Building and Environment 258 (2024) 111568

Contents lists available at ScienceDirect

Building and Environment

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

Transformer encoder based self-supervised learning for HVAC fault detection with unlabeled data

M.A.F. Abdollah *, R. Scoccia, M. Aprile

Department of Energy, Politecnico di Milano, Via R. Lambruschini 4a, 20156, Milan, Italy

A B S T R A C T

Data driven methods are the most studied fault detection and diagnostics (FDD) type in buildings HVAC systems. However, most studies rely on labeled data for specific faults which are hard to find and collect for real systems. While the fault-free data is easier to collect, it is still time consuming to label for large systems operation. Moreover, most of the studies rely on the usage of supervised learning algorithms which do not generalize well beyond the training data making unseen faults hard to detect. In this paper, we define a methodology to use a self-supervised learning method for HVAC systems' FDD using a Transformer encoder, moreover, we tested it on a real case study. By strategically masking portions of the multivariate time-series data using Markov chain approach with two states. The model is trained by predicting these concealed segments. This approach, independent of labeled data, offers a scalable solution for practical HVAC applications. Anomalies are labeled using the Peak Over Threshold (POT) method, which dynamically determines thresholds by fitting reconstruction errors to a generalized Pareto distribution. Subsequent fault diagnostics emphasize features with pronounced reconstruction errors, pinpointing potential HVAC malfunctions. This methodology reduces dependence on labeled datasets and augments the model's generalization, facilitating detection of unobserved faults. This approach was applied to data from a real building. As a results multiple faults were detected mainly due to the malfunctioning of the monitoring system. The model demonstrates the ability to detect both sequential and individual faults. The period from October 19th to December 23rd was detected as a fault period due to the change in the trend of the data because of the monitoring system.

1. Introduction

HVAC systems in buildings are often prone to defects which can result in suboptimal outcomes, such as increased energy consumption, elevated maintenance expenses, compromised comfort in terms of thermal conditions, and deteriorating air quality. These defects can arise from malfunctioning sensors, equipment breakdowns, or incorrect operations of the system. Research indicates that building system inefficiencies and inadequate control measures can lead to energy losses ranging from 15 % to 30 % [1]. Consequently, the implementation of FDD, or AFDD as it is sometimes called, is essential for the assurance of dependable system functioning and the conservation of energy. Fault detection is primarily concerned with recognizing any improper or unsatisfactory building operations, while fault diagnostics involves pinpointing the exact reasons for these operational failures [2]. In the U.S., within office spaces and institutions of higher learning, the application of FDD has been linked to median energy savings of about 10 % per year, along with a simple payback period of two years [3]. This highlights the FDD systems' viability and appeal as an investment in the infrastructure domain.

Numerous investigations have demonstrated the effectiveness of supervised machine learning in identifying and diagnosing faults in heating, ventilation, and air conditioning (HVAC) systems [4,5]. These studies leverage supervised learning algorithms to decipher the intricate links between various monitoring parameters (like temperature, pressure, and flow rates) and the operational conditions (such as normal or faulty operations) [6,7]. The resulting data-driven models vary in complexity, encompassing everything from simple linear to complex nonlinear equations, individual to collective models, and basic to advanced architectural designs [8,9]. Significant progress has been noted, particularly in accurately identifying issues in critical HVAC components, including chillers [10,11] and air handling units (AHUs) [12]. A fundamental assumption in supervised learning is the availability of labeled data for trustworthy predictive modeling. However, labeling data to accurately reflect the real operational status of systems can be an exhaustive and labor-intensive process. Consequently, most operational data from buildings remain unlabeled, and only a few buildings can afford the application of sophisticated supervised learning methods for accurate fault classification in HVAC systems.

Recent research in the building sector has concentrated on two learning approaches to address the scarcity of labeled data: transfer learning and semi-supervised learning [13,14]. Solutions based on

* Corresponding author.
E-mail address: mohammadabdollah.abdollah@polimi.it (M.A.F. Abdollah).

https://doi.org/10.1016/j.buildenv.2024.111568
Received 26 January 2024; Received in revised form 1 April 2024; Accepted 23 April 2024
Available online 24 April 2024

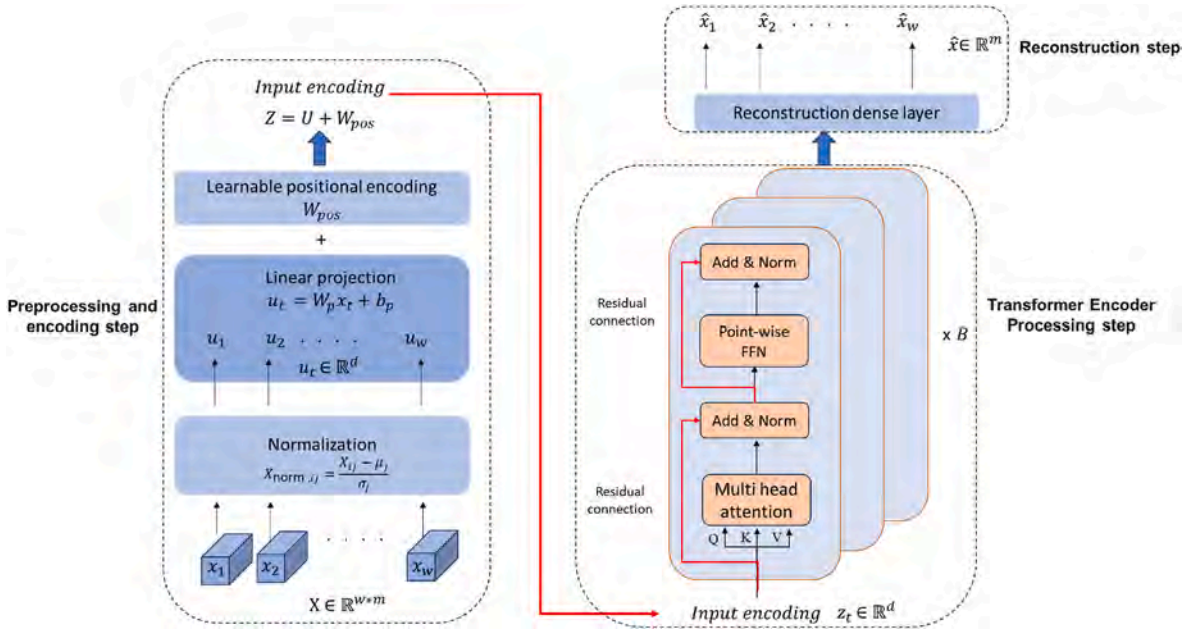

Fig. 1. The main Architecture of the algorithm on the left figure (starting from the bottom) the steps of preprocessing and encoding of the data to the model dimension (d) are displayed, and on the right the modeling and reconstruction of the data to the original dimension (m).

Table 1
Model architecture breakdown.

Layer name	Layer type	Description	Number of layers
Input normalization	Preprocessing	Standardizes the input data to zero mean and unit variance.	1 per feature
Linear projection	Transformation	Projects normalized features into a d-dimensional model space.	1
Positional encoding	Encoding	Adds learnable temporal context to input sequences.	1
Multi head attention	Self-Attention	Processes sequences in parallel, focusing on different parts of the sequence simultaneously.	2 (7 attention heads each)
Feedforward network	Transformation	Applies point-wise transformations to the output of the attention layer.	2
Add & Norm	Residual connection	Combines the outputs of the attention and feedforward networks with layer normalization.	4 (2 per encoder layer)
Output projection	Reconstruction	Maps the encoded sequence back to the original feature space for reconstruction.	1

transfer learning propose using insights gleaned from data-rich buildings to tailor models for buildings with less data [15,16]. This approach offers a promising way to exploit operational data from various building systems and conditions. However, it presumes the availability of data from buildings with similar characteristics, which may not always be the case. In focusing on the data from individual buildings, other studies have assessed the merits of semi-supervised learning in using unlabeled operational data [17]. Yan et al. examined the efficacy of various semi-supervised algorithms in categorizing faults in AHUs [18], finding that this approach can significantly enhance model performance even with limited labeled data. Fan et al. introduced a unique semi-supervised framework using artificial neural networks for diagnosing faults in AHUs, employing a base model trained on limited labeled data and iteratively updating it with high-quality pseudo labels derived from unlabeled data [19]. Li et al. applied semi-supervised generative adversarial networks to better understand the distribution of unlabeled data, thereby improving fault diagnosis in chillers [20,21]. Their approach involved training a discriminator model to classify real data labels while distinguishing between real and artificial data samples, thus facilitating the creation of a reliable fault classification model with minimal labeled data. A notable limitation of semi-supervised learning is its partial dependence on initial labeled data. For example, in the widely used self-training method, the quality of pseudo labels generated from unlabeled data can be substandard if the initial model is developed with an extreme scarcity of data, potentially leading to decreased performance in predictive modeling. One of the main reasons behind the lack of adoption of data driven FDD in the building sector is due to the fact that most proposed methods depend entirely or partially on labeled data which is inherently difficult to systematically obtain for several reasons:

1. Expertise Requirement: Accurately labeling faults requires a deep understanding of building systems and operations, which necessitates the involvement of domain experts. This can significantly increase the time and cost associated with the data labeling process.
2. Variability and Complexity: Buildings vary greatly in their design, usage, and maintenance, leading to a wide range of potential faults that are often complex and interrelated. This variability makes it challenging to create a comprehensive labeling schema that accurately represents all potential faults.
3. Dynamic Environments: The operational conditions of buildings and their systems can change over time, affecting fault manifestations. This dynamic nature requires continuous updates to labeled data to remain relevant, adding to the complexity and cost of the labeling process.

Self-supervised learning emerges as a promising solution, offering a potential means to reduce the reliance on labeled data in predictive modeling [22] Self-Supervised Learning (SSL) represents a segment of unsupervised learning that leverages internally generated tasks, known as pretext tasks, to extract supervisory cues from data without labels. These internally devised challenges enable the model to extract knowledge from the dataset, which in turn fosters the creation of meaningful representations for subsequent analytical tasks. SSL circumvents the need for externally labeled data since the supervisory signals are intrinsically obtained from the data. Owing to the strategic design of these pretext tasks, SSL has marked notable advancements in

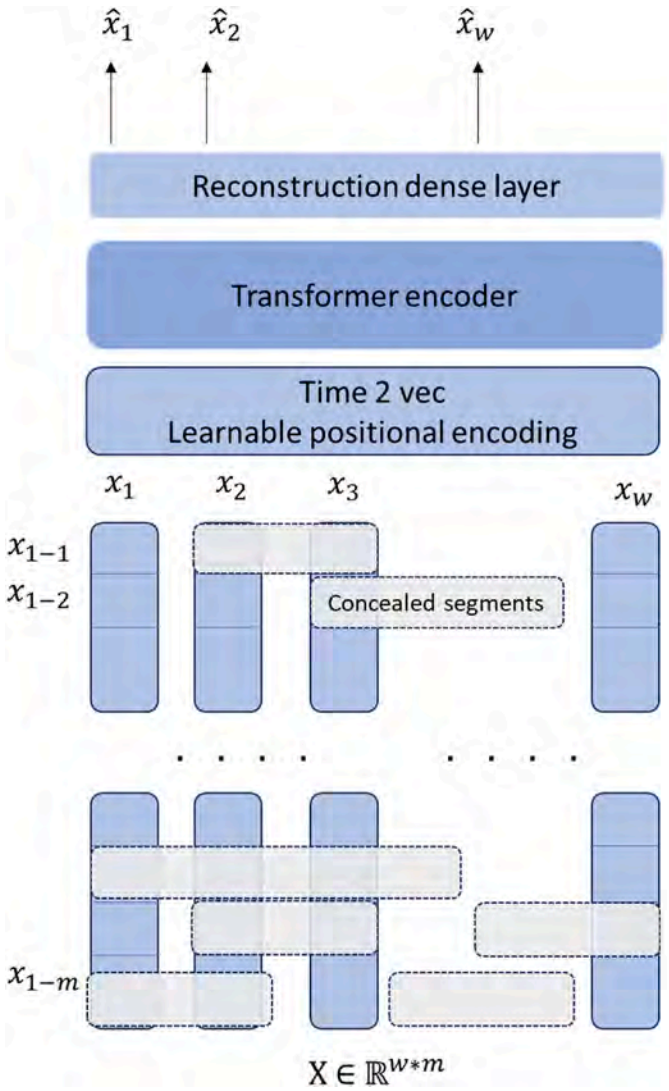

Fig. 2. Pre-training step overview.

the realms of Computer Vision (CV) and Natural Language Processing (NLP).

In this study, we used and trained an encoder-only transformer-based architecture in a generative, self-supervised manner. This method was tested against unlabeled data from a real building equipped with a heat pump (HP) that is connected to air handling unit (AHU) for ventilation and floor heating for heating. The main contribution of this study can be summarized as follows:

1. The method described does not rely on labeled data of specific faults which is very hard to collect and generate, especially in the building sector. The method can also be fine-tuned with labeled data if available.
2. Fault-free data is relatively easier to obtain but time consuming to label manually, the method described speeds up the process of labeling fault-free data.
3. The method can facilitate the diagnostics of fault by pointing to the features that have the highest reconstruction error.
4. Less pronounced faults can be masked by obvious faults in the mean squared reconstruction error curve, therefore dynamic thresholding technique was implemented to uncover those errors.

2. Theoretical background

The goal of this section is to give an overview of the original transformer architecture as it was introduced for Natural language processing (NLP) purposes and all its components (positional encoding, multi head attention, feed forward and residual network). Then how this architecture was adapted to be used for time series data.

The innovation of Transformer in deep learning [23] has brought great interest recently due to its excellent performances in NLP [24] computer vision (CV) [25], and speech processing [26]. Over the past few years, numerous Transformer variants have been proposed to advance the state-of-the-art performances of various tasks significantly. There are quite a few literature reviews from different aspects, such as in NLP applications [27], CV applications [28], and efficient Transformers [29].

2.1. Vanilla transformer

The classic Transformer introduced by Ref. [23] is essentially built on an encoder-decoder framework. This structure comprises multiple identical layers in both the encoder and decoder. Each layer is characterized by two main components: a multi-head attention mechanism and a position-specific feed-forward network. The decoder further integrates a cross-attention mechanism that works in tandem with the multi-head self-attention and the position-wise feed-forward module.

2.2. Encoding the input and position

In contrast to models like LSTM and RNN, the basic Transformer doesn't use a recurrent mechanism. Instead, it adds positional encoding to the input embeddings to capture sequential information. We briefly explain some prominent positional encoding methods:

2.2.1. Absolute positional encoding

In the standard Transformer, each sequence position, denoted as

$$PE(t)_i = \begin{cases} \sin(\omega_i t) & i\%2 = 0 \\ \cos(\omega_i t) & i\%2 = 1 \end{cases} \quad (1)$$

$\omega_i t$ represents a predefined frequency for each dimension. An alternative approach is to learn these positional embeddings, which offers more adaptability, as suggested by Ref. [30].

2.2.2. Relative positional encoding

The idea behind relative positional encoding is that the relationships between sequence positions can be more informative than their absolute positions. Some techniques have been devised to add relative positional encodings directly to the attention mechanism's keys. Shaw and team in 2018 provided insights into this. Additionally, there are hybrid methods that merge both absolute and relative positional encodings, where the positional information gets combined with the token embeddings directly.

2.3. Multi-head attention

With Query-Key-Value (QKV) model, the scaled dot-product attention used by Transformer is given by:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{D_k}}\right)\mathbf{V} \quad (2)$$

where queries $\mathbf{Q} \in \mathscr{R}^{N\times D_k}$, keys $\mathbf{K} \in \mathscr{R}^{M\times D_k}$, values $\mathbf{V} \in \mathscr{R}^{M\times D_v}$, N, M denote the lengths of queries and keys (or values), and D_k, D_v denote the dimensions of keys (or queries) and values. Transformer uses multi-head attention with H different sets of learned projections instead of a single attention function as:

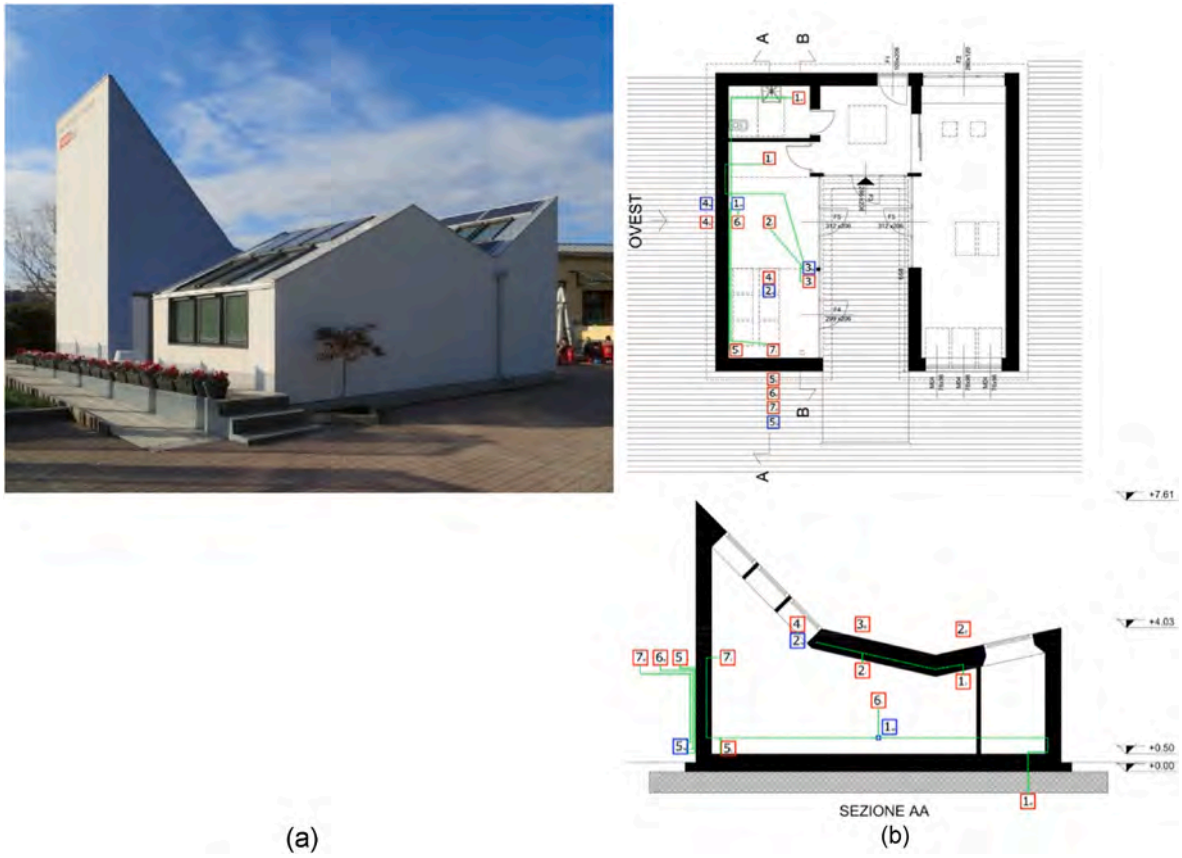

Fig. 3. a): A picture of the building after renovation and localization at Politecnico di Milano, Bovisa Campus, b): Plan and a section of the building.

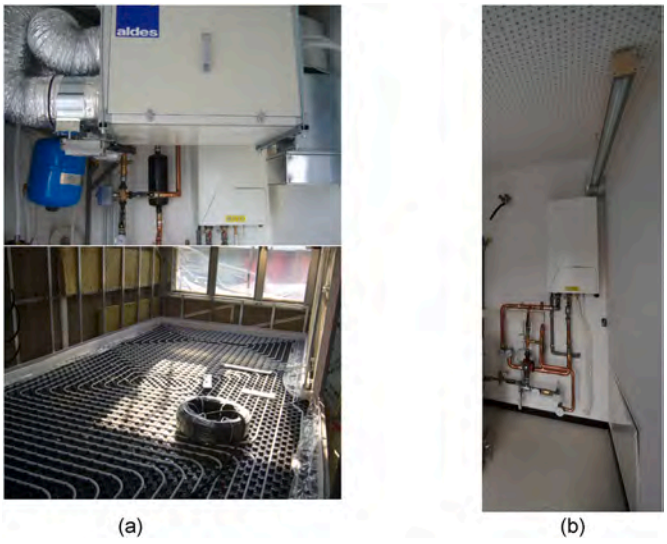

Fig. 4. Actual system implementation. a): Top: Air handling unit, bottom: radiant floor, b): Indoor unit of the heat pump.

$$\text{MultiHeadAttn}\,(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}\,(\,\text{head}_{\,1},\cdots,\,\text{head}_{\,H})\mathbf{W}^{O} \tag{3}$$

where $\text{head}_i = \text{Attention}\,(\mathbf{Q}\mathbf{W}_i^{Q},\mathbf{K}\mathbf{W}_i^{K},\mathbf{V}\mathbf{W}_i^{V})$. The Attention () function computes the relevance of different values based on the queries and keys. For each query, Attention () assigns weights to the keys based on their similarity, and these weights are used to aggregate the corresponding values into a single output. This allows the model to focus on the most relevant parts of the input when making predictions.

2.4. Feed-forward and residual network

The feed forward network is a fully connected forward propagation module defined by the following expression:

$$FFN(\mathbf{H}^{'}) = \text{ReLU}\left(\mathbf{H}^{'}\mathbf{W}^{1} + \mathbf{b}^{1}\right)\mathbf{W}^{2} + \mathbf{b}^{2}, \tag{4}$$

Where:

$\mathbf{H}^{'}$: The output of the previous layer

$\mathbf{W}^{1} \in \mathscr{R}^{D_m \times D_f}$, $\mathbf{W}^{2} \in \mathscr{R}^{D_f \times D_m}$, $\mathbf{b}^{1} \in \mathscr{R}^{D_f}$, $\mathbf{b}^{2} \in \mathscr{R}^{D_m}$.

In this formula $\mathbf{H}^{'}$ represents the output from the preceding layer. $\mathbf{W}^{1}$ is a matrix of dimensions appropriate for mapping the input features to an intermediary dimension, while $\mathbf{W}^{2}$ serves to map these intermediary features to the desired output dimension. Similarly, $\mathbf{b}^{1}$ and $\mathbf{b}^{2}$ are bias vectors corresponding to each weight matrix and are subject to optimization during training.

As the network depth increases, it becomes beneficial to incorporate a residual connection, along with layer normalization, to enhance the flow of gradients during training. Thus, the module can be extended as follows:

$$\begin{aligned} \overline{H} &= \text{LayerNorm}(\text{SelfAttn}\,(X) + X) \\ H &= \text{LayerNorm}(FCN(\overline{H}) + \overline{H}) \end{aligned} \tag{5}$$

Here SelfAttn( ) signifies the self-attention mechanism that processes the input X. and LayerNorm( ) denotes the process of layer normalization.

2.5. Transformers for time series and anomaly detection

In recent advancements, the Transformer architecture, originally designed for natural language processing, has been extensively modified to cater to the intricacies of time series data [31,32]. One pivotal

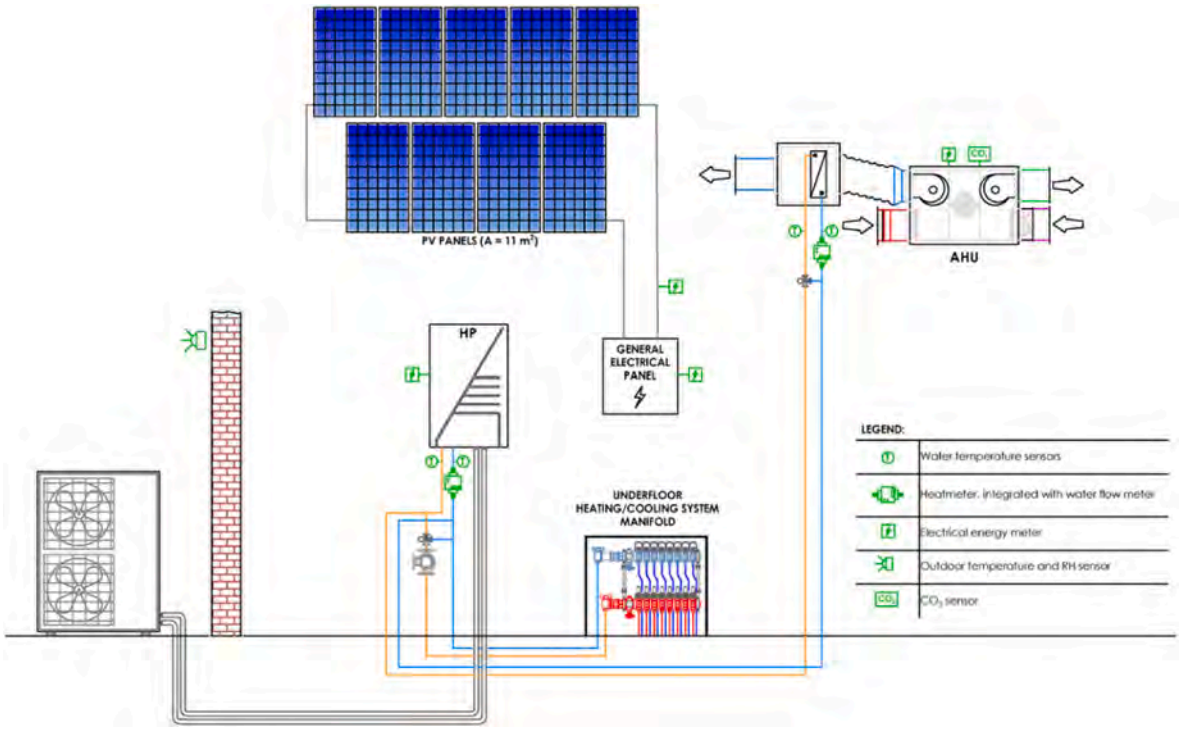

Fig. 5. Schematic of the HVAC system and the positions of the sensors of the monitoring system.

adaptation is the introduction of adaptive positional encoding techniques, moving beyond the vanilla model's basic positional encoding. Research indicates dynamic embeddings derived directly from time series data, such as those introduced by learning layers within the Transformer [33] or generated through LSTM networks [34], significantly enhance model efficacy by providing tailored flexibility and capturing the sequential order inherent in time series.

Moreover, leveraging timestamps as an additional form of positional encoding, as seen in models like Informer [35], Autoformer [36], and FEDformer [37], brings forth the untapped potential of time-specific data points. This approach underscores the value of incorporating both regular intervals and significant dates to enrich the model's temporal understanding.

Addressing the computational challenges of the self-attention mechanism, proposals like LogTrans [38] and Pyraformer [39] have introduced efficient strategies through inducing sparsity and exploiting the self-attention matrix's low-rank characteristics, respectively. Architectural innovations further include hierarchical structuring, as implemented by Informer [35] and Pyraformer [39], to process time series at varying scales, enhancing both model efficiency and data interpretation capabilities.

Transitioning to anomaly detection, the transformative application of the Transformer architecture [40] and its integration with generative neural models such as VAEs [41–44] and GANs [45] have marked significant improvements in detecting time series anomalies. Adversarial training methods [41], multi-scale approaches [42], and graph-based learning frameworks [46] exemplify the broadening scope of Transformers in capturing complex temporal relationships and multivariate series characteristics. These adaptations underscore the architecture's versatility in enhancing anomaly detection accuracy and addressing the limitations of traditional methods.

In sum, these modifications, and applications of the Transformer architecture to time series analysis and anomaly detection highlight the ongoing innovation in adapting deep learning models to the unique demands of time series data, significantly improving their performance and applicability across various tasks.

3. Methodology

In this section we introduce the data preprocessing procedure, the model used in the study, the self-supervised training method used for the pretraining step and finally the dynamic thresholding technique used to flag the anomalies.

3.1. Core architecture

Central to our approach is an encoding mechanism inspired by the transformer architecture delineated by Ref. [23]. Our model diverges from this foundational design in that it eschews the decoder module, opting instead for an encoder-only framework The primary reason for employing only an encoder in this research, focusing on multivariate anomaly detection in time series, is due to the non-generative nature of the task. Unlike the original Transformer architecture, which was designed for language translation - a generative task requiring an encoder to understand one language and a decoder to generate another - anomaly detection in time series data involves identifying deviations from normal patterns within the same data context. Therefore, a decoder is unnecessary; the encoder alone is sufficient to model and identify these anomalies effectively. This approach streamlines the model and makes it more computationally efficient, focusing its learning capabilities on recognizing irregularities in the time series data. The computational efficiency improvement stems from the fact that in traditional encoder-decoder architectures, both the encoder and decoder independently contribute to computational complexity due to the self-attention mechanism's pairwise comparison of tokens, resulting in a quadratic relationship with the input sequence length. By adopting an encoder-only model, we remove the need for the decoder and its associated complexity entirely. In the context of anomaly detection, where the decoder's generative function is not required, our approach effectively halves the self-attention computation. Therefore, for a time series of length (n), while an encoder-decoder model would require $O(2.l.n^2)$ operations for l operations due to the combined processing in both the encoder and decoder, our encoder-only model require only $O(l.n^2)$ operations. This is a conservative estimate, as it does not factor in the

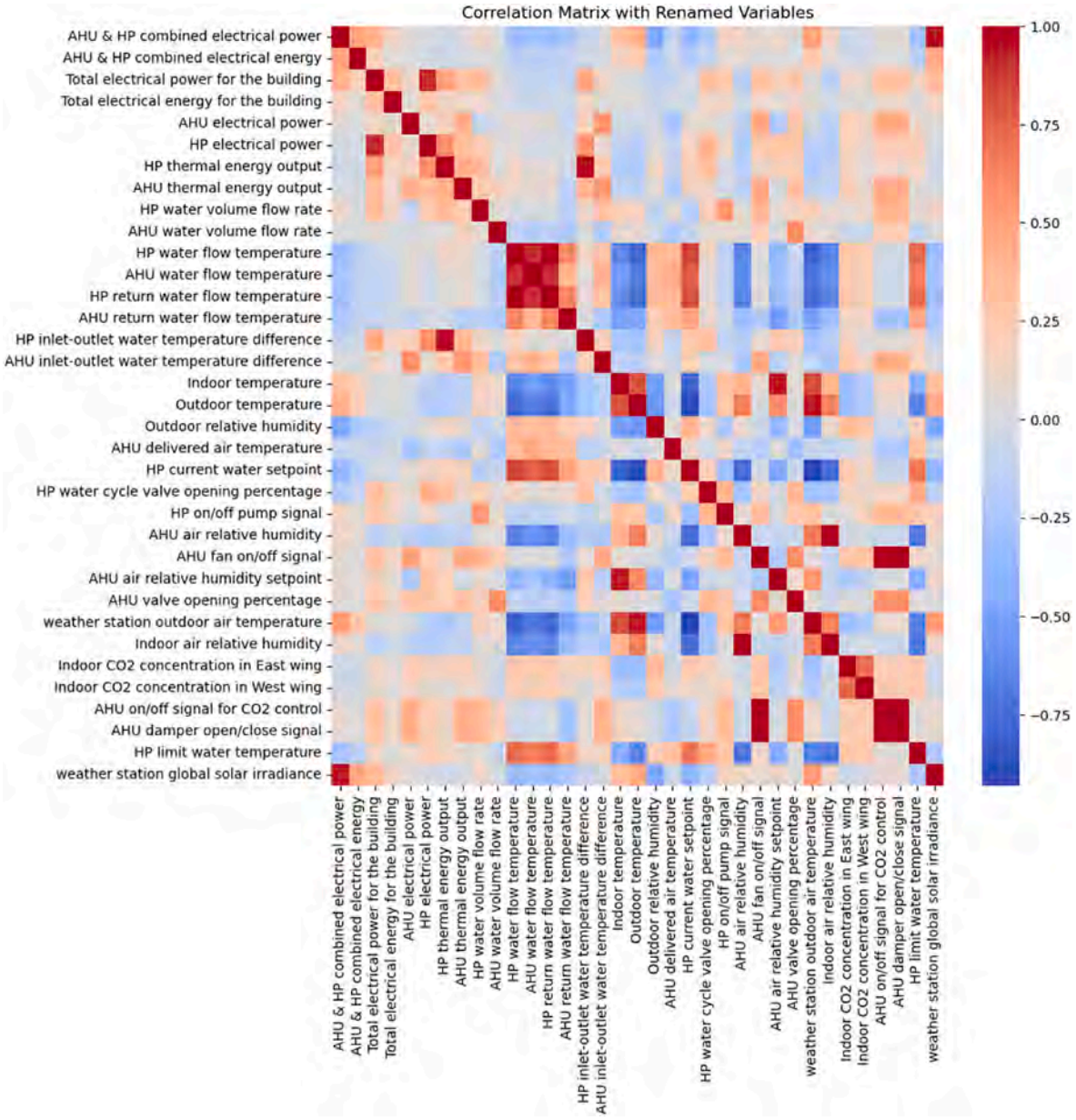

Fig. 6. Correlation between the features.

additional computational load imposed by the autoregressive nature of the decoder, which cannot be parallelized across sequence positions. We provide an illustrative representation of our model's universal structure in Fig. 1, applicable to an array of tasks. Table 1 provides a detailed breakdown of each layer within our model, its functionality, and the extent of its utilization within the core architecture. The reader is directed to the seminal transformer literature for a comprehensive elucidation of the model, whilst this discourse will focus on the modifications, we introduced to facilitate the processing of multivariate temporal sequences as opposed to linguistic token sequences.

Each datum for training, denoted as X within the real value space $\mathbb{R}^{w*m}$, represents a multivariate temporal sequence comprising w instances across m distinct variables, thus forming a series of feature vectors x_t within $\mathbb{R}^m$. Prior to dimensionality transformation, the feature vectors x_t are subjected to a normalization process—subtracting the mean and scaling by the variance computed across the training dataset—and subsequently projected linearly into a d dimensional vector space, d being the inherent dimensionality of the transformer's internal sequence representation, often referred to as the model dimension:

$$u_t = W_p x_t + b_p \tag{6}$$

Herein $W_p \in \mathbb{R}^{d*m}$ and $b_p \in \mathbb{R}^d$ are parameter matrices and vectors subject to optimization, with $u_t \in \mathbb{R}^d$ representing the series of model inputs analogous to the lexical embeddings in linguistic transformers. These inputs are subsequently transformed into the queries, keys, and values for the self-attention mechanism upon integration of positional encodings and subsequent application of the associated transformation matrices.

The transformer, inherently a feed-forward construct, lacks innate sensitivity to input sequence order. To instill an awareness of temporal structure within the model, we introduce positional encodings $W_{pos} \in \mathbb{R}^{w*d}$ into the input vector sequence $U \in \mathbb{R}^{w*d} = [u_1, u_2 \ldots u_w]$, thereby obtaining $Z = U + W_{pos}$.

In a departure from the fixed, sinusoidal positional encodings posited in the original transformer paradigm, our model utilizes a set of positional encodings that are subject to optimization. This alteration is substantiated by empirical evidence indicating enhanced performance

Table 2
The measurements from the system that is used in the training of the model.

System component	Measurement	Measurement Uncertainty	
Heat pump	1 Forward temperature	Electric power	±1 %
	2 Forward temperature set point.	7. Water temperature	±0.12 °C
	3 Return temperature.		
	4 Electric power consumption	8. Dry bulb temperature	±0.5 °C
	5 Water flow rate		
	6 Modulating signal of the mixing valve		
Air handling unit	1 Forward temperature	Water flow rate	±2 %
	2 Return temperature.		
	3 Electric power consumption		
	4 Water flow rate		
Outdoor conditions	1 Dry bulb temperature	Relative humidity	±2 %
	2 Relative humidity		
Indoor conditions	1 Dry bulb temperature	CO_2 concentration	±50 ppm
	2 CO_2 concentration		

across all considered datasets. These learnable encodings seem to minimally interfere with the temporal data's quantitative attributes. We postulate that this is attributable to the encodings evolving to occupy a vector subspace that is approximately orthogonal to that of the time series data, a hypothesis supported by the higher-dimensional nature of the embedding space which simplifies the attainment of orthogonality. In this study, time2vec method [47] was used to encode time stamps in the data.

$$\mathbf{t2v}(\tau)[i] = \begin{cases} \omega_i\tau + \varphi_i, & if\ i = 0 \\ \mathcal{F}(\omega_i\tau + \varphi_i), & if\ 1 \le i \le k \end{cases} \quad (7)$$

Where $\mathbf{t2v}(\tau)[i]$ is the i^{th} element of $\mathbf{t2v}(\tau)$. $\mathcal{F}$ is a periodic activation function and ω_i s and φ_i s are learnable parameters.

Time series data is inherently variable in length. Our architecture effectively addresses this heterogeneity by establishing a uniform maximum sequence length w for the dataset. Sequences falling short of this length are augmented. The model was trained on a window size of 96 corresponding to Ine day of measurement. We used 2 layers of transformer encoders and 2 layers of feed forward unit of encoders. While 7 heads were used in the multi head attention.

3.2. Self-supervised learning pre-training

For the foundational self-supervised pre-training phase of our model, we engage an autoregressive task wherein a portion of the input data is occluded with zeros, compelling the model to predict the concealed information. This process entails the systematic obscuration of subsets of the input sequence—achieved through the multiplication of the input $X \in \mathbb{R}^{w*m}$ with binary mask M, generated independently for each sample. In this masking schema, a proportion r of each mask column (equivalent to a singular variable in the time series) oscillates between segments of zeros and ones, following a predetermined state transition probability distribution to determine the length of each obfuscated segment, thereby generating sequences with a geometric distribution characterized by a mean unmasked segment length l_u and a mean masked segment length l_m , as given by $l_u = \frac{1-r}{r} l_m$ with l_m being set to 3

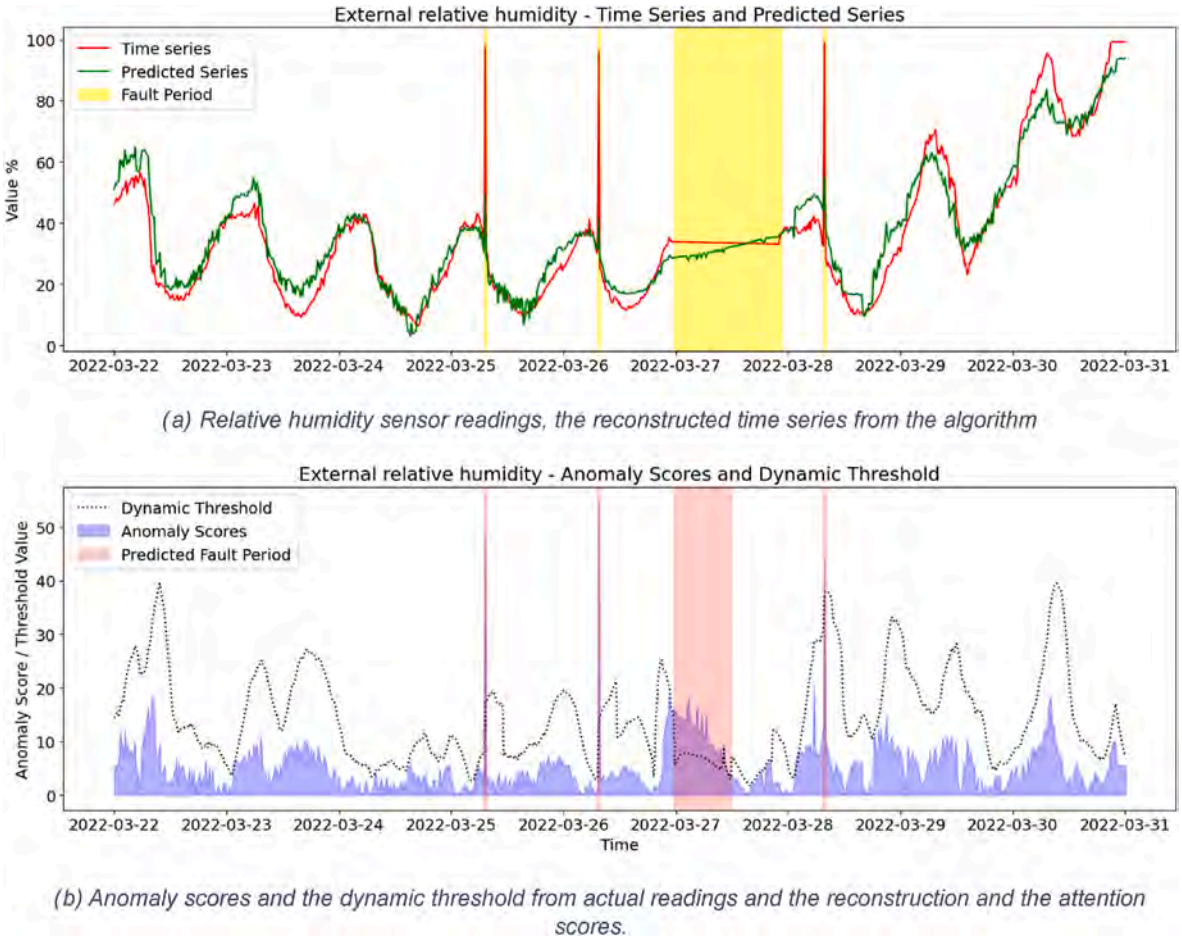

Fig. 7. Demonstration of the fault detection process in case of both sequential and point anomalies.

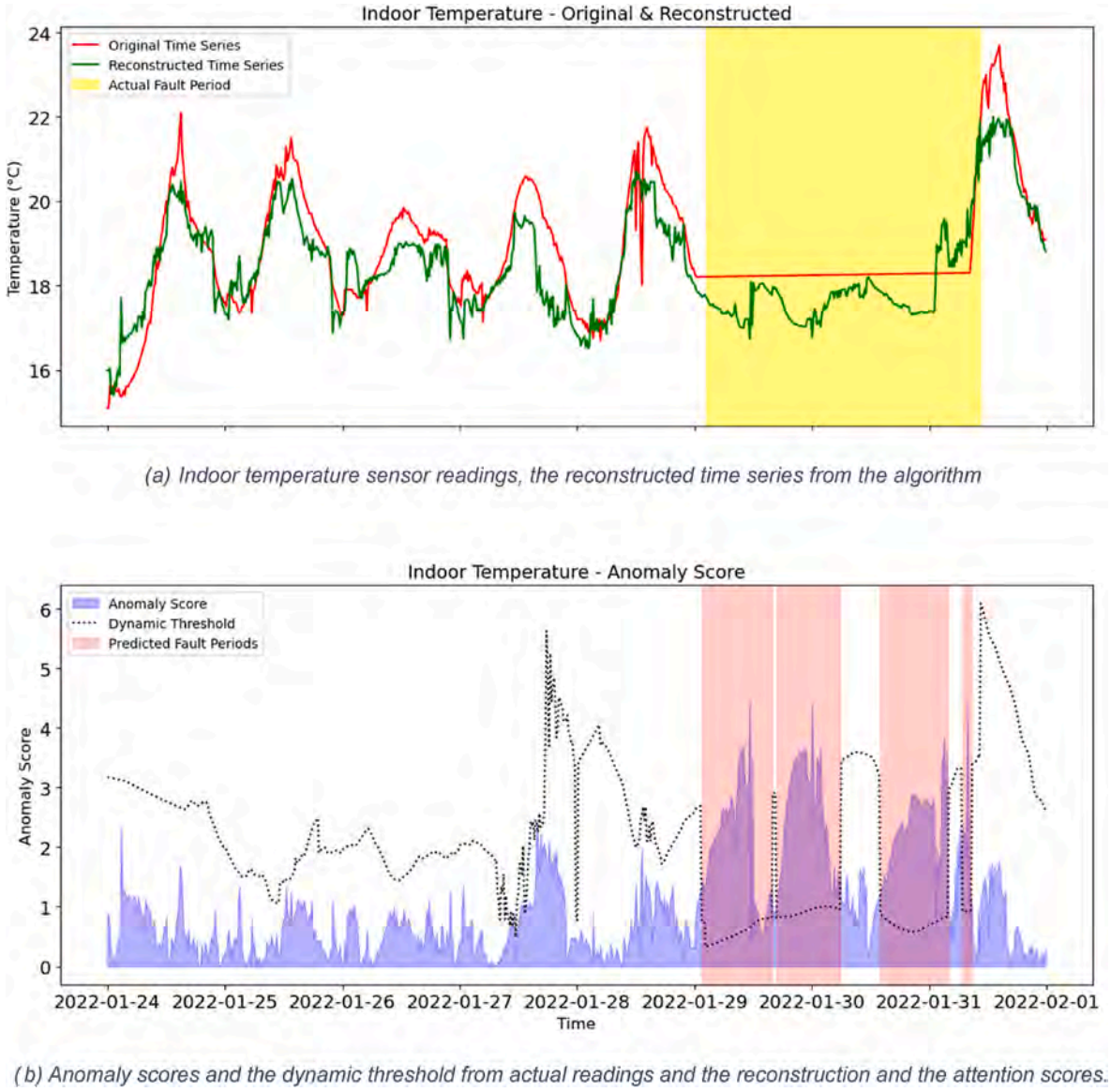

Fig. 8. Indoor temperature sensor readings, the reconstructed time series from the algorithm and the anomaly scores as a demonstration of sequential anomaly in the readings.

for the conducted experiments.

As shown in Fig. 2, we adopt this particular masking strategy—distinct from the "cloze" method employed in NLP models such as BERT—where the masked values in the time series are supplanted by zeros, as opposed to replacing word embeddings. This method is designed to incite the model to not only predict the immediate succeeding values but also to integrate the temporal dependencies between variables.

A linear layer with optimizable parameters $W_O \in \mathbb{R}^{m\times d}, b_O \in \mathbb{R}^m$ s applied to the terminal vector representations $z_t \in \mathbb{R}^d$ at each time step, with the model simultaneously estimating the complete unobscured input vectors x_t ; however, the Mean Squared Error (MSE) is computed solely for the predictions on the masked segments as indicated by the mask set $M = \{t_i : m_{ti} = 0\}$ where m_{ti} are the elements of the mask M. The MSE for each data sample is as follows:

$$\widetilde{x}_t = W_O z_t + b_O$$
$$L_{MSE} = \frac{1}{|M|}\sum_{(t,i)\in M}(\widetilde{x}(t,i) - x(t,i))^2 \tag{8}$$

This pre-training objective is methodologically divergent from denoising autoencoders as it does not consider the entire input reconstruction but rather focuses on the masked segments. Notably, this approach is not reliant on assumptions of noise characteristics typically postulated in denoising paradigms, such as Gaussian distributions. The design also takes into account the distributions of the actual masked values and the subsequent impact on learning.

3.3. Dynamic thresholding and fine tuning

After the reconstruction of the multivariate time series, the anomaly scores are calculated using the absolute difference between the original and the predicted ones, multiplied by the average attention weights of each window averaged over multiple heads. Dynamic thresholding technique is then applied to the anomaly scores to flag anomalies that exceed the threshold.

In this study, we implement the Peak Over Threshold (POT) method, which enables the automatic and dynamic selection of thresholds [48] and used by Ref. [41]. This technique is grounded in the principles of extreme value theory, facilitating the fitting of data distributions using a

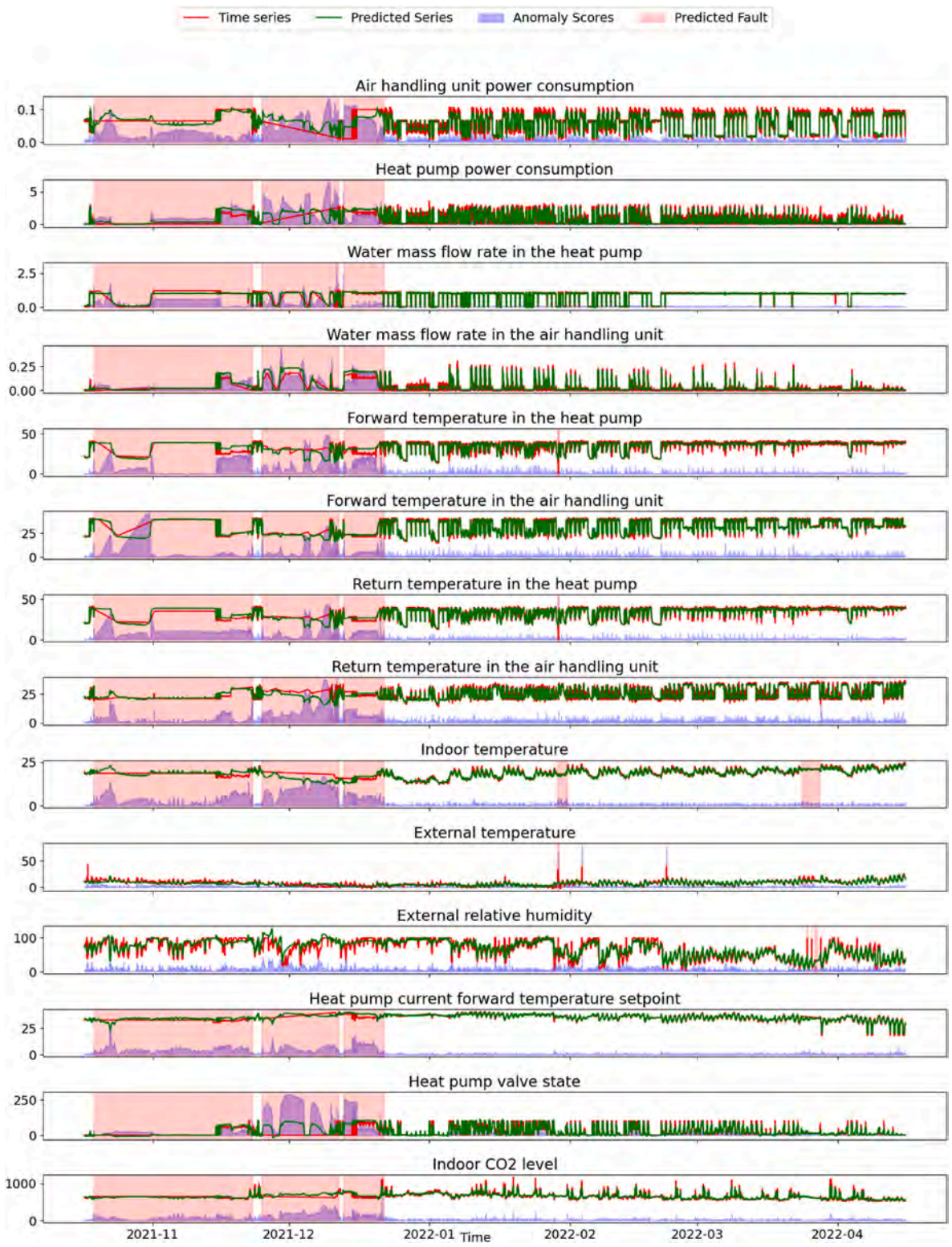

Fig. 9. Summary of the outcome for all the features. The highlighted areas are the periods labeled as faults.

Generalized Pareto Distribution. The first step in the POT method is to set an initial threshold t in a window w. This threshold is set such that only the most extreme values in the data set for each window are considered for further analysis. In this research this value was chosen as 85th percentile with a window of 2 h. Once the threshold is set, the method focuses on the excesses over this threshold. These excesses are defined as:

$$Y_i = X_i - t \tag{9}$$

Where Y_i is the excess over the threshold, X_i is the individual anomaly score at a certain time stamp and t is the initial threshold. The distribution of excesses over the threshold is fitted to generalized pareto distribution (GPD). The GPD is characterized by two parameters: scale parameter σ and the shape parameter γ. The cumulative distribution

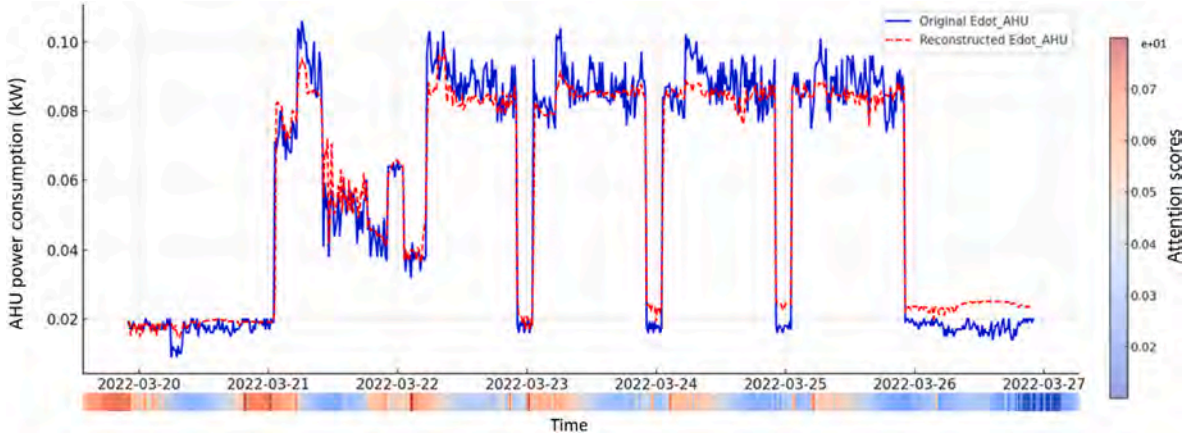

Fig. 10. Original time series vs reconstructed time series vs reconstructed for AHU power consumption with average attention scores for every time step.

function (CDF) of GPD is given by:

$$G(y;\sigma,\gamma)=\begin{cases}1-\left(1+\frac{\gamma y}{\sigma}\right)^{-\frac{1}{\gamma}}, & \text{if } \gamma\neq 0\\ 1-\exp\left(-\frac{y}{\sigma}\right), & \text{if } \gamma=0\end{cases} \tag{10}$$

where $y>0,\sigma>0,$ and $y\leq-\frac{\sigma}{\gamma}$ for $\gamma<0$. The σ,γ are the scale and the shape parameters respectively. Those parameters are estimated using the Maximum Likelihood Estimation (MLE). In this paper we used the Grimshaw tricks [49] to calculate the maximum value of maximum likelihood function.

Once the GPD parameters are estimated, the distribution in each window can be used to assess the extremeness of new observations. An observation is flagged as an anomaly if its excess over the threshold has a low probability under the estimated GPD. This is typically done by computing the quantile or the survival function of the GPD for a new observation and comparing it to a pre-defined risk level q. If the probability of observing an excess over the threshold is lower than q.

Finally, the quantiles are calculated for a given probability level using the inverse of the GPD's. CDF. This quantile represents the value for which there is a probability q that the observed value will exceed it. The formula for quantile calculation under GPD is:

$$z_q=t+\frac{\sigma}{\gamma}((1-q)^{-\gamma}-1)\text{ for }\gamma\neq 0 \tag{11}$$

Where z_q is the quantile for a probability of q. The calculated quantiles will be used as the calculated threshold for the anomaly scores. In the case of $\gamma=0$, the quantile is calculated using the exponential distribution formula. This dynamic threshold is set for each feature's anomaly score. An anomaly is flagged if the anomaly score of any feature exceeded the threshold.

Lastly, if labeled data becomes available, the model can be fine-tuned to further refine its fault detection capabilities. During fine-tuning, the labeled data is used to adjust the encoder's weights through back-propagation, specifically training the model to better classify instances as normal or faulty. This step enhances the model's precision in identifying faults by leveraging direct feedback from the labeled examples.

4. Case study

4.1. Building envelope and systems

VELUXlab (see Fig. 3) stands as the pioneering Nearly Zero Energy Building in Italy, situated within the confines of a university campus. The journey of VELUXlab began in 2011 when VELUX embarked on a project to transform the Atika demo-house into an innovative laboratory under the auspices of Politecnico di Milano. Initially designed to exemplify a model home suitable for the Mediterranean climate, the building underwent significant enhancements under the expert guidance of Politecnico di Milano's design team. These upgrades transformed it into an active prototype, offering a tangible example for the development of future sustainable buildings [50].

The retrofit process of the building involved both the improvement of the envelope's layering with new and high performances materials that increased the technical performances of the building case (U-values up to 0,124 W/m^2/K), and the implementation of systems. Static and dynamic simulations helped to calibrate the design choices to lead through the minimization of energy needs [50].

The HVAC system is comprised of air water heat pump as a generation source in the system with 7 kW in heating and 6.1 in cooling. As a mechanical ventilation and emission system, air handling unit with maximum flow rate of 470 m^3/h with over 90 % heat recovery. Radiant floor is also used as an emission system with capacity of 90 W/m^2 for heating and 30 W/m^2 for cooling. 11 m^2 of photovoltaic panels are used, the field is capable of generating 2 kWp. HVAC system implementation shown in Fig. 4.

The HVAC system undergoes continuous monitoring to evaluate the efficiency of its components and to optimize system control, thereby ensuring optimal indoor comfort. This monitoring framework incorporates a range of sensors, including those for temperature, relative humidity, and CO2, as well as heat and electrical meters. Fig. 5 presents a schematic representation of the HVAC system, highlighting the specific locations of these sensors.

4.2. Data description

In this study, we utilized operational data from the system spanning October 15, 2021, to April 15, 2022. The monitoring system initially incorporated over 50 sensors, measuring both numerical and discrete variables. To ensure data integrity, sensors exhibiting more than 15 % missing values were excluded, effectively narrowing down the dataset to the most complete time series. The missing values were the results of sensors not logging the values either due to communication errors or sensors malfunction. Further refinement was achieved by evaluating feature correlations; in instances where feature pairs demonstrated a correlation exceeding 95 %, one feature from each pair was removed to reduce computational complexity. The data has a range of time steps from 1 to 10 min then later resampled to 15 min. The resampling was done by mean aggregation to preserve central tendency of the time series. Analyzing the frequency content and probability distribution of the data post-resampling, we confirmed that the mean aggregation process did not introduce significant artifacts or biases. Fig. 6 shows the correlation matrix among the features. Correlated features such as AHU damper signal, AHU CO_2 control state and AHU fan signal were detected

and chosen from. Moreover, features were representing the same measurement but from different monitoring systems -such as external weather station outdoor temperature reading and internal monitoring system reading for external temperature-were detected. This process resulted in the selection of 14 features for subsequent analysis, as detailed in Table 2.

5. Results and discussion

As explained previously, anomalous periods are flagged subsequent to the computation of anomaly scores for each feature, upon the application of Peak-Over-Threshold (POT) thresholding. A timestep is classified as anomalous if it surpasses the threshold for any feature. Fig. 7 provides a comprehensive visualization of the outcomes for each feature, along with the anomalies identified by the algorithm.

Since no labeled data is available, the assessment of the method was done through two metrics. First how well the reconstructed time series matches the original in the features and the second is analyzing the flagged instances. In general, for time series fault instances, there are two types of anomalies, point and sequential. The point anomalies are mostly labeled correctly in the data as demonstrated in Fig. 7 where multiple point anomalies in the relative humidity sensor readings are correctly labeled. Those anomalies are quite common in sensor readings and easy to detect for most anomaly detection algorithms. Sequential anomalies on the other hand are much harder to detect and label by anomaly detection algorithms, since it requires identifying the underlying trend or multiple trends in the data and detect the deviation from it. In Figs. 7 and 8, sequential anomalies are apparent in the data as the cyclical nature of the trend stops and a non-zero linear trend starts for a period of time. The method proposed was able to detect the change in the trend correctly but not for the entirety of the anomalous period. Since the method is primarily built on the anomaly score, which is a function of the reconstruction error, when the reconstruction error reach zero on the points where there is intersection between the reconstructed time series and the original one, the anomaly score reaches zero preventing the continuation of the detection.

In Fig. 7, we can see that while the beginning of the fault correctly detected with a spike of anomaly score, the detection stopped when the reconstruction error reached zero and 3 h of the fault was not detected. In Fig. 8, the same behavior appeared in the indoor sensor readings. Four different anomalous periods were detected and a total of 6 h out of 50 h were detected.

The highlighted intervals within Fig. 9 are instances where the anomaly score exceeded the designated threshold, signaling a fault. A notable aggregation of such faults is observable between October 19th and December 23rd. Upon scrutiny of this interval, a significant malfunction within the monitoring system was revealed, impacting all sensors with the exception of those associated with the weather station that records external dry bulb temperature and relative humidity. This malfunction led to shifting of the measurement trend to be linear instead of noisy cyclical.

Since the model has a window size of 96 which represents one day, the model was able to reconstruct the trends well as shown previously, however, some peaks were not captured in the same precision due to the window size choice. A smaller window choice might solve this issue but will increase the computation cost and compromise capturing the longer trends in different dimensions. A future solution might be to have a dual encoder with different window sizes. Despite the fact that this will lead to increased computational cost, the results should attend to both long and short trends given a correct way of combining the outcome form the dual encoders.

Fig. 10 visualizes the average attention weights of each window averaged over multiple heads. It is apparent that there is a high correlation between the attention weights and peaks and sudden changes in the time series. Analyzing the attention weights across different dimensions it was also noticed that the model higher attention weights to different dimensions where the deviations are higher, allowing the model to specifically detect faults in each dimension individually with the contextual trend of the complete sequence as prior.

6. Conclusions

Data-driven technologies are pivotal in the efficient operation of smart buildings. However, the scarcity of adequately labeled data presents significant obstacles in developing dependable data-driven approaches for diagnosing faults in building systems. This research introduces an innovative self-supervised approach that leverages unlabeled operational data from buildings for the purpose of fault detection and aid in diagnostics, moving beyond the sole dependence on labeled data sources. We introduced an encoder only transformer model for fault detection in multi variate time series. The model has been tested against real data from a case study of a university building with an HVAC system composed of a heat pump as a generation system connected to air handling unit and floor heating as an emission system. The data consists of 14 different features from the building and from an external weather station. The self-supervised training was done by strategically masking portions of the multivariate time-series data using Markov chain approach with two states. The model is trained by predicting these concealed segments. After the pretraining task the model uses a feed forward neural network to reconstruct the original multi-variate time series back to the original dimensions and then the reconstructed results are compared to the original data. We implemented the Peak Over Threshold (POT) method, which enables the automatic and dynamic selection of thresholds. For the purpose of anomaly detection, we aid an anomaly diagnosis labeling by pointing to the feature where the anomaly was detected. This results in a light model since only the encoder is used and since the transformer architecture allows for parrel computation on the sequences. Also, the model does not require any labeled data, moreover any labeled data available can be used for fine tuning. Applying the model to the case study, a number of faults were detected mainly due to malfunctioning of the monitoring system. The period from October 19th to December 23rd was detected as faulty. The model combined with the dynamic thresholding showed the ability to detect both sequential and point faulty operation. Analyzing the attention weights, it was found that the model gives higher attention to peaks and sudden changes in the data. Also, higher attention is given to the dimensions in the data with higher deviations. Some limitations and room for improvements were also noticed. Namely the model not able to capture the trends within a day with the same precision as for the longer trends which to solve we propose a dual encoder with different window sizing and specific concatenation of the results to capture both longer and shorter trends.

CRediT authorship contribution statement

M.A.F. Abdollah: Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **R. Scoccia:** Writing – review & editing, Data curation, Conceptualization. **M. Aprile:** Writing – review & editing, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

References

[1] S. Katipamula, M.R. Brambley, Review article: methods for fault detection, diagnostics, and prognostics for building systems—a review, part I, HVAC R Res. 11 (1) (2005) 3–25, https://doi.org/10.1080/10789669.2005.10391123.

[2] K. Cant, R. Evins, Improved calibration of building models using approximate Bayesian calibration and neural networks, J Build Perform Simul 16 (3) (May 2023) 291–307, https://doi.org/10.1080/19401493.2022.2137236.

[3] H. Kramer, G. Lin, C. Curtin, E. Crowe, J. Granderson, Building analytics and monitoring-based commissioning: industry practice, costs, and savings, Energy Effic 13 (3) (Mar. 2020) 537–549, https://doi.org/10.1007/S12053-019-09790-2/TABLES/3.

[4] G. Li, et al., Interpretation of convolutional neural network-based building HVAC fault diagnosis model using improved layer-wise relevance propagation, Energy Build. 286 (May 2023) 112949, https://doi.org/10.1016/J.ENBUILD.2023.112949.

[5] V. Singh, J. Mathur, A. Bhatia, A comprehensive review: fault detection, diagnostics, prognostics, and fault modeling in HVAC systems, Int. J. Refrig. 144 (Dec. 2022) 283–295, https://doi.org/10.1016/J.IJREFRIG.2022.08.017.

[6] Z. Du, X. Liang, S. Chen, X. Zhu, K. Chen, X. Jin, Knowledge-infused deep learning diagnosis model with self-assessment for smart management in HVAC systems, Energy 263 (Jan. 2023) 125969, https://doi.org/10.1016/J.ENERGY.2022.125969.

[7] L. Wang, J. Braun, S. Dahal, An evolving learning-based fault detection and diagnosis method: case study for a passive chilled beam system, Energy 265 (Feb. 2023) 126337, https://doi.org/10.1016/J.ENERGY.2022.126337.

[8] Y. Zhao, T. Li, X. Zhang, C. Zhang, Artificial intelligence-based fault detection and diagnosis methods for building energy systems: advantages, challenges and the future, Renew. Sustain. Energy Rev. 109 (Jul. 2019) 85–101, https://doi.org/10.1016/J.RSER.2019.04.021.

[9] Z. Chen, et al., A review of data-driven fault detection and diagnostics for building HVAC systems, Appl. Energy 339 (Jun. 2023) 121030, https://doi.org/10.1016/J.APENERGY.2023.121030.

[10] H. Han, X. Cui, Y. Fan, H. Qing, Least squares support vector machine (LS-SVM)-based chiller fault diagnosis using fault indicative features, Appl. Therm. Eng. 154 (May 2019) 540–547, https://doi.org/10.1016/J.APPLTHERMALENG.2019.03.111.

[11] Y. Gao, H. Han, Z.X. Ren, J.Q. Gao, S.X. Jiang, Y.T. Yang, Comprehensive study on sensitive parameters for chiller fault diagnosis, Energy Build. 251 (Nov. 2021) 111318, https://doi.org/10.1016/J.ENBUILD.2021.111318.

[12] Y. Zhao, J. Wen, F. Xiao, X. Yang, S. Wang, Diagnostic Bayesian networks for diagnosing air handling units faults – part I: faults in dampers, fans, filters and sensors, Appl. Therm. Eng. 111 (Jan. 2017) 1272–1286, https://doi.org/10.1016/J.APPLTHERMALENG.2015.09.121.

[13] X. Fang, et al., Cross temporal-spatial transferability investigation of deep reinforcement learning control strategy in the building HVAC system level, Energy 263 (Jan. 2023) 125679, https://doi.org/10.1016/J.ENERGY.2022.125679.

[14] Q. Zhang, Z. Tian, Y. Lu, J. Niu, C. Ye, Experimental study on performance assessments of HVAC cross-domain fault diagnosis methods oriented to incomplete data problems, Build. Environ. 236 (May 2023) 110264, https://doi.org/10.1016/J.BUILDENV.2023.110264.

[15] C. Fan, W. He, Y. Liu, P. Xue, Y. Zhao, A novel image-based transfer learning framework for cross-domain HVAC fault diagnosis: from multi-source data integration to knowledge sharing strategies, Energy Build. 262 (May 2022) 111995, https://doi.org/10.1016/J.ENBUILD.2022.111995.

[16] X. Zhu, K. Chen, B. Anduv, X. Jin, Z. Du, Transfer learning based methodology for migration and application of fault detection and diagnosis between building chillers for improving energy efficiency, Build. Environ. 200 (Aug. 2021) 107957, https://doi.org/10.1016/J.BUILDENV.2021.107957.

[17] J. Zhang, Y. Xu, H. Chen, L. Xing, A novel building heat pump system semi-supervised fault detection and diagnosis method under small and imbalanced data, Eng. Appl. Artif. Intell. 123 (Aug. 2023) 106316, https://doi.org/10.1016/J.ENGAPPAI.2023.106316.

[18] K. Yan, C. Zhong, Z. Ji, J. Huang, Semi-supervised learning for early detection and diagnosis of various air handling unit faults, Energy Build. 181 (Dec. 2018) 75–83, https://doi.org/10.1016/J.ENBUILD.2018.10.016.

[19] C. Fan, Y. Liu, X. Liu, Y. Sun, J. Wang, A study on semi-supervised learning in enhancing performance of AHU unseen fault detection with limited labeled data, Sustain. Cities Soc. 70 (Jul. 2021) 102874, https://doi.org/10.1016/J.SCS.2021.102874.

[20] B. Li, F. Cheng, X. Zhang, C. Cui, W. Cai, A novel semi-supervised data-driven method for chiller fault diagnosis with unlabeled data, Appl. Energy 285 (Mar. 2021) 116459, https://doi.org/10.1016/J.APENERGY.2021.116459.

[21] B. Li, F. Cheng, H. Cai, X. Zhang, W. Cai, A semi-supervised approach to fault detection and diagnosis for building HVAC systems based on the modified generative adversarial network, Energy Build. 246 (Sep. 2021) 111044, https://doi.org/10.1016/J.ENBUILD.2021.111044.

[22] K. Zhang, et al., "Self-Supervised Learning for Time Series Analysis: Taxonomy, Progress, and Prospects,", Jun. 2023 [Online]. Available: https://arxiv.org/abs/2306.10125v2. (Accessed 12 November 2023).

[23] A. Vaswani, et al., Attention is all you need, Adv. Neural Inf. Process. Syst. 2017-December (Jun. 2017) 5999–6009 [Online]. Available: https://arxiv.org/abs/1706.03762v7. (Accessed 2 November 2023).

[24] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, vol. 1, Oct. 2018, pp. 4171–4186 [Online]. Available: https://arxiv.org/abs/1810.04805v2. (Accessed 2 November 2023).

[25] A. Dosovitskiy, et al., An image is worth 16x16 words: transformers for image recognition at scale, in: ICLR 2021 - 9th International Conference on Learning Representations, Oct. 2020 [Online]. Available: https://arxiv.org/abs/2010.11929v2. (Accessed 2 November 2023).

[26] D. Wang, X. Wang, S. Lv, An overview of end-to-end automatic speech recognition, Symmetry 2019 11 (8) (Aug. 2019) 1018, https://doi.org/10.3390/SYM11081018. Vol. 11, Page 1018.

[27] M. Han, X. Zhang, X. Yuan, J. Jiang, W. Yun, C. Gao, A survey on the techniques, applications, and performance of short text semantic similarity, Concurr. Comput. 33 (5) (Mar. 2021) e5971, https://doi.org/10.1002/CPE.5971.

[28] K. Han, et al., A survey on vision transformer, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (Jan. 2023) 87–110, https://doi.org/10.1109/TPAMI.2022.3152247.

[29] Y. Tay, M. Dehghani, D. Bahri, D. Metzler, Efficient transformers: a survey, ACM Comput. Surv. 55 (6) (Sep. 2020), https://doi.org/10.1145/3530811.

[30] P. Dufter, M. Schmitt, H. Schütze, Position information in transformers: an overview, Comput. Ling. 48 (3) (Feb. 2021) 733–763, https://doi.org/10.1162/coli_a_00445.

[31] Q. Wen, et al., Transformers in time series: a survey, IJCAI International Joint Conference on Artificial Intelligence 2023-August (Feb. 2022) 6778–6786, https://doi.org/10.24963/ijcai.2023/759.

[32] S. Li, et al., Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, Adv. Neural Inf. Process. Syst. 32 (Jun) (2019) [Online]. Available: https://arxiv.org/abs/1907.00235v3. (Accessed 11 November 2023).

[33] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, C. Eickhoff, A transformer-based framework for multivariate time series representation learning, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Oct. 2020, pp. 2114–2124, https://doi.org/10.1145/3447548.3467401.

[34] B. Lim, S. Arık, N. Loeff, T. Pfister, Temporal Fusion Transformers for interpretable multi-horizon time series forecasting, Int. J. Forecast. 37 (4) (Oct. 2021) 1748–1764, https://doi.org/10.1016/J.IJFORECAST.2021.03.012.

[35] H. Zhou, et al., Informer: beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, May 2021, pp. 11106–11115, https://doi.org/10.1609/AAAI.V35I12.17325, 12.

[36] H. Wu, J. Xu, J. Wang, M. Long, Autoformer: decomposition transformers with auto-correlation for long-term series forecasting, Adv. Neural Inf. Process. Syst. 34 (Dec. 2021) 22419–22430 [Online]. Available: https://github.com/thuml/Autoformer. (Accessed 11 November 2023).

[37] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, R. Jin, FEDformer: frequency enhanced decomposed transformer for long-term series forecasting, Proc Mach Learn Res 162 (Jan. 2022) 27268–27286 [Online]. Available: https://arxiv.org/abs/2201.12740v3. (Accessed 11 November 2023).

[38] S. Li, et al., Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, Adv. Neural Inf. Process. Syst. 32 (Jun) (2019) [Online]. Available: https://arxiv.org/abs/1907.00235v3. (Accessed 11 November 2023).

[39] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, Pyraformer: low-complexity pyramidal attention for long-range time series modeling and forecasting, Int. J. Forecast. 36 (3) (Oct. 2021) 1181–1191, https://doi.org/10.1016/j.ijforecast.2019.07.001.

[40] J. Xu, H. Wu, J. Wang, M. Long, Anomaly transformer: time series anomaly detection with association discrepancy, in: ICLR 2022 - 10th International Conference on Learning Representations, Oct. 2021 [Online]. Available: https://arxiv.org/abs/2110.02642v5. (Accessed 12 November 2023).

[41] S. Tuli, G. Casale, N.R. Jennings, TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data, Jan. 2022, https://doi.org/10.14778/3514061.3514067.

[42] X. Wang, D. Pi, X. Zhang, H. Liu, C. Guo, Variational transformer-based anomaly detection approach for multivariate time series, Measurement 191 (Mar. 2022) 110791, https://doi.org/10.1016/J.MEASUREMENT.2022.110791.

[43] U. Yokkampon, A. Mowshowitz, S. Chumkamon, E. Hayashi, Robust unsupervised anomaly detection with variational autoencoder in multivariate time series data, IEEE Access 10 (2022) 57835–57849, https://doi.org/10.1109/ACCESS.2022.3178592.

[44] D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: 2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings, Dec. 2013 [Online]. Available: https://arxiv.org/abs/1312.6114v11. (Accessed 12 November 2023).

[45] I.J. Goodfellow, et al., Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014) [Online]. Available: http://www.github.com/goodfeli/adversarial. (Accessed 12 November 2023).

[46] Z. Chen, D. Chen, X. Zhang, Z. Yuan, X. Cheng, Learning graph structures with transformer for multivariate time-series anomaly detection in IoT, IEEE Internet Things J. 9 (12) (Jun. 2022) 9179–9189, https://doi.org/10.1109/JIOT.2021.3100509.

[47] S.M. Kazemi, et al., Time2Vec: learning a vector representation of time. https://arxiv.org/abs/1907.05321v1, Jul. 2019. (Accessed 19 November 2023) [Online]. Available.

[48] A. Siffer, P.A. Fouque, A. Termier, C. Largouet, Anomaly detection in streams with extreme value theory, in: Proceedings of the ACM SIGKDD International

Conference on Knowledge Discovery and Data Mining, Part F129685, Aug. 2017, pp. 1067–1075, https://doi.org/10.1145/3097983.3098144.

[49] S.D. Grimshaw, Computing maximum likelihood estimates for the generalized pareto distribution, Technometrics 35 (2) (1993) 185–191, https://doi.org/10.1080/00401706.1993.10485040.

[50] M. Imperadori, F. Brunone, Active House and user-friendly visualization of sensors' monitored data: VELUXlab, a real cognitive and smart NZEB prototype, in: IOP Conference Series: Earth and Environmental Science, Institute of Physics Publishing, Jul. 2019, https://doi.org/10.1088/1755-1315/296/1/012042.