FROM ATTENTION WEIGHTS TO ATTENTION ZONES: LOCALIZING FAULT-RELEVANT SIGNALS AND TIME INTERVALS IN MULTIVARIATE TIME-SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Root cause identification from multivariate time-series data in automotive systems is challenging because labels are typically available only at the vehicle level, not for individual signals or time steps, leading to a weakly supervised learning problem. This paper presents an attention-zone identification framework based on a hybrid convolutional–transformer architecture that jointly models local channel-wise patterns and global temporal structure to localize fault-relevant signals and time windows. Channel-wise CNN encoders preserve signal-level interpretability, while a transformer encoder enable attribution of the classification decision to specific channels ("attention channels") and time intervals ("attention zones") via attention weights, without post-hoc explanation methods. The method is evaluated on (i) a benchmark Racket Sports dataset augmented with synthetic localized faults and (ii) a large-scale vehicles dataset representing a complex real-world data analysis task. The results demonstrate that the proposed attention-zone framework provides robust, intrinsically interpretable root-cause insights for weakly supervised multivariate time-series diagnostics in complex automotive systems.

1 INTRODUCTION

In modern automotive systems, increasing complexity and integration of large number of components can create situations in which a failure occurs in a subset of the fleet without the diagnostic system being able to determine the cause. For these unexpected failure modes, root cause identification can be complex when the contributing factors to the failure are unknown. In such cases, a rigorous analysis of vehicle data is required to gather evidence and identify the contributing factors for each failure. An important element of this root cause identification activity is time-series (or sequentially structured) data collected before and after the fault happened. When analyzing the time series data (similar to the illustrative example shown in Figure 1), the user must identify the time intervals (windows) that contribute most to the failure, as well as the signals that provide the most informative evidence for root cause analysis. When this crucial information is identified, the investigation can be taken to next steps which is to deep dive into the identified relevant information (time intervals and signals).

In this paper, an attention-zone identification methodology is proposed that has two primary goals. First, the algorithm can identify the time instances (called attention zones) that play the most significant role in classifying a time series sample. Second, it determines which signals (channels) are most influential in the classification (called attention channels). The proposed methodology is formulated as a weakly supervised learning problem where labels are provided only at the vehicle level. The diagnostic model must then infer the important time steps and channels. Here, we propose to directly use the capability of attention networks to identify important time instances and channels because the attention weights capture the contribution of each token to the final decision. Attention mechanisms have become essential tools for interpreting deep learning models, particularly in sequence based classification tasks. In transformer architectures, each token in the input sequence contributes to the final prediction through a weighted aggregation, where attention weights quantify how much information the model extracts from each token. If one selects the classification output to rely on a designated token called the [CLS] token (i.e. an augmented token concatenated after CNN flattening stage as shown in Figure 2), its attention distribution naturally indicates token level impor-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

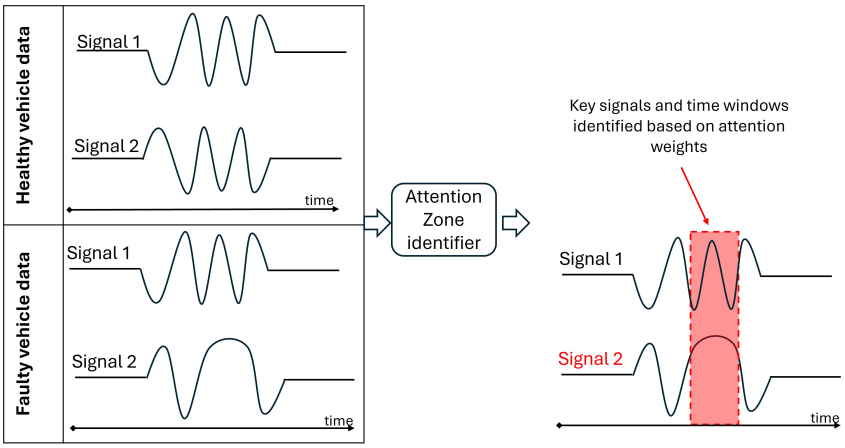


Figure 1: concept of time series data analysis and weekly supervised problems. The label is known only at the vehicle level and time-steps are not labeled. The algorithm needs to find the signal (here signal 2) and time steps at which the fault is mostly observable.

tance. Tokens receiving higher attention contribute more strongly to the model's decision, enabling fine grained attribution across channels, time steps, and learned features. In contrast to existing weakly supervised and Multi-instance learning (MIL)-based approaches for multivariate time-series analysis, the proposed method introduces a unified convolutional–transformer architecture that enables simultaneous and intrinsic identification of both fault-relevant channels and temporally localized attention zones within a single model. Unlike prior MIL formulations that rely on temporal instance aggregation or hierarchical bag structures to infer importance, the proposed approach preserves channel identity through dedicated channel-wise CNN encoders while directly leveraging transformer self-attention to attribute classification decisions to individual time steps and signals via the [CLS] token. This design eliminates the need for auxiliary pooling mechanisms or post-hoc explanation methods, ensuring that interpretability is an inherent property of the model architecture. Furthermore, by jointly tokenizing high-level channel summaries and raw time-series samples, the framework uniquely combines global signal-level abstraction with fine-grained temporal resolution under weak supervision. The effectiveness and robustness of this attention-zone identification strategy are demonstrated not only on benchmark datasets with controlled synthetic faults but also on a large-scale real-world automotive fleet dataset, highlighting its practical value for root-cause analysis in complex industrial systems where fine-grained annotations are unavailable.

2 Background and Related Work

Weakly supervised learning for multivariate time series analysis is commonly formulated using Multiple Instance Learning (MIL), where labels are provided at the sequence (bag) level while discriminative temporal regions (instances) remain unlabeled. Attention based MIL methods demonstrated that instance level attention weights can be learned jointly with bag level classification, enabling soft localization of informative segments without requiring dense annotations Ilse et al. (2018). Building on this foundation, recent work has explicitly adapted MIL frameworks to time series data by incorporating temporal structure and ordering. TimeMIL proposes a time aware MIL formulation for multivariate time series classification, showing that modeling temporal dependencies improves both classification accuracy and localization of relevant time windows Chen et al. (2024). Similarly, TAIL MIL extends time aware MIL to anomaly detection and learns instance level representations and attention weights under inexact supervision Jang & Kwon (2025). Other approaches, such as TreeMIL, explore hierarchical MIL constructions to localize anomalies in time series data when labels are only available at the sequence level Liu et al. (2024). In parallel, inherently interpretable MIL models have been introduced to impose architectural constraints that ensure learned instance importance directly corresponds to meaningful temporal segments, rather than relying on post hoc explanation techniques Early et al. (2024)]. More recent work combines attention mechanisms and transformer based architectures with MIL to enhance both performance and interpretability in time

series analysis. Attention based MIL approaches provide intrinsic explanations by assigning importance scores to individual time instances that directly influence the final prediction Ilse et al. (2018). Time aware extensions such as TimeMIL and TAIL MIL further demonstrate that explicitly modeling temporal relationships within attention mechanisms improves robustness and localization accuracy in multivariate sequences Chen et al. (2024); Jang & Kwon (2025). While attention mechanisms have also been explored extensively for tabular learning, for example in TabNet and broader studies revisiting deep learning models for tabular data Arik & Pfister (2021); Gorishniy et al. (2021), these methods assume independent feature vectors and do not explicitly capture sequential dependencies. In contrast, MIL based time series models emphasize joint temporal and instance level attribution, making them well suited for root cause identification problems where both the timing and the contributing signals must be identified. The method proposed in this paper follows this line of research by leveraging attention as an intrinsic explanatory mechanism under weak supervision to identify informative time intervals and channels in multivariate time series data.

3 PROPOSED NETWORK STRUCTURE

We introduce a hybrid convolutional–transformer architecture designed for multivariate time series classification with explicit channel level and temporal interpretability (Figure 2). The proposed framework consists of:

(i) channel-wise CNNs for local feature extraction,

(ii) a tokenization strategy that preserves the identities of channels and time steps, and

(iii) a transformer encoder whose attention weights enable attribution of model decisions to specific channels and time regions.

The input matrix $X \in R^{S \times L}$ is a batch of multivariate time-series samples, where L is the sequence length (or sampled points), and S is the number of channels. Before the attention mechanism, each channel X_s is treated as an independent univariate signal, ensuring well-defined interpretability at the channel level. To enable the transformer to jointly reason over global channel-level features and fine-grained temporal information, we construct a unified token sequence composed of three components:

(i) CNN-derived channel tokens. To capture channel (signal) contributions to the classification decision, each channel is processed by a dedicated convolutional subnetwork. For channel s, the corresponding 1-D CNN module receives the univariate sequence X_s and applies two convolutional layers with kernel size k, each followed by ReLU activation, dropout, and max pooling. An adaptive max pooling layer compresses the temporal dimension to a fixed length vector of size F, where F is the number of convolutional filters.
Formally, the output of the CNN stage is:

$$H_{cnn} = [h_1, h_2, \ldots, h_S]^T \in R^{S \times F} \tag{1}$$

where $h_i \in R^F$ denotes the feature vector extracted from channel i. This design ensures that each channel's representation remains isolated, enabling clean attribution of importance to individual channels. As illustrated in Figure 2, the tensor H_{cnn} is then flattened to shape $(1, S \cdot F)$ for each batch and expanded to form a block of tokens representing channel-level summaries. These tokens provide the transformer with a compact, high-level view of each channel's behavior.

(ii) Classification token. A learnable vector $[CLS] \in R^{1 \times 1}$ is prepended to the sequence. This token aggregates information from all other tokens and serves as the sole input to the final classifier in Figure 2. The $[CLS]$ token is initialized randomly and updated during training when it passes the attention layer.

(iii) Raw time-series tokens. The original sequence X is appended after the CNN tokens, preserving the temporal ordering and allowing the model to attend to specific time steps.

All tokens are projected into a shared d-dimensional embedding space using a linear layer. A learnable positional encoding is added to each token to preserve ordering across the combined sequence of length $M = 1 + S.F + L$.

The embedded token sequence is processed by a single-attention transformer encoder layer with multi-head self-attention to keep the network and explanation clear in this paper. The transformer is implemented by a sequence of input embedding, positional embedding, multi-head attention, and feedforward networks. We explicitly extract the attention weights from the self-attention module before the feed-forward sublayers. Given the query, key, and value matrices $Q \in R^{M\times D}, K \in R^{M\times D}, V \in R^{M\times D}$, the attention mechanism computes:

$$A : Attention(Q, K, V) = softmax(\frac{Q.K^\top}{\sqrt{d_k}})V. \tag{2}$$

The resulting attention matrix $A \in R^{H\times M}$, where H is the number of heads and M the token count, quantifies the contribution of each token to every other token. As we will discuss in the following, because the classification decision depends solely on the final representation of the CLS token, the row of A corresponding to CLS provides a direct measure of the importance of each channel and each time step. No additional pooling or aggregation is applied, ensuring that the classification decision is entirely mediated by the attention-based information flow into the CLS token.

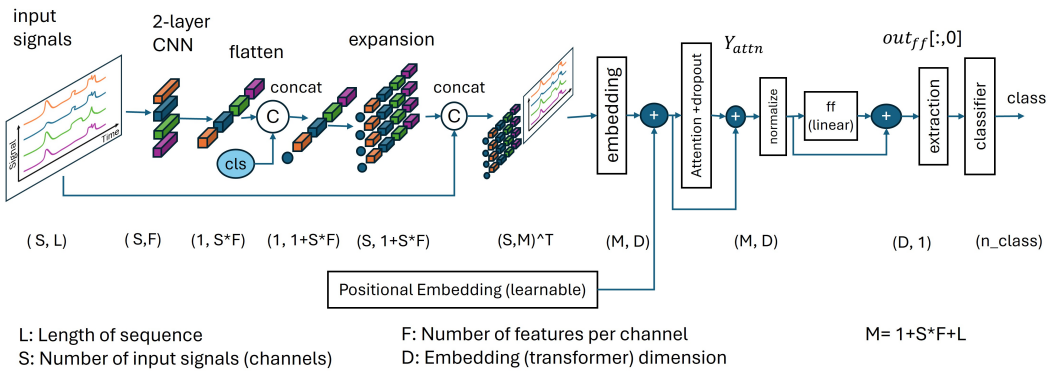


Figure 2: designed architecture for attention zone identifier.

3.1 Interpretability Through Attention-Based Attribution

The architecture of the network shown in Figure 2 is particularly designed so that attention weights are meaningful indicators of feature importance. Channel-wise CNNs ensure that each channel's high-level representation is isolated; raw tokens preserve temporal granularity. As a result, the network architecture ensures that attention weights reliably indicate feature importance:

- Attention from CLS $\rightarrow$ CNN tokens reveals channel importance.
- Attention from CLS $\rightarrow$ raw data tokens reveals temporal importance.

This enables the model to support root-cause identification in multivariate time-series classification tasks. To show how this happens in the network, we track the final classification output from Figure 2). For a single perceptron layer, the classification is performed using only the first row of the feedforward layer:

$$class = \Psi(out_{ff}[:, 0]) \tag{3}$$

where

$$Out_{ff} = FFN(Y_{attn}). \tag{4}$$

As eq. 4 shows, $out_{ff}[:, 0]$ is formed by the first row of Y_{attn}. As a note here that the linear operation in eq. 4 preserves the row-order (i.e. the first row of the output is formed by the first row of the input Y_{attn}). The transformer layer output $Y_{attn}[:, 0]$ is formed by the first row of the query matrix Q. To show that, one can reshape the query, key and value matrices as:

$$Q = [q_1, \ldots, q_M]^T, \quad q_i \in R^D \tag{5}$$

$$K^T = [k_1, \ldots, k_M], \quad k_i \in R^D \tag{6}$$

Then, the attention matrix will be:

$$\begin{aligned}\mathbf{Y}_{\text{attn}} &= (QK^T)V \\ &= \begin{bmatrix} q_1k_1 & \cdots & q_1k_M \\ \vdots & \ddots & \vdots \\ q_Mk_1 & \cdots & q_Mk_M \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{1D} \\ \vdots & \ddots & \vdots \\ v_{M1} & \cdots & v_{MD} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{M} q_1k_jv_{j1} & \cdots & \sum_{j=1}^{M} q_1k_jv_{jD} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^{M} q_Mk_jv_{j1} & \cdots & \sum_{j=1}^{M} q_Mk_jv_{jD} \end{bmatrix}.\end{aligned} \tag{7}$$

The self-attention mechanism in eq. 7 is presented schematically in Figure 3. As shown, the token representation (first row of the multi-head attention layer output) is formed by $[q_1k_j]$, $j : 1..M$ weighted by the value matrix V. Each row of the key matrix represents either an aggregated feature of the time series or the embedding of time instances from the raw signals. Therefore, by looking at the attention weights from q_1K^T, one can get insights of how much each signal (represented by its features) or time instance (of the raw input signal) plays a role in the final decision for the classification problem.

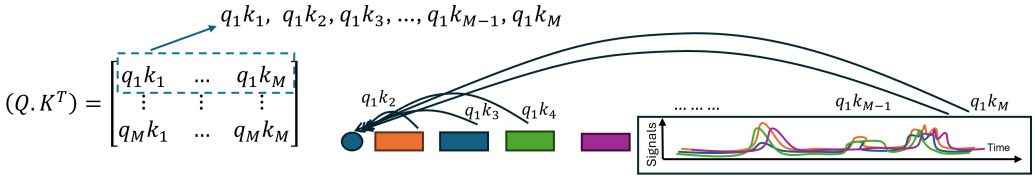


Figure 3: 1-dimentional representation of the inputs to the MHA layer.

The actual transformer output includes a residual path: $Y_{attn} = \text{LayerNorm}(X + \text{Attention}(X))$. However, since the CLS token is initialized randomly and carries no task-relevant information prior to attention, its pre-attention representation contributes minimally to the final output. The dominant information flow into the CLS position comes through the attention mechanism, making the attention weights a reliable proxy for feature importance in this single-layer architecture.

3.2 Training Procedure

The model is trained end to end using cross entropy loss. Input level dropout is essential in the designed network and used to simulate missing or corrupted channels. During training, we apply sample level dropout to both raw inputs and CNN derived features. For a randomly selected subset of samples, either the raw input or the CNN output is zeroed out. This regularization strategy is required since both the extracted features and raw signals sections could individually be used for the classification (i.e. there is redundancy). The sample drop out encourages robustness and prevents the model from over relying on redundant signals.

4 Results

The proposed algorithm is tested on two dataset cases. The first dataset is from UEA Time Series Classification Archive Bagnall et al. (2018) augmented by synthesized faulty classes. The second dataset is a real-world case in which data collected through a cloud service from field vehicles is used. In the second dataset, an actual faulty class exists in the population due to a component fault.

4.1 CASE STUDY 1: BODY MOTION CLASSIFICATION

As the first case study, the Racket Sports multivariate time series classification dataset from the aeon/UEA archive is used. Each sample is treated as an S signals–by–L motion trajectory (S: number of signals/channels from multiple sensors over L time steps). Class labels are originally string motion classes (e.g., racket strokes), which are later encoded to integers for training. To create "faulty" behaviors, synthetic samples are generated by cloning randomly selected normal trajectories and injecting localized high amplitude segments into $channel\ 1$ over two short time windows. Two classes of faults are generated namely $Faulty_1$ in which indices 18–23 are perturbed and $Faulty_2$ in which indices 25–30 are perturbed (shown in Figure 4). These modified copies are appended as two new classes, effectively augmenting the dataset with controlled, class specific anomalies. Figure 4 shows samples from the two synthetic faulty distributions. Along with the time trajectories on the right, the distribution of each class is plotted on the left compared with the original signal used to generate the synthetic faulty set. As one can see, the histograms cannot isolate $fault\ 1$ from $fault\ 2$ since the signal values are the same but the time instance that the fault happens is different. This indicates a simple statistical method cannot be used here to isolate the two fault classes.

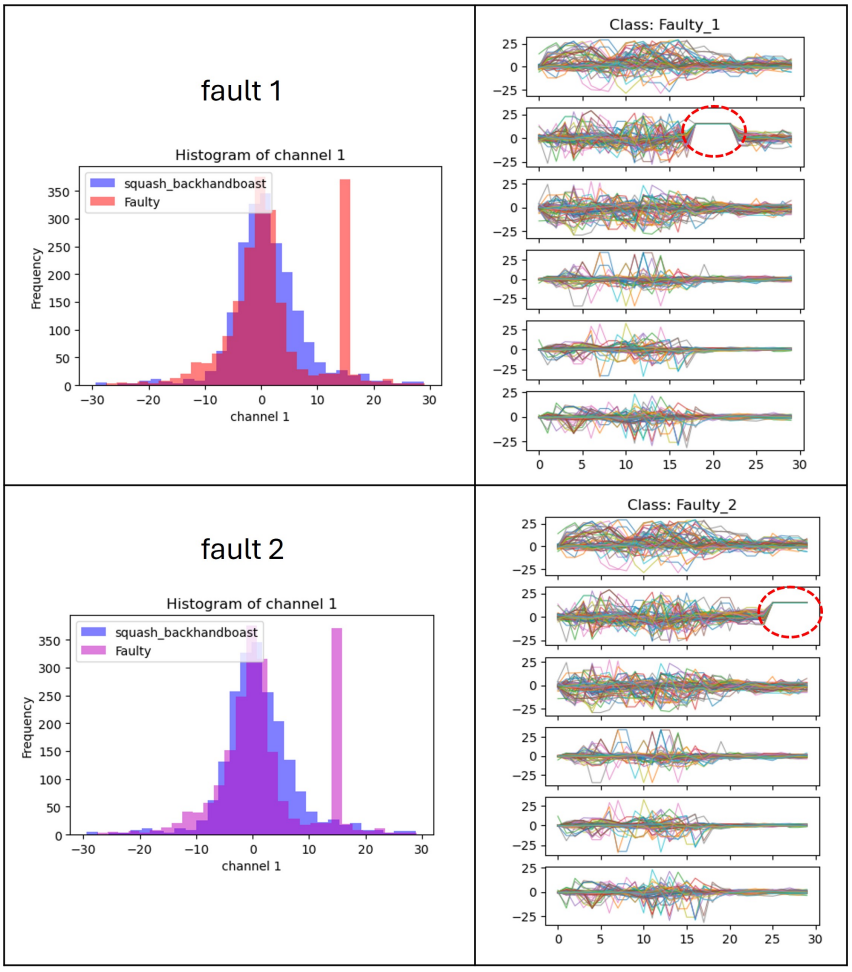


Figure 4: (left) distributions and (right) time series presentation of the two synthetic fault classes along with their original (healthy) data. The fault injection window is circled in red. Since both faults are generated by perturbing signal 1, they cannot be isolated if temporal relations are ignored (i.e. if time sequence is not considered).

The network is trained in multiple training sessions to study the robustness of the algorithm and in all cases, it correctly detects $channel_1$ as the most related signal for the classification problem

Table 1: Importance of each channel out of 59 synthetic faulty cases that are predicted correctly (from $Faulty_1$ class with 70 members). Channel 1 is always selected to be the most important channel in all cases.

Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
0%	100%	0%	0%	0%	0%

(Table 1). Also, results of the time window identification shown in Figure 5 indicate the network detects the most informative time windows (when the fault is injected) correctly based on the attention weights. In Figure 5-left, the fault is injected synthetically by disturbing channel 1 between time steps $[18, 23]$ while in Figure 5-right, the fault injection is applied to the same channel between time steps $[25, 30]$. In both cases, it can be observed that the calculated importance elevates during time instances that the fault is injected.

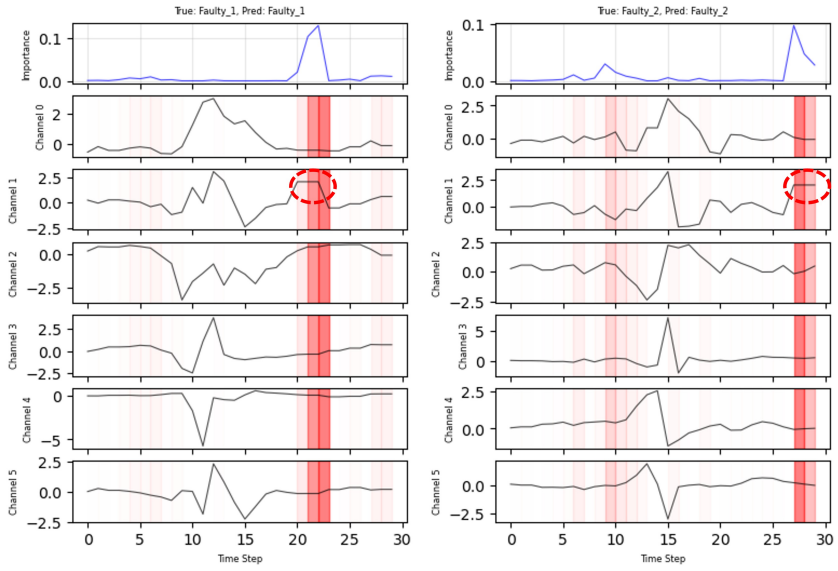


Figure 5: (Importance of time steps for for a sample from $faulty_1$ class (left) and $faulty_2$ class (right). The circled time steps indicate the algorithm correctly detect time steps where the fault happened (darker = more informative time step)

4.2 Case Study 2: Real World Fleet Vehicle Data

The attention zone detection method is applied to data collected from vehicles using cloud services. For each vehicle, a sequence of the last 10 samples is used. Figure 6 shows statistics of six signals (channels) and the comparison between distributions of the faulty and healthy population. As shown, there is overlap between faulty and healthy populations in all channels which makes the classification a challenging problem. However, the $pressure\ 2$ signal shows some level of separation between the two populations. This dataset is selected as a case study where the classification problem needs to narrowly separate the two populations and identify the important channel and time steps.

The proposed network in Figure 2 is applied to train a classifier predicting the health status of each vehicle. After the training, channel importance is calculated in two different training sessions. Figure 7 shows percentage of times each channel is selected as the most important signal in each run. As the figure shows, for both cases, channel 5 (the $pressure\ 2$ signal) is selected to be the most informative signal. Figure 8 shows the attention weights and the corresponding most informative time instances selected for two faulty vehicles. As the plots show, the most important time distance coincides with the lowest $pressure\ 2$ measured for each vehicle. This is a reasonable selection as the component fault is confirmed to interrupt the fluid transportation and control system and reduce the pressure in the corresponding pipeline.

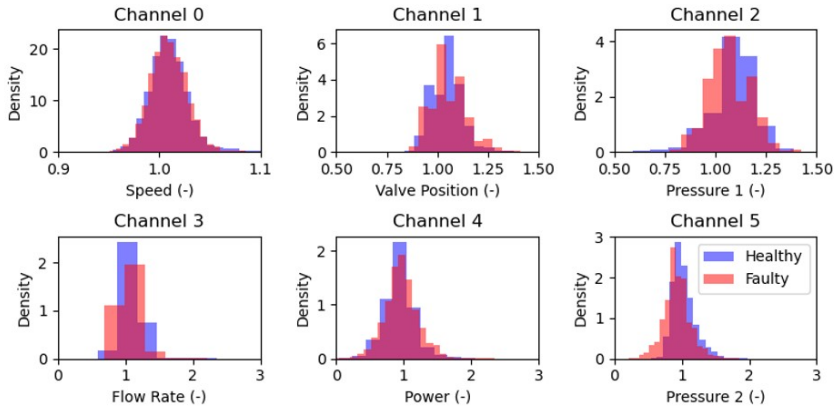


Figure 6: Statistical distribution of the 6 signals comparing healthy and faulty vehicles distributions. One can see that the $pressure\ 2$ shows some levels of separation as the faulty vehicles reported slightly lower pressure.

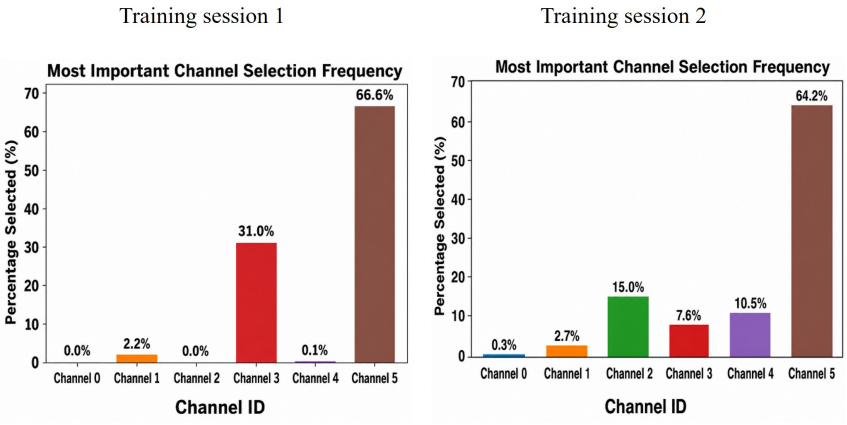


Figure 7: Importance detected for each channel from two different training sessions. The example trainings show robustness of the algorithm to pick the most important channel correctly in both runs.

5 Conclusion

This work shows that a deliberately structured hybrid convolutional–transformer can provide intrinsic interpretability, with attention weights directly localizing fault-related information at both the signal and time-step levels. Input- and sample-level dropout during training are required because the raw and CNN-derived representations are redundant; enforcing dropout improves robustness and prevents over-reliance on any single information source. Across two datasets, the method (i) correctly recovers injected synthetic faults by identifying discriminative time windows and the responsible signal that cannot be isolated from simple distributions, and (ii) in real fleet data, highlights measured pressure as the key signal and pinpoints low-pressure intervals associated with a vehcile component failure mode.

References

Sercan O. Arik and Tomas Pfister. TabNet: Attentive interpretable tabular learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-21)*, pp. 6679–6687, Virtual Conference, 2021.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The UEA multivariate time series classification archive,

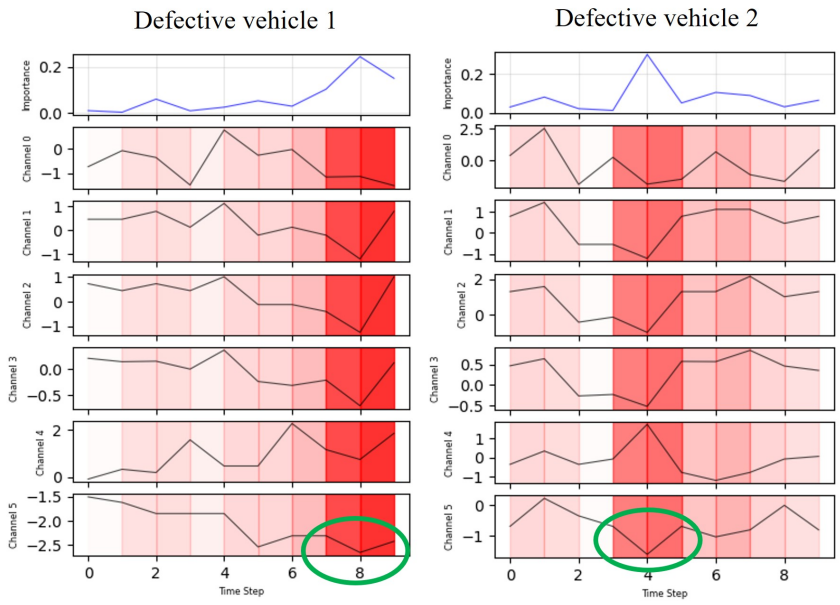


Figure 8: Attention time steps identified for two fault vehicles. For both vehicles, the attention is higher when $pressure$ 2 (channel 5) is lower.

2018. *arXiv preprint arXiv:1811.00075*, 2018. doi: 10.48550/arXiv.1811.00075. URL `https://doi.org/10.48550/arXiv.1811.00075`.

Xiwen Chen, Peijie Qiu, Wenhui Zhu, Huayu Li, Hao Wang, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. TimeMIL: Advancing multivariate time series classification via a time-aware multiple instance learning. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, Vienna, Austria, 2024.

Joseph Early, Gavin KC Cheung, Kurt Cutajar, Hanting Xie, Jas Kandola, and Niall Twomey. Inherently interpretable time series classification via multiple instance learning. In *The Twelfth International Conference on Learning Representations (ICLR)*, Vienna, Austria, 2024.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18932–18943, Virtual Conference, 2021.

Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pp. 2127–2136, Stockholm, Sweden, 2018.

Jaeseok Jang and Hyuk-Yoon Kwon. TAIL-MIL: Time-aware and instance-learnable multiple instance learning for multivariate time series anomaly detection. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 17582–17589, Philadelphia, PA, USA, 2025.

Chen Liu, Shibo He, Haoyu Liu, and Shizhong Li. TreeMIL: A multi-instance learning framework for time series anomaly detection with inexact supervision. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seoul, South Korea, 2024.