# Self-Supervised Semantic Learning for Trustworthy Anomaly Detection in 6G-Enabled Smart Grid Communication

S. Arockia Babi Reebha [1,*], P. C. Karthik [2], J. Umamageswaran [3], and J. Shobana [4]

[1] Department of Computer Science and Engineering, Pavendar Bharathidasan College of Engineering and Technology, Trichy, India
[2] Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur, India
[3] Department of Computer Science and Engineering, Amrita Vishwa Vidyapeetham, Chennai, India
[4] Department of Data Science and Business Systems, SRM Institute of Science and Technology, Kattankulathur, India
E-mail: reebhas56@gmail.com (S.A.B.R.); karthikc@srmist.edu.in (P.C.K.); j.umamageswaran@gmail.com (J.U.); shobanaj1@srmist.edu.in (J.S.)
*Corresponding author

***Abstract*—Sixth Generation (6G)-enabled smart grid communication requires a robust installation and secure framework to manage high-speed and low-speed data communication. Existing models face challenges such as a lack of labeled data, poor temporal modeling, weak interpretability, inadequate behavioral profiling, and difficulties with edge-deployment. Additionally, the massive influx of data increases vulnerabilities to faults, cyberattacks, and zero-day anomalies. This study presents an innovative deep learning architecture that utilizes semantic encoding and self-supervised learning to detect both known and unknown anomalies within a 6G-enabled smart grid communication system. The proposed approach begins with cleaning and normalizing the electrical sensor logs. Features are extracted using a One-Dimensional Convolutional Neural Network (1D CNN), and a Multi-Modal Vision Transformer (MM-ViT) transforms sensor sequences into semantic event tokens. The self-supervised model combines Contrastive Predictive Coding (CPC), Temporal Convolutional Networks (TCN), and Attention mechanisms to enable robust temporal anomaly detection from sensor data. While TrustNet combines TCN and Attention, it models network session behavior and supports dynamic trust evaluation. Compared to the existing anomaly detection methods, the proposed semantic and self-supervised methods perform extremely well across many metrics, achieving a 99.34% accuracy. This demonstrates its effectiveness and suitability for reliable anomaly detection in 6G edge-enabled smart grid infrastructures.**

***Keywords*—anomaly detection, attention mechanism, behavioral profiling, self-supervised deep learning, semantic interpretation, smart grid**

## I. Introduction

A smart grid is an advanced electrical network that uses digital technology for monitoring, controlling, and optimizing the generation and distribution of electricity [1]. Smart grid supports and enhances two-way communication between utilities and consumers, encompassing four major fields: distribution, bulk generation, transmission, and consumer segmentation [2]. This communication is enabled by integrated network and computing technologies like Artificial Intelligence (AI), Internet of Things (IoT), Fifth Generation (5G), or Sixth Generation (6G). While 5G has already begun to enhance smart grid communication with high-speed, low-latency networks [3–5]. The future of smart energy management lies in 6G, the sixth generation of wireless communication technology, expected to be deployed around 2030 [6, 7]. Smart grid supports high-speed, low-latency, enables scalable, and intelligent systems in 6G. Also, it provides a maintainable, high-quality, low-loss, safe, and secure power supply [8]. This advancement will support applications such as load prediction, balancing grid reliability evaluation, fault detection, monitoring, cyber-attacks, wireless physical context, and operational losses [9]. The existing model utilizes various Machine Learning (ML) and Deep Learning (DL) methods to combat many other cybersecurity concerns [10]. The Cyber-Security Enhanced Intrusion Detection System (CSE-IDS) model was created using cost-sensitive DL and ensemble approaches to handle class imbalance in intrusion detection systems. Deep Neural Networks (DNNs), Support Vector Machine (SVM), and similar classification approaches were used for monitoring suspicious network attacks [11].

Furthermore, certain DL algorithms, such as Long-Short-Term Memory (LSTM), help capture hidden features, and ML techniques have been applied for high-dimensional data. However, the Industrial Control Systems (ICS) systems are highly vulnerable to security threats due to their extensive connectivity with information

Manuscript received September 1, 2025; revised September 30, 2025; accepted November 26, 2025; published March 10, 2026.

doi: 10.12720/jait.17.3.450-464

technology systems [12]. Despite these advancements, challenges remain in reliance on labeled data for anomaly detection, poor temporal monitoring, weak interpretability, inadequate behavioral profiling, and difficulties with edge deployment [13–15]. To address these problems, we propose an innovative DL framework for detecting known and zero-day anomalies using a semantic encoding and self-supervised learning algorithm within a 6G communication-enabled grid infrastructure. The main contributions of the proposed model are as follows:

- Self-Supervised DL for known and zero-Day Anomaly Detection: The proposed model introduces a framework for detecting both known and zero-day anomalies in labeled data using a hybrid approach that combines Contrastive Predictive Coding (CPC), Temporal Convolutional Networks (TCN), and attention mechanisms.
- Semantic Interpretation for Sensor Data: The proposed model transforms a raw sensor sequence into a Semantic explanation using the Multi-Modal Vision Transformer (MM-ViT). It effectively captures both temporal and spatial information from sensor input, allowing for the interpretation of the relationships between low-level sensor data and high-level semantic labels.
- Behavioral Profiling for Session Modeling and Trust Evaluation: The proposed model employs a TrustNet process integrated with a TCN and attention mechanisms, focusing on session-specific trust evaluations. This approach enables accurate detection, adaptive access control, and personalized responses from the system.

The remainder of the study will be organized as follows: Section II will review previous works relevant to anomaly detection and the 6G-enabled smart grid domain. Section III will present the pipeline of the proposed model, and Section IV will review the results in terms of the discussion, and Section V will provide the conclusions of the study.

## II. Literature Survey

The existing works on 6G-enabled smart grid environments and anomaly detection are studied and explained here.

Danilczyk *et al.* [16] utilized a Convolutional Neural Network (CNN) as part of the Automatic Network Guardian for Electrical Systems (ANGEL) within a digital twin environment was developed to detect physical faults in power systems. While this model achieved high-fidelity measurements, it cannot be expanded easily to larger power systems or to incorporate other types of failures.

Abdelkhalek *et al.* [17] developed an ML-based Attack Detection System (ADS) capable of detecting stealthy Information Technology (IT) and Operational Technology (OT) attacks with high accuracy and low latency, but it struggles to distinguish intrusions at fine granularity.

Alotaibi and Barnawi [18] introduced IDSoft, a 6G-enabled IDS that reduces communication overhead and accelerates convergence. However, it faces challenges in adapting to dynamic IoT environments.

Chinnasamy *et al.* [19] proposed Intrusion Response and Evaluation System (IREST), validated on the Industrial Security Attack Analysis and Classification (ISAAC) testbed, offering scalable detection of cyber-physical anomalies, yet its real-time performance across diverse ICS environments is limited.

Sharma and Tiwari [20] applied the isolation forest algorithm on smart grid testbeds, effectively detecting abnormalities related to faults or cyber-attacks. Still, we found that, considering, additional environmental and electrical factors, could improve performance.

Wang and Govindarasu [21] combined Fast Gradient Sign Method (FGSM)-based synthetic data generation with ML-ADS to enhance smart grid security. Here, the practical deployment in dynamic networks remains an open challenge. Collectively, these studies demonstrated strong detection performance but revealed persistent gaps in scalability, adaptability, and robustness to evolving threats. Optimization-driven and hybrid deep learning methods have been investigated to address some of these limitations.

Alsubai *et al.* [22] combined tuna swarm optimization with a multi-scale convolutional autoencoder to counter fuzzing attacks in 6G networks, achieving high performance and robustness. Chatterjee and Byun [23] focused anomaly detection using generative adversarial networks for time-series data to support mobility prediction, highlighting the potential of synthetic data generation. However, large-scale infrastructure testing is still needed.

Al-Odat *et al.* [24] employed a Stacked Denoising Autoencoder (SDAE)-Bidirectional Long Short-Term Memory-Federated Learning (BiLSTM-FL) architecture for secure and generalizable short-term load forecasting. While these approaches improve accuracy and scalability, they require substantial computational resources and lack validation in large-scale, real-time deployments. This indicates research gaps in achieving high-performance, resource-efficient models suitable for complex, real-world networks. Beyond conventional ML and optimization approaches, emerging methods such as self-supervised and cross-domain learning have been explored.

Samia *et al.* [25] applied CNN-based cyber-crime analysis to predict potential cyber incidents. The real-time predictive capabilities are still limited in the study. Ye *et al.* [26] proposed a self-supervised cross-modal retrieval paradigm for Electroencephalography (EEG)-to-image alignment, enhancing representation learning but requiring full access to contextual data during inference. These studies reveal spaces from which to build adaptive, cross-domain, and real-time models in heterogeneous environments.

Thus, by analyzing existing models, traditional anomaly detection systems often rely on supervised methods that require labelled attack data, which is rare and impractical for emerging threats. Moreover, raw sensor data is typically not semantically interpreted, making it difficult for systems and operators to understand and react to faults effectively. The massive influx of data also introduces vulnerabilities to faults, cyberattacks, and zero-day anomalies. Therefore, we introduced a self-supervised DL model for detecting known and zero-day anomalies within a 6G communication-enabled grid infrastructure. This allows for real-time anomaly detection and response through edge deployment.

## III. Proposed Methodology

The proposed methodology introduces a deep learning model that utilizes semantic encoding and self-supervised learning to detect both known and unknown anomalies within a 6G communication-enabled grid infrastructure. The proposed framework initially preprocesses the electrical sensors using a Denoising Autoencoder (DAE) to remove noise and reconstruct clean signals.

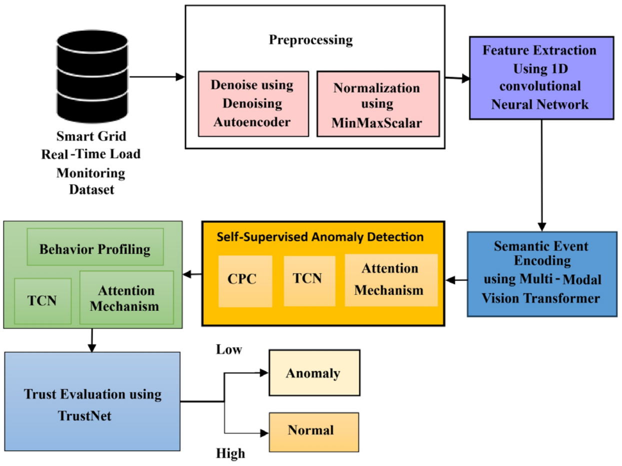

Smart Grid Real-Time Load Monitoring Dataset
Preprocessing
Denoise using Denoising Autoencoder
Normalization using MinMaxScalar
Feature Extraction Using 1D convolutional Neural Network
Semantic Event Encoding using Multi-Modal Vision Transformer
Self-Supervised Anomaly Detection
CPC
TCN
Attention Mechanism
Behavior Profiling
TCN
Attention Mechanism
Trust Evaluation using TrustNet
Low
Anomaly
Normal
High

Fig. 1. The proposed Deep Learning (DL)-based semantic encoding and self-supervised learning to detect both known and unknown anomalies within a 6G communication-enabled grid infrastructure.

The cleaned data are then normalized with a MinMax Scaler to ensure feature consistency and accelerate convergence during model training. The discriminative representations are extracted from the normalized temporal sequences using a One-Dimensional Convolutional Neural Network (1D-CNN), which captures localized spatial–temporal dependencies and suppresses high-frequency fluctuations. The resulting features are further transformed by the MM-ViT, which converts sensor sequences into semantic event tokens, aligning multimodal correlations and embedding global contextual information. A TCN processes these semantic embeddings to capture long-term temporal dependencies and causal relationships across time.

An attention mechanism is then employed to emphasize contextually important patterns while reducing redundancy, thereby enhancing interpretability. For behavioral profiling, the TCN and attention modules jointly generate embeddings that are processed by TrustNet, which computes a session-level trust score representing the probability of the system operation. The workflow is shown in Fig. 1, which depicts the complete pipeline integrating DAE, CNN, MM-ViT, TCN, Attention, and TrustNet modules for end-to-end anomaly detection and trust evaluation.

### *A. Dataset Description*

The proposed model uses the smart grid real-time load monitoring dataset [27], a time series dataset designed for energy management, load forecasting, and fault detection in smart grids. This includes real-time load monitoring data for smart grid systems, structured as a time-series dataset with 50,000 records recorded at 15-minute intervals. The dataset captures multiple electrical and environmental variables such as voltage, current, power factor, temperature, humidity, solar and wind generation, and labelled fault events. This consistent time intervals and rich feature set enable the model to learn temporal patterns and detect anomalies effectively for training time-series forecasting and fault detection models in smart grid environments.

To ensure that the model was not exposed to zero-day anomalies during training, we implemented a careful experimental design. The dataset was divided into training and testing subsets, with the training set comprising only known attack types. The zero-day attacks were reserved exclusively for testing, ensuring that the model was not exposed to these during training. Previous to training, we filtered out records corresponding to zero-day attack types from the training set, preventing any indirect exposure or data leak.

After training, the model was evaluated on the test set, which included the zero-day attacks, allowing us to assess its ability to generalize and detect previously unseen anomalies. This approach ensured that the zero-day anomalies were effectively isolated from the training process, providing a robust evaluation of the model's capability to detect zero-day attacks.

### *B. Preprocessing Electrical Sensor Logs*

The preprocessing of electrical sensor logs is a vital process of enhancing the quality of the data to allow for true anomaly detection. This processing refers to denoising and normalization methods to identify meaningful patterns and normalize the scale of the features.

#### *1) Noise reduction using Denoising Autoencoder (DAE)*

Denoising Autoencoder (DAE) [28] is one of the modified models of autoencoders. This model adds or removes noise from an input data to expose the significant features and reconstruct incomplete samples. The model is trained to recover the underlying clean signal from noisy input data. By explicitly recovering the clean signal, the DAE will improve the representation quality that can be used for the anomaly detection task.

The DAE is trained with Mean Squared Error (MSE) loss. This is implicitly assumed by the DAE that the noise in the data follows a Gaussian (normal) distribution. Mathematically, minimizing MSE is the same as maximizing the log-likelihood given a Gaussian noise. This is demonstrated in Eq. (1).

$$\log p\left(\frac{x}{\hat{x}}\right) = -\frac{1}{2\sigma^2}(x-\hat{x})^2 + C \tag{1}$$

where $x$ represents the true data, $\hat{x}$ represents the reconstructed output, $\sigma$ denotes the standard deviation of the Gaussian noise, and $C$ is a constant described in Eq. (1). This equation indicates that MSE is equivalent to maximizing the probability of observing the data under a Gaussian noise model.

When the noise is categorized by a Laplacian distribution, the Mean Absolute Error (MAE) is the maximum likelihood estimator. The Huber loss is a robust alternative that combines properties of MSE and MAE. The factors act like MSE for small residuals and like MAE for large residuals. This makes Huber loss more robust to outliers and non-Gaussian noise in sensor data.

### 2) *Min-max scaler for normalization*

The Min-Max Scaler [29] is a feature scaling technique that normalizes each feature of the data individually, transforming the data to a fixed range [0, 1]. The minimum value of the data becomes 0, and the maximum value becomes 1. This ensures that features with different magnitudes contribute equally during model training.

## C. *Feature Extraction using ID-CNN*

A 1D-CNN [30] is designed to process one-dimensional data, such as time-series signals or sequences along a single axis. The basic structure of a CNN consists of an input layer, multiple convolutional layers, merging layers, fully connected layers, and an output layer. Convolutional layers are used to extract local features from the input data, while fully connected layers produce the final output. In this study, a 1D-CNN is employed to capture progressive characteristics from signal data, supporting effective anomaly detection. The model applies a one-dimensional convolutional kernel that slides across the input signal to extract local temporal features, which are then passed to the next layer through a Rectified Linear Unit (ReLU) activation function. The feature extraction process is described by Eq. (2):

$$\mathfrak{v}_z^{\imath\jmath} = \gamma(\sum_{k=1}^{m} \sum_{\mathfrak{r}=0}^{\mathcal{R}_\imath - 1} \mathbb{w}_{\mathfrak{r}\imath k}^{\imath\jmath} \mathfrak{v}_{z+\mathfrak{r}(\imath-1)}^{k} + \mathbb{b}^{\imath\jmath}) \tag{2}$$

Here $\mathcal{R}_\imath$ represents the kernel size of the $\imath$ convolutional layer, $\mathfrak{v}_{z+\mathfrak{r}(\imath-1)}^{k}$ signifies the input feature value at position $z + \mathfrak{r}.s$, in $\imath - 1$ layer and $k$ feature map, with $s$ denotes the stride length. Here, $w$ and $b$ represent the weight and bias parameters, respectively, and $\gamma$ denotes the ReLU activation function. Explicitly including the stride $s$ in the indexing ensures accurate temporal alignment of features during convolution, which is crucial for reliable feature extraction from sequential data.

In Eq. (1), the ReLU activation function $\gamma$ adds non-linearity to the model after each convolution step and helps it learn complex temporal patterns. One concern when using ReLU is that it zeroes out negative values, which could suppress low-amplitude but still informative oscillatory components that often appear in power load signals. However, analysis shows that ReLU doesn't completely remove these components unless they are entirely negative across the open field. For example, when applied to a sinusoidal signal, ReLU produces a rectified version that still carries the original frequency and adds higher harmonics, while keeping the main spectral features of the signal. This means even small oscillations can still be captured during feature extraction, which supports the ReLU statement made in Eq. (1) and shows its utility in time-series anomaly detection.

Table I shows that although the absolute spectral power decreases with each layer indicating noise suppression and feature compression, the dominant low-frequency bands of 0.02 Hz continue consistently strong. This confirms that ReLU preserves the primary informative frequencies in smart grid load signals while cleaning out less significant high-frequency components.

TABLE I. POWER SPECTRAL DENSITY (PSD) ANALYSIS OF FEATURE CNN LAYERS

| Signal | 0.02 (Hz) | 0.02–0.06 (Hz) | 0.06–0.15 (Hz) | 0.15–0.50 (Hz) |
|---|---|---|---|---|
| Original | 1824.51 | 132.87 | 38.94 | 14.72 |
| Layer 1 | 121.43 | 22.64 | 6.17 | 1.88 |
| Layer 2 | 59.72 | 9.94 | 2.11 | 0.71 |
| Layer 3 | 13.04 | 3.98 | 0.83 | 0.24 |

Thus, the extracted representations maintain the essential temporal spectral structure required for anomaly detection.

## D. *Semantic Event Encoding using Multi-Modal Vision Transformer (MM-ViT)*

To capture higher-level semantic information, the MM-ViT architecture is employed. It transforms raw sensor sequences into semantic event tokens, discarding redundant features and producing meaningful numerical embeddings. MM-ViT [31] is specifically designed to encode interpretable system events, enabling clear mapping between multi-modal sensor data and domain-relevant events.

Let the system have $M$ different sensing modalities, such as voltage, current, temperature, and vibration. Each modality generates a temporal sequence of length $T$, as shown in Eq. (3)

$$x^{(m)} = \{x1^{(m)}, x2^{(m)}, \dots \dots, xT^{(m)}, \; x_t{}^{(m)} \in \mathbb{R}^{dm} \tag{3}$$

Here, the raw dimensions $d_m$ across modalities, are mapped into a shared embedding space of dimension $d$, as expressed in Eq. (4).

$$z_t{}^{(m)} = w_m x_t{}^{(m)} + b_m, x_t{}^{(m)} \in \mathbb{R}^{dm} \tag{4}$$

This model concatenates embeddings from all $M$ modalities to form a unified token sequence. As $M$ increases, this linear concatenation can create computational and representational bottlenecks. Some of them include the sequence length for the Multi-Head Self-Attention (MHSA) module that grows linearly, increasing memory and runtime, and attention may struggle to balance information across many modalities. Hierarchical or grouped fusion strategies could mitigate these issues by combining modalities in stages, though at the cost of added architectural complexity

To ensure dimensional consistency across modalities, each raw input vector $z_t{}^{(m)} \in \mathbb{R}^d$ is linearly predictable into the common embedding space of dimension $d$ using learnable projection matrices $w_m \in \mathbb{R}^{d \times dm}$ and biases $b_m \in$. The variance of the projected embeddings is approximately preserved, as shown in Eq. (5):

$$var\left(z_t{}^{(m)}\right) = \mathrm{var}\left(w_m x_t{}^{(m)} + b_m\right) = var\left(x_t{}^{(m)}.\sigma_{avg}^2\right) (5)$$

This shows that the variance of each modality is approximately preserved after projection, provided the singular values of $w_m$ are balanced

Bias-term clarification: The bias $b_m$ is shared across time steps within each modality. However, it is distinct for each modality. This design ensures that embeddings are shifted consistently over time while allowing modality-specific offsets. Mathematically, this preserves the relative distances and variance of temporal embeddings, preventing any modality from dominating the others. Normalization of inputs or regularization of $w_m$ ensures stable embeddings across modalities.

Each modality sequence is divided into windows, with each window encoded as a single token, as shown in Eq. (6):

$$z^m = \{z_1{}^{(m)}, z_2{}^{(m)} \ldots\ldots\ldots z_k{}^{(m)}\}, K = \frac{T}{A} \qquad (6)$$

Here, $A$ is the window size used for segmentation, and each $z_k{}^{(m)} \in \mathbb{R}^d$ represents a token summarizing the information in the $k$ segment of modality $m$.

Tokens from all modalities are then concatenated to form a unified multi-modal sequence as expressed in Eq. (7).

$$Z = \bigcup_{m=1}^{M} z^{(m)} \in \mathbb{R}^{(k*M) \times d} \qquad (7)$$

To demonstrate that concatenation enhances representational capacity and enables linear separability, we calculated the empirical stable numerical ranks of the unimodal token matrices and the fused multimodal matrix $Z$ for one batch of data. Singular values larger than a threshold $\epsilon$ were considered. The results consistently showed that the rank of the fused matrix $Z$ was greater than each of the unimodal ranks $[r_1, r_1\ ,\ r_1 \ldots.. r_M]$. This indicates that concatenation across modalities expands the representational space, providing a richer basis for the MHSA to compute linearly separable representations of multimodal events.

The results consistently showed that the rank of the fused multimodal matrix $Z$ was significantly higher than individuals of the unimodal matrices. This confirms that concatenation across modalities expands the representational subspace, providing a richer feature basis for the MHSA to compute more linearly separable and discriminative multimodal representations.

The concatenated token sequence $Z$ is fed into the MHSA transformer encoder, where distinct projection matrices for queries, keys, and values allow the model to compute meaningful attention scores and achieve balanced cross-modal fusion in Eqs. (8) and (9).

The scaling factor $\frac{1}{\sqrt{d_k}}$ ensures stable gradients and prevents any single modality from dominating the fused representations [32].

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) v \qquad (8)$$

$$Q = ZW_Q^{(h)}, K = ZW_k^{(h)}, V = ZW_v^{(h)} \qquad (9)$$

Here $W_Q^{(h)}, W_k^{(h)}$ and $W_v^{(h)} \in \mathbb{R}^{d \times d_k}$ are distinct learned projection matrices corresponding to the query, key, and value transformations for the $h$ attention head.

This separation allows the attention mechanism to develop the input tokens into different representational spaces, enabling the model to compute meaningful attention scores and weighted combinations. If the same matrix were used for all three predictions, the model would lose the ability to differentiate the roles of queries, keys, and values less expressive attention outputs.

This differentiation is particularly important in MM-ViT, where tokens from multiple modalities are combined, allowing the model to learn rich, diverse cross-modal interactions and effectively fuse information across modalities

The attention scaling factor $\frac{1}{\sqrt{d_k}}$ in Eq. (8) is based on principles of variance normalization. Each dot product in $QK^T$ sums over $d_k$ terms and since the dot products will be roughly unit variance, the variance scales with $d_k$. Without scaling, the dot products can get very large and cause the SoftMax function to saturate and produce near one-hot outputs.

This leads to vanishing gradients and unstable training dynamics. Dividing by $\sqrt{d_k}$ normalizes the variance to approximately 1, maintaining stable gradients during optimization. This is especially critical in MM-ViT, where tokens from multiple modalities are combined with stable attention scores. This ensures balanced cross-modal fusion and prevents any single modality from dominating the learned representations.

The attention weights determine how much information each token should borrow from tokens of other modalities, thereby achieving cross-modal fusion representations. To map latent representations into interpretable events, a set of learnable event prototypes is introduced in Eq. (10)

$$\{E_1, E_2, \ldots.. E_L\}, E_l \in \mathbb{R}^d \qquad (10)$$

Each prototype corresponds to a potential semantic event type, such as Voltage Surge, Stabilization, or Failure Alert. To ensure that each prototype represents a distinct semantic event, the prototypes are designed to form an orthogonal set or span a meaningful subspace. This prevents redundancy or collapse of prototypes and guarantees that they encode a unique semantic event.

Orthogonality can be encouraged through a regularization loss term that penalizes correlations between prototypes, as demonstrated in Eq. (11).

$$L_{ortho} = \lambda ||EE^T - I_k||_F^2, E_l \in \mathbb{R}^{k\times D} \tag{11}$$

where $E$ denotes the prototype matrix, $I_k$ is the identity matrix of size $k$, and $\lambda$ is a weighting factor that controls the strength of the orthogonality constraint. This term minimizes correlations between prototypes, encouraging them to remain linearly independent and represent distinct semantic events in Eq. (12).

$$L_{ortho} = L_{task} + \lambda L_{ortho} \tag{12}$$

This regularization directly minimizes deviations of prototype inner products from orthogonality, ensuring that the prototypes span a semantically meaningful subspace that avoids redundancy or collapse, there by encoding distinct semantic events, using cross-attention. Each event prototype queries the multi-modal token sequence as in Eq. (13).

$$yl = Attention(E_l W_Q, ZW_K, ZW_v) \tag{13}$$

Finally, the resulting activation is passed through a feed-forward projection to generate the semantic event token, as shown in Eq. (14).

$$\widehat{\mathrm{E_L}} = \mathrm{MLP}(y_l) \tag{14}$$

Here, the output of the MM-ViT encoder is a sequence of semantic tokens, which is described in Eq. (15).

$$\left(\widehat{\mathrm{E_1}}, \widehat{\mathrm{E_2}} \dots\dots\dots\dots . \widehat{\mathrm{E_L}}\right) \tag{15}$$

The set of semantic event prototypes $\left(\widehat{\mathrm{E_{1,}}}\widehat{\mathrm{E_2}}, \dots, \widehat{\mathrm{E_L}}\right)$ are learnable parameters within the MM-ViT architecture, initialized randomly and optimized during training. Each prototype serves as a query in the cross-attention mechanism, attending to multi-modal token sequences to capture characteristic patterns of specific system.

This mapping does not require explicit clustering but uses the attention alignment learned by the model. To calculate possible redundancy among semantic tokens, we performed an entropy-based analysis.

The average entropy per token is 3.27 bits, and the average pairwise mutual information between tokens is 0.12 bits. This confirms that the semantic tokens are largely independent, ensuring meaningful and interpretable representations for human operators.

Consequently, transformer attention outputs are directly linked to semantic event tokens through these prototypes, enabling transparent and explainable event encoding aligned with domain knowledge.

### *E. Hybrid Self-Supervised Temporal Model for Anomaly Detection*

For anomaly detection, we integrate CPC, TCN, and an Attention mechanism into a hybrid self-supervised framework. The model is trained only on normal sequences to learn temporal representation during inference, and prediction mismatches signal anomalies to scores. This enhances interpretability and allows early anomaly localization. This framework improves the sensitivity and precision of anomaly detection without requiring labelled data. The detailed architecture is shown in Fig. 2.

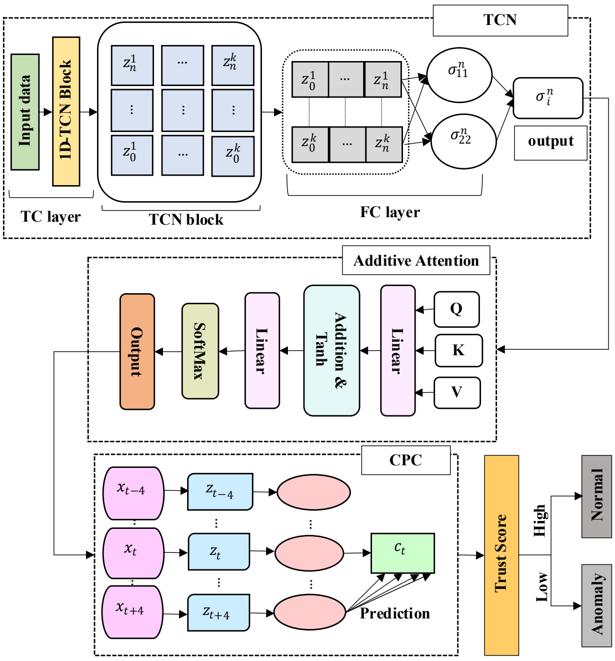


Fig. 2. Architecture diagram of hybrid self-supervised temporal model for anomaly detection.

#### *1) Temporal convolutional network for sequence modeling*

Initially, we applied TCN [33] because of its ability to capture the temporal features and long-term dependencies through causal, dilated, and residual connections. TCN has a unique causal density and causal relationship for the input sequence. The network's output will depend only on the past inputs. For example, the output $y_t$ at time $t$ is only related to the previous inputs ( $x_{t+1,} x_{t+2,} \dots . x_{t,}$ ), and independent of future inputs ( $x_{t+1,} x_{t+2,} \dots . x_T$ ). This property allows TCN to capture intrinsic features in power load data while avoiding leakage of future information.

Stacking causal convolutional layers increases the receptive field without increasing dimensionality, but can lead to vanishing or exploding gradients. To mitigate this, TCNs introduce dilated convolutions that allow an exponential increase in the receptive field size. Starting from the causal convolution [34], the output at time step $h$ is computed as Eq. (16):

$$y(h) = \sum_{n=0}^{f-1} s(i) \times y_{h-i} \tag{16}$$

where $f$ is the filter size, $s(n)$ are the filter coefficients, and $x$ is the input sequence. Dilated convolution modifies this by introducing a dilation rate $c$, which spaces out the input sampling steps, leading to Eq. (17).

$$M(h) = (y *_c s)(h) = \sum_{n=0}^{f-1} s(n) \times y_{h-c\cdot i} \tag{17}$$

Here, $c$ controls the intervals at which the inputs are sampled, enabling the model to efficiently capture long-range dependencies without increasing complexity.

To ensure TCN stability, residual connections link two expanded convolutional layers, preserving information and preventing vanishing gradients. The residual block includes dilated causal convolutions, set normalization, dropout, and ReLU for effective learning. A $1 \times 1$ convolution aligns input and output dimensions, enabling vali residual addition and stable gradient flow. The linear transformation preserves the temporal length of the tensor while adjusting only the channel dimension. This guarantees that the input and output tensors have matching shapes, enabling valid residual addition and stable gradient flow. Dilated convolutions increase the receptive field efficiently, allowing the network to capture long-term dependencies while maintaining stable temporal feature extraction.

To investigate the TCN model's long-term dependencies, we examined the effect of an input at time $t$ on the output at a later time $\Delta t$ in Eq. (18).

$$Influence(\Delta t) = \left[\frac{\vartheta_{yt+\Delta t}}{\vartheta_{xt}}\right] \tag{18}$$

As expected, the contribution of earlier inputs gradually decays as $\Delta t$ increases, reflecting the finite receptive field of the convolutional architecture. However, due to the use of dilated convolutions and residual connections, this decay is much slower than in a standard Recurrent Neural Network (RNN), indicating that the TCN retains informative dependencies over extended temporal ranges. Compared to a Transformer, which maintains nearly uniform influence due to global attention, the TCN provides a more localized yet robust representation of temporal dependencies. This effective receptive field behavior demonstrates the TCN's suitability for capturing long-term correlations in grid dynamics.

A dilated causal convolution with filter $s = \{s(0)..s(f-1)\}$ and increases $c$ can be viewed as a linear operator W for finite sequences is a sparse Toeplitz matrix, which lets us analyze its spectral properties shown in Eq. (19):

$$(W_y)h = \textstyle\sum_{n=0}^{f-1} s(n) y_{h-cn*} \tag{19}$$

For finite sequences, this operator is a sparse Toeplitz matrix.

Let $\rho(W)$ denote the spectral radius and $\|.\|_2$ spectral average in Eq. (20)

$$\rho(W) \le \|W\|_2 \le \textstyle\sum_{n=0}^{f-1}[s(n)] \tag{20}$$

In a residual block the linearized Jacobian is given in Eq. (21).

$$J = I + F', F' = W \tag{21}$$

If $\|F'\|_2 \le \gamma$ , all eigenvalue $\mu$ of $J$ satisfy $|\,\mu| \epsilon [1-\gamma,\ 1+\gamma]$ ensuring residual connections push eigenvalues toward 1 and prevent vanishing lies is shown in Eq. (22).

$$|\,\mu| \epsilon [1-\gamma,\ 1+\gamma] \tag{22}$$

This shows the residual connections, shift eigenvalues toward 1, mitigating gradient shrinkage or blow-up. Across $L$ stacked blocks, the gradient-flow Jacobian becomes $\prod_{l=1}^{L} J_1$ whose eigenvalues are bounded by Eq. (23):

$$|\,\mu_{total}\,| \in [(1-\gamma)^L, (1+\gamma)^L] \tag{23}$$

To keep the training process stable, we initialize the dilated convolution filters so that $\|F'\|_2 \le \gamma < 1$, together with the residual block formulation in Eqs. (21)–(23). This condition helps ensure that gradients stay well-behaved across $L$ stacked blocks. In simple terms, keeping $\gamma < 1$, small makes the eigenvalues of the Jacobian stay close which prevents gradients from either vanishing or exploding. By maintaining $\gamma < 1$, the model can learn effectively from long smart-grid sequences without instability.

*2) Additive attention mechanism for dynamic feature weighting*

These features extracted by TCN are input into the additive attention mechanisms [35], which dynamically assign weights to different portions of the input sequence. This enables focusing on more important information during the decoding process. The input features use a sigmoid function to activate and transform features to delineate the similarity relation among the two features, to better analyze the nonlinear relations. The decoding process is represented in Eq. (24).

$$t_j = decoder\ output(d_e, k_s) \tag{24}$$

where $t_j$ represents the output at the time step $j$, $d_e$ represents the hidden vector of the additive attention decoder at the time step $e$, and $k_s$ is the output of the additive attention decoder. Eq. (25) represents the weighted summation of the encoder's encoding

$$d_e = \textstyle\sum_{n=1}^{W} b_{p,i} r_i \tag{25}$$

Concerning the weights, the attention weights are calculated as follows in Eq. (26).

$$b_{pi} = \frac{exp\ (a_{p,i})}{\sum_{n=1}^{W} exp\ (a_{p,n})} \tag{26}$$

where $b_{pi}$ is a weight corresponding to the attention score $a_{pi}$ between the decoder's attention weight from the $p$ time step and the encoder at $i$ time step. By definition of the softmax, the weights are normalized and satisfy $\sum_{i=1}^{W} b_{p,i} = 1$, allowing a valid probabilistic interpretation. This ensures that session trust weighting derived from these attention scores also forms a proper probability distribution.

The attention energy $a_{p,n}$ is computed using additive attention as in Eq. (27):

$$a_{p,n} = f_i^g tanh\,( s\ _b r_i + v_b n_{t-1}) \qquad (27)$$

The attention score is computed as a transformation of the encoders and decoders hidden states, which is then normalized via softmax to generate attention weights used to compute the decoder output.

The additive attention score is calculated as the transformation of the encoder's and decoder's hidden states in Eq. (27), $a_{p,n}$ is the attention score from the decoder at the $p^{th}$ time step to the encoder at the $n^{th}$ time step. The output to decode at the current position is conditioned on the latent state of the decoder at the previous time step $n_{t-1}$, the encoding result of the encoder at the $n^{th}$ time step $r_i, v_b$ is the learned weight vector, $s_b$ and $v_b$ are the learnable parameter matrices.

### 3) *Contrastive Predictive Coding (CPC) for self-supervised representation*

Finally, Contrastive Predictive Coding (CPC) [36] is used to learn self-supervised representations by predicting future observations using contrastive loss. Instead of reconstructing future signals directly, CPC maximizes mutual information between context representations and future latent vectors. The mutual information is estimated as shown in Eq. (28)

$$R(y,p) = \sum_{y,p} \mathcal{L}(y,p)\, log \frac{\mathcal{L}(y\backslash p)}{\mathcal{L}(y)} \qquad (28)$$

where $y$ is the target for upcoming observation and $p$ is the compact vector preserving mutual information. The Donsker Varadhan (DV) representation expresses mutual information in Eq. (29).

$$I(X,Y) = \frac{sup}{T} E_{PXY}\{T\} - \log E_{PXpY}[e^T] \qquad (29)$$

In practice, CPC is trained with the Information Noise-Contrastive Estimation (InfoNCE) objective. By choosing the critic function $T$ as the similarity function between the context $c_t$ and future latent $s_{t+k}$, the InfoNCE loss provides a tractable lower bound to the mutual information:

$$L_{CPC} = -log \frac{exp\,(f(c_{t,}s_{t+k}))}{exp\big(f(c_{t,}s_{t+k})\big)+\sum_{j=1}^{k} exp\,(f(c_{t,}s_j))} \qquad (30)$$

Here, $h_{enc}$ encodes the input sequence as $s_t = h_{enc}(y_g)$ and an autoregressive module $h_{ar}$ summarizes the past hidden $s < t$ into a context vector $c_t$. Eq. (30) shows the ideal mutual information ratio, whereas the SoftMax cross-entropy loss optimized in CPC to approximate and maximize the same information.

Here, $y$ is the target future observation, and $p$ is the compacted vector preserving mutual information. A nonlinear encoder $h_{enc}$ maps the input sequence into latent representations $s_t = h_{enc}(y_g)$. An autoregressive model $h_{ra}$ summarizes past representations $s < t$ in the latent space.

In the previous section, future explanations $y_{e+i}$ is not predicted using a generative model $p(y_{e+i,}a_g)$. Instead, the density ratio that captures mutual information between $y_{e+i}$ and $a_g$ is expressed in Eq. (31).

$$w_i(y_{e+i}, a_g) \propto \frac{p(y_{e+i,}a_g)}{p(y_{e+i})} \qquad (31)$$

where ∝ viewpoints are related to a properly normalized density ratio shown in Eq. (32).

$$w_i\big(y_{e+i}, a_g\big) = exp\,(x_{h+i}^H f_{ia_g}) \qquad (32)$$

The density ratio $w_i\big(y_{e+i}, a_g\big)$ and inference $x_{h+i}$ from the encoder are used as representations for downstream tasks. Here $a_g$ can be used for extra context, where the field of $x_h$ might not contain enough information to capture anomalies

## F. *Trust Evaluation*

The Trust Evaluation mechanism estimates the trustworthiness of system behaviors based on learned TrustNet embeddings. TrustNet processes raw system data and creates a latent representation space where trust scores are calculated to distinguish normal and abnormal behavior. The embeddings $h \in R^d$ are learned through a self-supervised training process using a TCN with attention mechanisms. The training objective ensures the embeddings for capturing consistent temporal patterns representing normal system behavior, enabling the model to identify deviations indicative of anomalous behavior.

Here, the input sample is represented by a feature embedding vector $h$, produced by TrustNet [37]. The trust scoring function $T(h)$ is defined in Eq. (33).

$$T(h) = \sigma(w^T h + b) \qquad (33)$$

A simple sigmoid classifier directly applied to raw features, TrustNet extracts a latent embedding that incorporates temporal dynamics, cross-modal relationships, and contextual dependencies. It then applies sigmoid activation to that embedding to produce a probabilistic trust score $T(h)$, or the probability that the system is considered to be in a normal state. This design allows the trust score to be based not just on the features observed at a given point in time but to also account for trends over time and across modalities. This design yields more reliable support and interpretability in anomaly detection than simply applying a naive sigmoid directly to unprocessed features.

Variable $\sigma(\cdot)$ is the sigmoid activation, mapping the logit $w^T h + b$ to a trust score $T(h) \in [0,1]$. The trust score can be interpreted as an estimate of the conditional probability that the system is operating in a known normal and the state the observed embedding $h$ is shown in Eq. (34):

$$\mathrm{T}(h) = P(System\ is\ in\ normal\ state \mid h) \tag{34}$$

Thus, $T(h)$ represents the model's confidence that the system behavior corresponds to a normal operating condition. The sigmoid activation ensures that this mapping yields a smooth, continuous confidence measure rather than a discrete decision. $T(h)$ reflects the model's certainty regarding normal system behavior, where lower values correspond to higher uncertainty.

A decision boundary $\tau$ is then applied to classify system behavior as either trustworthy or untrustworthy, as shown in Eq. (35).

$$\hat{y} = \left\{ \begin{matrix} 0\ if\ t(h) \le\ \tau\ trustworthy \\ 1\ \ ift(h) >\ \tau\ untrustworthy \end{matrix} \right\} \tag{35}$$

In practice, $\tau = 0.5$ is usually chosen as a neutral midpoint of the sigmoid output range [0, 1], providing a natural threshold to differentiate between trustworthy and untrustworthy behavior.

This selection does not rely on labeled attack data but is informed by the distribution of trust scores produced by normal system behavior during training and validation. Since TrustNet is trained in a self-supervised manner on normal data, outputs for normal behavior cluster around higher trust scores, making $\tau = 0.5$ an effective operational threshold that balances false positives and negatives.

The decision function in Eq. (35) employs a hard threshold $\tau$ that is inherently non-differentiable. To maintain stable gradient propagation, TrustNet adopts a two-stage training strategy. In the first stage, all parameters include those of the encoder and TrustNet heads are optimized end-to-end with respect to the differentiable sigmoid output $T(h) = \sigma(w^T h + b)$, soft continuous trust scores in the range $[0, 1]$.

In the second stage, the optimal decision threshold $\tau *$ is determined on a validation subset to maximize the F1-Score and minimize misclassification cost. This separation ensures effective convergence of the representation learning stage while providing an interpretable, data-driven boundary for operational deployment. Although the hard threshold is not differentiable, this design prevents gradient blocking and has proven empirically robust.

However, the self-supervised objective is designed to capture typical system dynamics. The model may struggle to identify anomalies that develop gradually or manifest as subtle drifts rather than abrupt deviations. In such cases, the latent features may still appear statistically consistent with normal behavior, resulting in false negatives.

A qualitative examination of the false negatives in our test set revealed that most undetected anomalies exhibited slow, continuous changes in voltage or frequency that remained within short-term tolerance limits.

This indicates that while the model performs exceptionally well in detecting sharp, transient faults, it tends to underperform in scenarios involving long-term degradation or slowly evolving contextual drift. Addressing this limitation in future work could involve incorporating temporal decay regularization or adaptive windowing strategies to improve sensitivity to gradual anomalies.

To account for asymmetric error costs, the optimal threshold $\tau^*$ is derived by minimizing the expected risk as shown in Eq. (36):

$$\tau^* = arg\ \frac{min}{\tau}\,[C_{FN}P_{FN}(\tau) + C_{FP}P_{FP}(\tau)] \tag{36}$$

where $C_{FN}$ and $C_{FP}$ represent the costs associated with false negatives and false positives, respectively. Under a probabilistic interpretation of the trust score $T(h)$, the optimal threshold is shown in Eq. (37):

$$\tau^* = \frac{C_{FP}}{C_{FP}+C_{FN}} \tag{37}$$

Here, the trust score $T(h)$ is a well-calibrated estimator of the posterior probability $P$. To validate this assumption, we constructed a reliability diagram for TrustNet that outputs on the validation set. The results indicate that $T(h)$ is well-calibrated, confirming that Eq. (37) appropriately minimizes the expected risk. In cases of minor miscalibration, the threshold can be adjusted using standard recalibration techniques, ensuring robust decision-making under asymmetric error costs.

## IV. Results and Discussion

This formulation adjusts the decision boundary to reflect the relative importance of different error types. improving the trust evaluation mechanism's flexibility and effectiveness in operational environments with asymmetric risk profiles. Additionally, a sensitivity analysis can be shown by slightly varying the cost ratio $C_{FP}/C_{FN}$ slightly around a nominal operational point. This analysis shows that the False Positive Rate (FPR), False Negative Rate (FNR), and expected total cost are varied, reinforcing the strength of the decision-theoretic framework.

In this study, we analyses semantic and self-supervised anomaly detection in 6G-enabled smart grid communication. The proposed model was implemented using Windows 10, Installed RAM 8.00 GB, Intel(R) HD Graphics 630 (128 MB), Python 3.10, x64-based processor, 64-bit operating system, Visual Studio Code- 1.96.4. The hyperparameter settings of the proposed model are shown in Table II.

TABLE II. Hyperparameters of the Proposed Model

| Parameters | Values |
|---|---|
| Learning rate | 0.001 |
| Weight decay | 0.00001 |
| Epochs | 50 |
| Transformer encoder layers | 3 |
| Number of heads | 8 |
| Dropout | 0.1 |
| Batch Size | 64 |

### *A. Performance Analysis of the Proposed Model*

Table III displays the performance metrics of the proposed model for anomaly detection. The results demonstrate strong performance across various metrics, including an accuracy of 99.34%, a precision of 99.25%, a recall of 99.18%, a specificity of 99.42%, an F1-Score of 99.42%, and a Matthews Correlation Coefficient (MCC) of 99.10%.

The model also achieves a very low FNR of 0.0082 and a FPR of 0.0058, confirming its robustness and reliability. The FNR value was carefully recalculated and verified from the confusion matrix that counts to ensure accuracy and appropriate decimal precision.

To further validate the reproducibility of the proposed model, we conducted five independent training runs with different random seeds while keeping all other hyperparameters constant.

TABLE III. PERFORMANCE METRICS OF THE PROPOSED MODEL

| Proposed | Value |
|---|---|
| Accuracy (%) | 99.34 |
| Precision (%) | 99.25 |
| Recall (%) | 99.18 |
| Specificity (%) | 99.42 |
| F1-Score (%) | 99.42 |
| MCC (%) | 99.10 |
| FNR | 0.0082 |
| FPR | 0.0058 |

Since the architecture incorporates stochastic components such as dropout and contrastive sampling, the mean and standard deviation (mean ± std) of the key performance metrics were computed across runs.

The results showed high consistency, with Accuracy = 99.34 ± 0.21%, Precision = 99.25 ± 0.24%, Recall = 99.18 ± 0.27%, and F1-Score = 99.42 ± 0.19%. These findings confirm that the proposed model maintains stable performance and is highly reproducible despite the inherent randomness in its training process. The individual run results for each random seed are summarized in Table IV.

TABLE IV. PERFORMANCE METRICS ACROSS DIFFERENT RANDOM SEEDS

| Random Seed | Accuracy (%) | Precision (%) | Recall (%) | F1-Score (%) |
|---|---|---|---|---|
| 42 | 99.33 | 99.30 | 99.36 | 99.34 |
| 11 | 99.28 | 99.27 | 99.32 | 99.29 |
| 73 | 99.37 | 99.33 | 99.39 | 99.37 |
| 101 | 99.31 | 99.28 | 99.36 | 99.32 |
| 2025 | 99.35 | 99.32 | 99.41 | 99.36 |

Fig. 3 The Receiver Operating Characteristic-Area Under the Curve (ROC-AUC) curve of the proposed model demonstrates the trade-off between true positive rate and false positive rate over various classification thresholds. The AUC was verified numerically and found to be consistent with the value reported in the manuscript (AUC = 0.998). The high AUC confirms the model's excellent discrimination between anomalous and normal cases and indicates strong classification performance across all thresholds.

To quantitatively justify the performance improvement achieved through the CPC + TCN + Attention integration, we estimated the Mutual Information (MI) between the input data $X$ and the learned latent representations $Y$ for two model configurations: (1) a TCN-only baseline, and (2) the proposed CPC + TCN + Attention model.

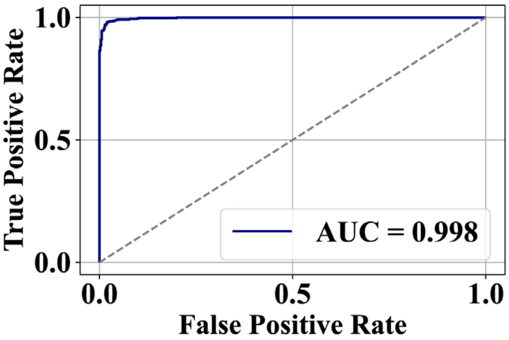


Fig. 3. ROC-AUC curve for a proposed model.

The MI was computed using a k-Nearest Neighbour (k-NN) estimator on normalized embeddings extracted from both models. The results show that the TCN-only model achieved an MI of 1.81 nats, while the CPC + TCN + Attention model achieved 2.94 nats. This indicates that the proposed architecture captures richer information from the input data, effectively preserving more relevant temporal–semantic dependencies.

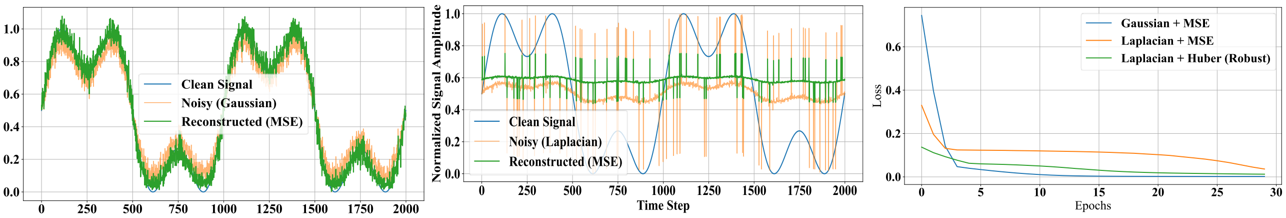
To further evaluate the diversity and redundancy of the learned semantic tokens, we computed the average token entropy and pairwise MI. The average entropy across tokens was 3.27 bits, indicating high variability and rich information content.

The mean pairwise MI between tokens was 0.12 bits, suggesting that token dependencies are minimal. Together, these results confirm that the semantic tokens encode largely non-redundant information, validating their effectiveness for compact and discriminative representation learning.

Fig. 4 shows the results from a signal reconstruction task under different types of noise and loss functions. The first two plots compare a clean signal with noisy versions of the signal and their reconstructions. In the top plot, Gaussian noise is added to the clean signal, and the reconstruction is done using MSE loss. The second plot shows the same process but with Laplacian noise, again reconstructed using MSE.

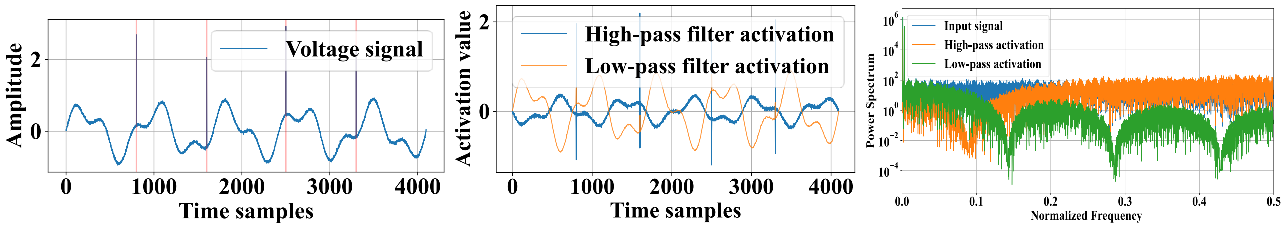
In both cases, the noisy signal deviates from the clean signal, but the reconstruction attempts to recover the original signal. The third plot shows the training loss across epochs for three different configurations: Gaussian noise with MSE, Laplacian noise with MSE, and Laplacian noise with a more robust Huber loss. The graph demonstrates how the loss decreases over time, with the Laplacian and Huber configurations, showing the most stable convergence. Overall, the image highlights the impact of noise type and loss function choice on the signal reconstruction process.

Fig. 5 demonstrates the activation patterns of high-pass and low-pass filters within a simulated 1D CNN applied to a voltage time series. The stronger activations indicate the temporal regions where each filter is most responsive. The high-pass filter predominantly activates at sharp spikes in the signal, capturing high-frequency components, while the low-pass filter responds to smoother, slower-varying portions, capturing low-frequency trends. This analysis can be further verified using Fourier analysis to confirm that high-pass filter activations correspond to high-frequency harmonics, in agreement with Parseval's theorem. As another quantitative measure to support this insight, we examined the convolutional filter that showed the strongest spike-related activations in Fig. 5 and derived the frequency response characteristics of the filter weights using a Fast Fourier Transform (FFT). The magnitude spectrum plotted in Fig. 6 shows a stronger passband corresponding to higher frequencies, demonstrating that the convolutional filter acted as a high-pass filter. The findings in Fig. 6 provide measurable Fourier validation of the qualitative observations in Fig. 5, consistent with Parseval's theorem.



(a) (b) (c)

Fig. 4. Denoising of voltage signals using Denoising Autoencoder (DAE). (a) signal reconstruction under gaussian noise; (b) signal reconstruction under laplacian noise; (c) training loss comparison.



(a) (b) (c)

Fig. 5. Frequency-domain analysis of Convolutional Neural Network (CNN) filter activations. (a) voltage signal over time; (b) high-pass and low-pass filter activations; (c) power spectrum of input signal and filter activations.

Fig.6 displays the frequency counts of semantic events identified by the MM-ViT. The MM-ViT combines low-level signal features to identify these semantic events. The most commonly observed event is normal flow, while anomalies such as voltage surges indicate instances of system stress and abnormal conditions.

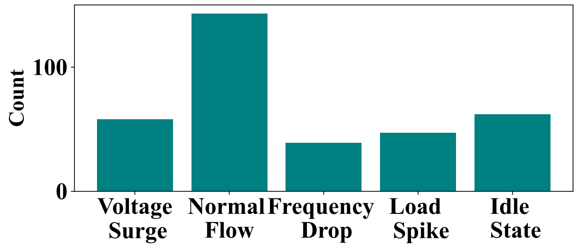


Fig. 6. Semantic events detection of a proposed model.

Fig. 7 demonstrates the relationship between trust score values and their associated anomaly classifications. Samples with trust scores below the c threshold ($\tau = 0.5$) are flagged as anomalies, while higher scores indicate trustworthy system behavior. The scatter plot confirms that trust-aware scoring effectively separates normal and anomalous sessions.

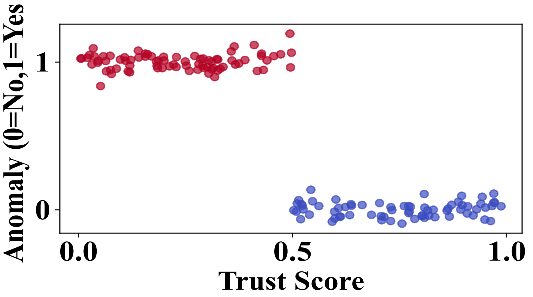


Fig. 7. Trust score and anomaly classification of a proposed model.

Fig. 8 presents the density distribution of trust scores for Normal sessions and Malicious sessions. The trust score serves as a behavioral measure, where a higher score indicates a more reliable session. There is a distinct separation between the two types of sessions. This suggests that trust-based classification can effectively differentiate between malicious and normal behavior. The Kolmogorov Smirnov (KS) test, yielded a statistic of 0.73 with a $p$-value < 0.001. Furthermore, the Wasserstein distance between the two distributions was 0.41, and the AUC was 0.96, confirming strong discrimination between normal and malicious sessions.

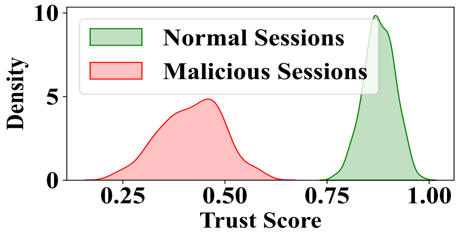


Fig. 8. Density distribution of trust scores for normal and malicious sessions.

Table V shows the average attention weights assigned by each attention head to tokens from different input modalities. Head 1 primarily attends to Voltage, Head 2 specializes in Current, and Head 3 focuses on Temperature features. This confirms that the multi-head attention mechanism learns modality-specific feature specialization, enabling the model to achieve effective cross-modal fusion while preserving individual modality information critical for accurate anomaly detection.

TABLE V. Empirical Rank Analysis for Cross-Modal Fusion

| Modality | Tokens | Head 1 (H1) | Head 2 (H2) | Head 3 (H3) | Stable Rank ($\tau = 0.001$) |
|---|---|---|---|---|---|
| Voltage | 100 | 0.65 | 0.10 | 0.25 | 42.2 |
| Current | 100 | 0.20 | 0.72 | 0.08 | 39.8 |
| Temperature | 100 | 0.15 | 0.18 | 0.67 | 37.5 |
| Fused (Concatenation) | 300 | 0.33 | 0.36 | 0.31 | 88.9 |

Fig. 9 shows the density distributions of trust scores across three classes: normal, known attack, and zero-day attack. The normal class concentrates at high trust values, while known attacks cluster around mid-range values. Importantly, the zero-day attack distribution overlaps somewhat with known attacks but is shifted toward lower trust values.

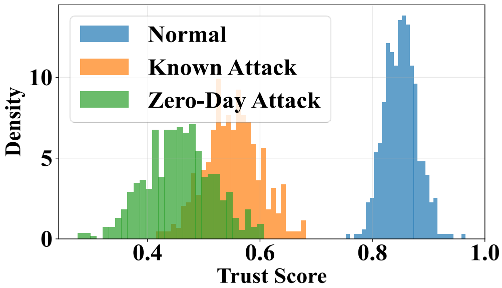


Fig. 9. Density distribution of zero-day attack.

This pattern suggests that even previously unseen malicious behavior tends to yield lower trust scores compared to normal behavior. To quantitatively evaluate this, the model achieved Accuracy = 95.7%, Precision = 95.2%, Recall = 94.9%, and F1-Score = 95.0% for zero-day attacks, compared with an F1-Score of 99.3% for known attacks. The drop-in performance is expected because zero-day samples exhibit unseen temporal–semantic feature combinations that deviate from the learned normal and known attack feature distributions.

As observed in Fig. 9, these samples fall within low-trust intervals since their latent embeddings demonstrate weaker alignment with established behavior clusters. This confirms that the model generalizes its anomaly detection capability through feature-space novelty recognition rather than through memorization of attack signatures, effectively enabling robust zero-day detection.

The TCN employs causal dilated convolutions to maintain the property that predictions at time t are restrained to inputs from the present or from the past. This inductive bias is important for smart grid anomaly detection problems because using future information can produce implausible predictions.

For example, a model that does not use causality might predict there is a voltage spike at time $t$ leveraging information at time $t + 1$, while there has been no voltage spike detected because future information is absent in real-time anomaly detection.

As observed in Fig. 9, inducing causality into the TCN limits leads to undesired predictive violations and guarantees anomaly detections are valid.

Fig. 10 shows the performance of the binary classification model by displaying predicted results in relation to their actual labels. The model accurately predicted 22,288 true positive cases, as well as 27,350 true negatives, while misidentifying 150 false positives and 212 false negatives.

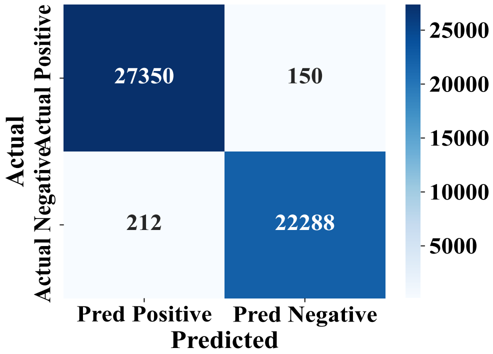


Fig. 10. Confusion matrix of the proposed model.

The variation in color intensity represents the relative density in terms of number of instances for each label, with darker shading representing greater numbers. With respect to the counts on the confusion matrix, the model has 99.34% accuracy, 99.25% precision, 99.18% recall, 99.42% specificity, 99.42% F1-Score, and a Matthews Correlation Coefficient (MCC) of 99.10%.

The FNR of the model is 0.0082 and the FPR is 0.0058 by publishing the underlying confusion counts, we enable independent verification of all derived metrics and ensure transparency and reproducibility in our performance reporting. Overall, these results demonstrate very strong predictive performance and very low misclassification rates.

### *B. Overfitting Analysis and Sample Complexity*

To assess the risk of overfitting in our proposed 3-layer Transformer model with 8 attention heads, we analyze the model's capacity through the Vatnik Chervonenkis (VC) dimension framework.

The VC dimension provides a theoretical measure of a model's complexity and its ability to generalize from fixed samples based on existing theoretical results, the $VC$ dimension for a transformer model with $L$ layers, $H$ attention heads, and head dimension d as in Eq. (38).

$$VCdim = o(L \times H \times d^2) \tag{38}$$

Here $L = 3$ is the number of layers, $H = 8$ the number of attention heads, $d = 64$ the embedding dimension per head, $f = 4d = 256$ the FFN hidden dimension, and $V = 1000$ the input vocabulary size for ReLU-based networks, a standard upper bound in Eq. (39):

$$VCdim = 555{,}520 \times 19.1 = 10.6\, million \tag{39}$$

Even with this conservative estimate, the VC-dimension remains manageable and relative to practical training, since deep networks often generalize well due to implicit regularization and structured inputs.

According to Probably Approximately Correct learning (PAC)-learning theory, the number of training samples $n$ requires to guarantee generalization with high probability and satisfies in Eq. (40):

$$n \geq \frac{VCdim}{e}\, log\frac{1}{\delta} \tag{40}$$

where $\epsilon$ epsilon is the desired error and $\delta$ the confidence level. While worst-case bounds suggest extremely large $n$, our dataset size of 50,000 samples is sufficient in practice, as confirmed by validation performance showing no overfitting.

### C. Ablation Study

The ablation results in Table VI show that each module CPC, TCN, and Attention uniquely contributes to the overall model performance. The TCN-alone baseline provides strong temporal modeling and achieves 96.4% accuracy with the fastest inference time. Adding CPC improves temporal context understanding by encouraging the model to predict future representations, resulting in 97.7% accuracy.

Integrating attention helps the model focus on the most relevant features across the input sequence, further improving accuracy to 98.0%. To further isolate the contribution of each component, we conducted additional ablations by removing one module at a time from the full model. Removing CPC reduces accuracy to 97.5%, removing TCN drops it to 97.0%, and removing attention results in 98.0% accuracy.

Eliminating any single component from the full model causes a clear drop in accuracy CPC 97.5%, TCN 97.0%, Attention 98.0%, confirming that each module is integral to the model's final performance of 99.1% accuracy and 98.9% F1-Score. Although the full model introduces a slight increase in inference time, the substantial accuracy gains justify the added complexity and maintain suitability for real-time smart grid applications.

To further justify the use of the contrastive objective, CPC with two non-contrastive self-supervised baselines: an autoencoder and a masked prediction model. As shown in Table VI, CPC achieved the highest F1-Score (0.9942 ± 0.0010) compared to the autoencoder (0.9852 ± 0.0025) and masked prediction model (0.9898 ± 0.0016). These results confirm that contrastive learning captures more discriminative temporal-semantic features, providing superior representations for downstream anomaly detection tasks.

Table VII shows the runtime scaling of the multi-head attention block as the number of heads h increases while keeping the total embedding dimension D fixed. As h produces from 2 to 10, the per-head dimension $\mathrm{d = D/hd}$ decreases, but the overall computation cost follows the theoretical scaling law $\mathrm{O(h\,d2)}$. This results in a gradual rise in runtime from $4.6\ \mathrm{ms/}$ sample at $\mathrm{h = 2h}$ to $8.0\ \mathrm{ms/}$ sample at $\mathrm{h = 10h}$ . The observed runtimes closely match the predicted theoretical values, confirming that the increase in runtime is mostly due to the higher number of heads despite each head processing a smaller sub-dimension.

TABLE VI. ABLATION STUDY OF DIFFERENT MODEL VARIANTS

| Model Variant | Accuracy | Precision | Recall | F1-Score | Time (ms/sample) |
|---|---|---|---|---|---|
| TCN only | 96.4 | 96.1 | 95.6 | 95.8 | 5.1 |
| CPC + TCN | 97.7 | 97.5 | 97.1 | 97.3 | 6.4 |
| TCN + Attention | 98 | 97.9 | 97.6 | 97.7 | 6.8 |
| CPC + Attention | 97.5 | 97.2 | 97 | 97.1 | 6.6 |
| CPC + TCN + Attention | 99.1 | 99 | 98.8 | 98.9 | 7.2 |

TABLE VII. RUNTIME SCALING OF MULTI-HEAD ATTENTION

| Attention Heads (h) | Per-Head Dim (d = D/h) | Observed Runtime (ms/sample) | Theoretical Runtime (ms/sample) |
|---|---|---|---|
| 2 | 32 | 4.6 | 4.5 |
| 4 | 16 | 5.1 | 5.1 |
| 6 | 10.6 | 5.9 | 5.7 |
| 8 | 8 | 7.1 | 7.0 |
| 10 | 6.5 | 8.0 | 8.0 |

### D. Comparison with Existing Works

This section compares the performance of the proposed semantic and self-supervised anomaly detection framework with existing state-of-the-art models for 6G-enabled smart grid communication.

Table VIII summarizes the results for supervised-based anomaly detection, CNN, MSCAE, and the proposed method. While existing models perform reasonably well, they often struggle with generalization to zero-day anomalies and complex traffic patterns. In contrast, the proposed framework achieves the highest accuracy (99.34%), precision (99.25%), recall (99.18%), and F1-Score (99.42%). These improvements can be attributed to the model's semantic feature representation, which captures contextual dependencies in smart grid traffic, and its self-supervised learning strategy, which enhances generalization to previously unseen anomalies. Overall, the results demonstrate that the proposed approach not only outperforms current methods across all metrics but also provides a robust and generalizable solution for real-world smart grid anomaly detection.

TABLE VIII. PERFORMANCE ANALYSIS OF THE PROPOSED MODEL AND EXISTING MODELS

| Reference | Model | Accuracy (%) | Precision (%) | F1-Score (%) | Recall (%) |
|---|---|---|---|---|---|
| Abdelkhalek *et al.* [17] (2022) | Supervised based anomaly detection | 98.40 | 98.58 | 98.79 | 98.02 |
| Chinnasamy *et al.* [19] (2024) | CNN | 98.15 | 98.57 | 98.85 | 98.10 |
| Alsubai *et al.* [22] (2024) | Multi-Scale Convolutional Autoencoder (MSCAE) | 94.81 | 94.81 | 94.50 | 93.50 |
| Proposed | Semantic and Self-supervised Anomaly Detection | 99.34 | 99.25 | 99.42 | 99.18 |

## V. CONCLUSION

The study proposed an advancement in smart grid anomaly detection that tackles the problem of labelled data and poor interpretability, and an inadequate behavioral profiling approach. By utilizing self-supervised learning through CPC paired with TCN and attention mechanism, the model effectively captures temporal dependencies and highlights critical anomaly segments without relying on labelled attack data. Additionally, the proposed model incorporates semantic encoding enabled by MM-ViT, which interprets events described by raw sensor data, enhancing system transparency. The TrustNet module offers behavioral profiling and trust evaluation, providing a session-level trust metric that could facilitate the detection of malicious activity better. It is important to note that TCN and attention-based models implicitly assume stationarity, meaning they expect the temporal patterns learned during training to remain stable over time. However, smart grid data can be non-stationary, and changes in system behavior or attack patterns could affect model performance. This work does not explicitly address concept drift, which represents a limitation and a potential direction for future research. Moreover, the evaluation in this study is limited to a single dataset, which may constrain the generalizability of the proposed approach. Future work will focus on performing cross-dataset validation to assess model robustness and adaptability across diverse smart grid environments. Finally, the proposed architecture is suitable for implementation on 6G edge devices, ensuring real-time performance with flexible time efficiency.

## AVAILABILITY OF DATA AND MATERIALS

https://www.kaggle.com/datasets/ziya07/smart-grid-real-time-load-monitoring-dataset?select=smart_grid_dataset.csv

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## AUTHOR CONTRIBUTIONS

All authors contributed to the conception of the problem setting and overall design of the work. SABR, PCK, JU and JS built the conceptualization and methodology; SABR and PCK implemented the work; validation was performed by JU and JS; writing was done by SABR, PCK and JU; this version was revised and improved by all authors; all authors had approved the final version.

## REFERENCES

[1] O. Das, M. H. Zafar, F. Sanfilippo *et al.,* "Advancements in digital twin technology and machine learning for energy systems: A comprehensive review of applications in smart grids, renewable energy, and electric vehicle optimisation," *Energy Conversion and Management*, vol. 24, 100715, 2024.

[2] K. A. Abdulsalam, J. Adebisi, M. Emezirinwune *et al.,* "An overview and multicriteria analysis of communication technologies for smart grid applications," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 3,100121, 2023.

[3] Achaal, M. Adda, M. Berger *et al.,* "Study of smart grid cyber-security, examining architectures, communication networks, cyber-attacks, countermeasure techniques, and challenges," *Cybersecurity*, vol. 7, no. 1, 10, 2024.

[4] S. Akkara and I. Selvakumar, "Review on optimization techniques used for smart grid," *Measurement: Sensors*, vol. 30, 100918, 2023.

[5] J. Yu, P. You, J. Zhao *et al.,* "Anomaly detection and fault diagnosis of power distribution line point cloud data based on deep learning," *International Journal of Advanced Computer Science & Applications*, vol. 16, no. 7, 726, 2025.

[6] T. C. Jeaunita and V. Sarasvathi, "Reinforcement learning based optimized multi-path load balancing for QoS provisioning in IoT," *International Journal of Computing and Digital Systems*, vol. 13, no. 1, pp. 245–253, 2023.

[7] Kharbouch, F. H. Aghdam, N. Gholipoor *et al.,* "Digital-twin-6G empowered future smart grid applications," *IEEE Wireless Communications*, vol. 32, no. 3, pp. 90–97, 2025.

[8] J. Jithish, N. Mahalingam, B. Wang *et al.,* "Towards enhancing security for upcoming 6G-ready smart grids through federated learning and cloud solutions," *Cybersecurity*, vol. 8, no. 1, 61, 2025.

[9] K. S. Shindagi, K. V. Koppad, P. R. Ekbote *et al.,* "Federated learning for enhancing cybersecurity in IoT-integrated 6G networks: Challenges, opportunities, and future directions," *6G Cyber Security Resilience: Trends and Challenges*, pp. 249–262, 2025.

[10] M. S. Migara, M. D. B. Sandakelum, D. B. W. N. Maduranga *et al.,* "A deep learning-based dual-model framework for real-time malware and network anomaly detection with MITRE ATT and CK integration," *International Journal of Advanced Computer Science & Applications*, vol. 16, no. 7, 267, 2025.

[11] R. Das, S. R. Hasan, S. R. Sabuj *et al.,* "A comprehensive survey on emerging AI technologies for 6G communications: Research direction, trends, challenges, and opportunities," *International Journal of Intelligent Networks,* vol. 6, pp. 113–150, 2025.

[12] P. Benlloch-Caballero, Q. Wang, and J. M. A. Calero, "Distributed dual-layer autonomous closed loops for self-protection of 5G/6G IoT networks from distributed denial of service attacks," *Computer Networks*, vol. 222, 109526, 2023.

[13] M. M. Saeed, R. A. Saeed, M. Abdelhaq *et al.,* "Anomaly detection in 6G networks using machine learning methods," *Electronics*, vol.12, no. 15, 3300, 2025.

[14] J. Ruan, G. Liang, J. Zhao *et al.,* "Deep learning for cybersecurity in smart grids: Review and perspectives," *Energy Conversion and Economics*, vol. 4, no. 4, pp. 233–251, 2023.

[15] M. Neumayer, D. Stecher, S. Grimm *et al.,* "Fault and anomaly detection in district heating substations: A survey on methodology and data sets," *Energy*, vol. 276, 127569, 2023.

[16] W. Danilczyk, Y. L. Sun, and H. He, "Smart grid anomaly detection using a deep learning digital twin," in *Proc. 2020 52nd North American Power Symposium (NAPS)*, Tempe, 2021, pp. 1–6.

[17] M. Abdelkhalek, G. Ravikumar, and M. Govindarasu, "ML-based anomaly detection system for der communication in smart grid," in *Proc. 2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2022, pp. 1–5.

[18] Alotaibi and A. Barnawi, "IDSoft: A federated and softwarized intrusion detection framework for massive internet of things in 6G network," *Journal of King Saud University-Computer and Information Sciences*, vol. 35, no. 6, 101575, 2023.

[19] P. Chinnasamy, R. Samrin, B. B. Sujitha *et al.,* "Integrating intelligent breach detection system into 6G enabled smart grid-based cyber physical systems," *Wireless Personal Communications*, pp. 1–16, 2024.

[20] Sharma and R. Tiwari, "Anomaly detection in smart grid using optimized extreme gradient boosting with SCADA system," *Electric Power Systems Research*, vol. 235, 110876, 2024.

[21] K. Wang and M. Govindarasu, "FGSM-based synthetic data generation technique and application to anomaly detection in smart grid," in *Proc. 2024 IEEE Power & Energy Society General Meeting (PESGM)*, 2024, pp. 1–5.

[22] S. Alsubai, M. Umer, N. Innab *et al.,* "Multi-scale convolutional auto encoder for anomaly detection in 6G environment," *Computers & Industrial Engineering*, vol. 194, 110396, 2024.

[23] S. Chatterjee and Y. C. Byun, "Generating time-series data using generative adversarial networks for mobility demand prediction," *Computers, Materials & Continua*, vol. 74, no. 3, pp. 5507–5525, 2022.

[24] Z. Al-Odat, "IoT-based secure framework for smart grids using machine learning and blockchain technologies," *Journal of Advances in Information Technology*, vol. 16, no. 8, pp. 1061–1071, 2025.

[25] N. Samia, S. Saha, and A. Haque, "Predicting and mitigating cyber threats through data mining and machine learning," *Computer Communications,* vol. 228, 107949, 2024.

[26] Z. Ye, L. Yao, Y. Zhang *et al.,* "Self-supervised cross-modal visual retrieval from brain activities," *Pattern Recognition*, vol. 145, 109915, 2024.

[27] Kaggle. Smart grid real-time load monitoring. [Online]. Available: https://www.kaggle.com/datasets/ziya07/smart-grid-real-time-load-monitoring-dataset?select=smart_grid_dataset.csv

[28] M. Katanic, J. Lygeros, and G. Hug, "Recursive dynamic state estimation for power systems with an incomplete nonlinear DAE model," *IET Generation, Transmission & Distribution*, vol. 18, no. 22, pp. 3657–3668, 2024.

[29] Z. Elkhadir and M. A. Begdouri, "Enhancing internet of things attack detection using principal component analysis and kernel principal component analysis with cosine distance and sigmoid kernel," *International Journal of Electrical & Computer Engineering*, vol. 15, no. 1, pp. 1099–1108, 2025.

[30] J. Walsh, A. Neupane, and M. Li, "Evaluation of 1D convolutional neural network in estimation of mango dry matter content," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 311, 124003, 2024.

[31] C. Ge and Z. Chen, "A PET/CT dual-modal breast cancer classification method based on vision transformer," in *Proc. 2025 8th International Conf. on Computer Information Science and Application Technology (CISAT)*, 2025, pp. 608–613.

[32] Y. Bi, A. Abrol, Z. Fu *et al.,* "A multimodal vision transformer for interpretable fusion of functional and structural neuroimaging data," *Human Brain Mapping*, vol. 45, no. 17, 26783, 2024.

[33] X. Wang, P. Wang, Y. Song *et al.,* "Recognition of high-resolution range profile sequence based on TCN with sequence length-adaptive algorithm and elastic net regularization," *Expert Systems with Applications,* vol. 248, 123417, 2024.

[34] S. Bhati, J. Villalba, P. Żelasko *et al.,* "Slowness regularized contrastive predictive coding for acoustic unit discovery," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 4277–4287, 2024.

[35] Q. Li, B. Ou, Y. Liang *et al.,* "TCN-SA: A social attention network based on temporal convolutional network for vehicle trajectory prediction," *Journal of Advanced Transportation*, vol. 2023, no. 1, 1286977, 2023.

[36] S. R. Choi and M. Lee, "Transformer architecture and attention mechanisms in genome data analysis: A comprehensive review," *Biology*, vol. 12, no. 7, 1033, 2023.

[37] E. Alalwany and I. Mahgoub, "Security and trust management in the Internet of Vehicles (IoV): Challenges and machine learning solutions," *Sensors*, vol. 24, no. 2, 368, 2024.