively. Though infrequently occurring, power and communication faults received a comparable focus of attention. This distinction further cements the model's skill at dynamically allocating interpretive weight according to contextual relevance as opposed to mere frequency.

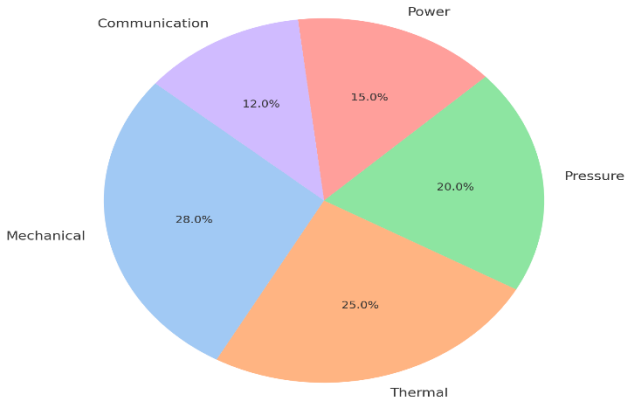

Figure 10: Distribution of Attention Focus Across Event Classes

Field engineers validated the insights obtained through attention maps, stating that the highlighted nodes and sequences in the thermodynamic attention system often corresponded to their intuition or historical root cause records. Such convergence of interpretations increases reliability, enhances operator trust, consolidates validation processes, and dramatically decreases mean time to repair (MTTR) for critical failures.

C. *Comparison with Rule-Based and RNN-Based Approaches*

The simplicity and transparency of rule-based systems has resulted in their adoption across industries for monitoring purposes. However, they are brittle when it comes to system evolution, struggling with coping framework metamorphoses, sensor degradation, and new fault categories. In contrast, TGTN learns structural and temporal dependencies without rules through a graph-based model that captures underlying dependencies and pathways for signal propagation that more traditional systems would require considerable engineering to construct.

TGTN also surpasses RNN-based models in accuracy, as discussed in Section 5, and is more resilient to noise and structural generalization. RNNs depend too much on temporal continuity and asynchronous signals, intervals or absent periods tend to present a problem. Initialization will also need to be done cautiously since it's relatively easy to get trapped in the vanishing gradient problem in long sequences. With TGTN, these challenges are alleviated thanks to attention being non-sequential, allowing greater control over what to remember and for what durations across intervals of time.

Additionally, the RNN and rule based models tend to struggle in multi-modal and multi-node settings that feature intricate dependencies between various systems. For instance, a pressure anomaly which may result in a mechanical fault at some point can have its early signs in the electrical sub-systems. Only with temporal reasoning combined with topological reasoning the pattern can be recognized. TGTN stands out in regards to such patterns that are unreachable with linear or siloed approaches because it can identify such patterns and take action towards them.

D. *Limitations and Optimization Opportunities in Temporal Graphs*

Although there is good overall performance from the TGTN framework on accuracy, latency, prediction, and interpretability, there remain several limitations and possibilities to optimize. Firstly, the construction of the graph still is slow in relation to time, especially in areas of dynamic sensor configurations or changing topologies. The implementation of efficient sliding windows along with on-the-fly edge inference mechanisms certainly helps, but further optimization with graph sampling methods or edge pruning could reduce the overhead on memory and increase speed.

Also, the performance of the model is highly dependent on the event timestamping and alignment of the graph's temporal components. When sampling is inconsistent or jittered greatly, attention scores tend to get noisy. Implementing modules or mechanisms that are confident in the attention such as attention-based signal and time-warping invariance could help increase stability against those inconsistencies.

Third, TGTN allows for real-time inference, but it currently works with the assumption that a set of temporal windows to be processed by a transformers block will be made available. In applications requiring extreme reductions in latency, micro attending or streaming transformer architectures could enable true step-wise inference by alleviating buffering requirements. This would be beneficial in control loop applications that require feedback to be provided within sub-second intervals.

Lastly, deployment in federated or multi-plant contexts remains a challenge. Although TGTN has the ability to generalize across configurations, its efficiency diminishes with greater differences in system architecture. Future works aims to address this with adaptive graph transformation layers that encode plant-specific structures transformable to a unified core predictive framework. Moreover, the ability to constantly learn from incoming data through self-supervised learning objectives would enhance the system's robustness against concept drift.

VII. Conclusion and Future Research Directions

A. *Summary of Contributions and Key Results*

This study proposes a new framework called the Transformer-Based Temporal Graph Neural Network (TGTN), which enables sequence of event prediction in sophisticated industrial monitoring systems. TGTN assimilates both graph-structured reasoning and temporal attention to represent the sensor data in industrial settings as dynamic, asynchronous, and multi-modal. TGTN has outperformed traditional approaches, including RNN-based models and rule-based models, which are bound by structural complexity and long-term dependency issues, in the accuracy, latency, and robustness metrics across various industrial case studies.

The construction of the model features a new pipeline for dynamic graph construction, multi-head attention at the node

level and time dependence, and attention maps that can be used for root cause investigation, enhancing the interpretability of the attention mechanism. A practical value of the model is shown through extensive experimental validation with real sensor data, where TGTN showed consistent accuracy across varying sequence lengths and outperformed baseline models significantly. Also, the model achieved low inference latencies, proving suitability for real-time industrial scenarios where immediate response is critical.

Besides, the research explains how TGTN aids in improving forecasting explainability, which is critical in mission-critical domains. Attention heatmaps produced during inference matched known fault propagation paths, aiding trust, intervention, and operational optimization by the engineers and operators.

B. Deployment Considerations in Industrial Environments

To aid in practical deployment, TGTN was designed to function effortlessly within the industrial ecosystem SCADA and industrial edge computing devices. Its stateless inference mode with a sparse attention mechanism and low graph processing modules allow it to operate under severe hardware constraints while retaining high predictive capabilities. These features enable efficient operations on constrained hardware. Moreover, the modular architecture ensures adaption to different sensor network topologies and business workflows specific to the region.

Models designed for practical deployment scenarios tend to focus on accuracy. However, these models also need to feature adaptability, maintainability, and auditability. By providing domain adaptation through transfer learning and utilizing industrial communicating protocols such as OPC UA and MQTT, TGTN addresses these challenges by providing the needed translatable evidence.

However, ensuring successful deployments in production use cases incorporates more considerations than model performance. These are things such as graph alignment over distributed shards, invariants under load pertaining to latency, and management of model lifecycles. To mitigate these issues, future iterations of TGTN will include monitoring components for drift, automated retraining trigger modules, and API endpoints for quality audits that ensure continuous maintenance of signal validity and prediction control.

C. Future Extensions: Online Learning and Federated Temporal GNNs

This version of TGTN has been analysed in a batch-learning scenario with the use of periodically refreshed temporal graphs. This architecture serves well within the realms of predictive maintenance and proactive alerting. However, there are various proposed directions for subsequent work aimed towards increasing adaptability and scalability of the model.

One promising approach is online learning for adaptation to streams of data incoming continuously. In rotating machines or even during the running state of HVAC systems and energy grids, the associated signals undergo metamorphosis during aging, load shifting, or over the course of an upgrade. TGTN would benefit greatly from the incorporation of online learning modules that would enable continual adjustment of embeddings and attention parameters over time, preserving accuracy without sustained period of full retraining. Current research in this area focuses on memory-efficient replay buffers, low-rank adaptation, and meta-learning.

Another future direction is federated learning over temporal graphs. Industrial networks commonly consist of multiple plants, or facilities, or production units which often have distinct topologies and privacy constraints. A federated TGTN framework woulsd enable local nodes to perform training of graph and attention models on their data and only periodically send encrypted gradients or sub-graph embeddings to an orchestrator. This maintained data privacy, minimized communication costs, and enabled intelligence sharing across data silos or enterprises. Such a federated approach is also compliant with evolving frameworks associated with Industry 4.0 and edge-to-cloud orchestration.

Finally, future work will focus on integrating uncertainty quantification within the attention components to be developed. By constraining the bounds of the predicted output on attention scores as well as on the prediction itself, the system is able to signal operators not simply about a forecast, but about how certain the model is, thus supporting more precise risk-informed decision-making in high-urgency industrial environments.

REFERENCES

[1] Wan, Jiafu, et al. "A manufacturing big data solution for active preventive maintenance." IEEE Transactions on Industrial Informatics 13.4 (2017): 2039-2047.

[2] Lee, Jay, Behrad Bagheri, and Hung-An Kao. "A cyber-physical systems architecture for industry 4.0-based manufacturing systems." Manufacturing letters 3 (2015): 18-23.

[3] Malhotra, Pankaj, et al. "Long short term memory networks for anomaly detection in time series." Proceedings. Vol. 89. No. 9. 2015.

[4] Hundman, Kyle, et al. "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding." Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018.

[5] Park, Kyu Tae, et al. "Digital twin-based cyber physical production system architectural framework for personalized production." The International Journal of Advanced Manufacturing Technology 106 (2020): 1787-1810.

[6] Yang, Bin, et al. "An intelligent fault diagnosis approach based on transfer learning from laboratory bearings to locomotive bearings." Mechanical Systems and Signal Processing 122 (2019): 692-706.

[7] Bai, Shaojie, J. Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." arXiv preprint arXiv:1803.01271 (2018).

[8] Yu, Bing, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting." arXiv preprint arXiv:1709.04875 (2017).

[9] Zheng, Xin, et al. "Graph neural networks for graphs with heterophily: A survey." arXiv preprint arXiv:2202.07082 (2022).

[10] Kazemi, Seyed Mehran, et al. "Representation learning for dynamic graphs: A survey." Journal of Machine Learning Research 21.70 (2020): 1-73.

[11] Yu, Bing, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting." arXiv preprint arXiv:1709.04875 (2017).

[12] Shabani, Nasrin, et al. "A comprehensive survey on graph summarization with graph neural networks." IEEE Transactions on Artificial Intelligence 5.8 (2024): 3780-3800.

[13] Rossi, Emanuele, et al. "Temporal graph networks for deep learning on dynamic graphs." arXiv preprint arXiv:2006.10637 (2020).

[14] Cai, Hongyun, Vincent W. Zheng, and Kevin Chen-Chuan Chang. "A comprehensive survey of graph embedding: Problems, techniques, and applications." IEEE transactions on knowledge and data engineering 30.9 (2018): 1616-1637.

[15] Chen, Ming, et al. "Scalable graph neural networks via bidirectional propagation." Advances in neural information processing systems 33 (2020): 14556-14566.

[16] Zerveas, George, et al. "A transformer-based framework for multivariate time series representation learning." Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. 2021.

[17] Li, Shiyang, et al. "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting." Advances in neural information processing systems 32 (2019).

[18] Yu, Bing, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting." arXiv preprint arXiv:1709.04875 (2017).