MoE-TransDLD: A Transformer-Driven Mixture of Experts for Cyber-Attack Detection in Power Systems

Luyu Wang[1], Biplab Sikdar[2], Kaifeng Zhang[3] and Ying Wang[4]

***Abstract*— The collaborative analysis of both cyber-layer and physical-layer data is crucial for improving detection accuracy and timeliness of cyber-attack. Cyber-layer features provide early indicators of attacks, while physical-layer features reflect the actual impact on the power system. To leverage this synergy, a cross-attention mechanism is introduced to generate cross-layer features to capture these cross-layer interactions. Furthermore, based on the traditional Mixture of Experts (MoE), a novel framework MoE-Transformer Dual Layers Detection (MoE-TransDLD) is proposed, which dynamically fuses multi-layer features to model cyber-physical dependencies. Specially, MoE-TransDLD assigns a dedicated expert to each layer, including a cyber-layer expert, a physical-layer expert, and a cross-layer expert, to more accurately model multi-layer data relationships in power systems. Notably, both the expert network and the gating network share a common Transformer architecture to extract global features, while maintaining corresponding independent feed-forward network (FFN), where each expert focuses on its respective domain and the gating network achieves adaptive and dynamic selection in decision making. The synthetic Texas 2000-bus model system is used as an experimental model and its physical-layer data and cyber-layer data are collected. The experimental results show that the MoE-TransDLD significantly outperforms the existing methods and achieves superior classification metrics and faster attack detection time.**

I. INTRODUCTION

As modern power systems increasingly depend on communication networks and automated control, they become vulnerable to cyber-attacks that can severely disrupt physical operations [1]–[3]. Attackers often exploit industrial protocols like DNP3 and TCP/IP to gain unauthorized access, manipulate data, or issue malicious control commands. Accordingly, detecting cyber-attacks in power systems has become a critical research challenge.

Traditional cyber-attack detection methods focus primarily on cyber-layer or physical-layer data. Cyber-layer approaches analyze anomalies in communication traffic, such as unusual packet rates, modified function codes, or unexpected source-destination pairs. For network protection in industrial control systems, intrusion detection systems (IDS) such as Snort, BRO, or Suricatta are increasingly being used [4]. They offer a pure cyber-centric approach that results in high false alarms. Meanwhile, physical-layer detection methods monitor electrical parameters, including voltage, power flow, and line status, to identify deviations from normal system behavior. Examples include fault detection [5] and intrusion detection using principal component analysis (PCA) [6]. However, relying on a single-layer approach has limitations, as sophisticated attacks can be designed to manipulate network traffic while mimicking normal physical conditions or vice versa.

Cyber-attacks often progress in multiple stages [7], [8], e.g., starting with a reconnaissance phase, executing intrusions and vulnerability exploitations, and culminating in actions targeting the physical system such as manipulating measurements and commands. The events that comprise these incidents and forensics about what occurred are not reflected using only coarse cyber-side features. For example, an intruder may take months in the reconnaissance phase, but during this period, none of the physical side features reflect any abnormality. By leveraging the fast response of cyber-layer data in tandem with the validating role of physical-layer measurements, it becomes feasible to anticipate attacks before they cause severe damage while maintaining accuracy in distinguishing genuine threats from benign operational variations.

However, cyber and physical systems differ greatly in sampling rate, semantics, and noise characteristics, making data fusion nontrivial. Direct fusion of raw features can lead to information dilution and loss of layer-specific advantages. Prior work has explored general-purpose fusion techniques such as co-training [9], multiple kernel learning [10], and subspace learning [11]. These methods, often referred to as direct fusion approaches, combine features or models from multiple sources to improve generalization. However, applying such methods directly to power system cyber-physical data may fail to exploit their structural heterogeneity. Notably, work by A. Sahu et al. [12] elaborated on such techniques for cyberattack detection but did not address dynamic expert selection or long-term temporal modeling.

To overcome these limitations, this paper proposes a dual-layer fusion framework based on a Mixture of Experts (MoE) architecture integrated with Transformer networks. The MoE mechanism dynamically assigns weights to expert models specializing in cyber, physical, or cross-layer features via a learnable gating network [13]–[15]. Unlike traditional fusion methods that statically merge inputs, MoE allows adaptive expert selection depending on data context. For instance, attention-weighted variants such as AW-MoE [16]

*This work is supported by State Grid Corporation of China Project "Research on self-organizing response architecture and methods of massive flexible resources to improve power grid regulation capacity"(5108-202218280A-2-285-XG).

[1]Luyu Wang, [3]Kaifeng Zhang and [4]Ying Wang are with the Key Laboratory of Measurement and Control of CSE, School of Automation, Southeast University, Nanjing 210096, China wangluyu@seu.edu.cn; kaifengzhang@seu.edu.cn; wyseu@seu.edu.cn

[2]Biplab Sikdar is with the Department of Electrical and Computer Engineering, National University of Singapore, Singaporebsikdar@nus.edu.sg

have shown success in capturing modality-specific knowledge. Incorporating Transformer layers further enhances the model's ability to capture long-range dependencies in time-series data, which is critical for identifying stealthy or delayed attacks.

Despite the growing interest in MoE for multi-source learning, its application to cyber-physical attack detection in power systems remains underexplored. We address this gap by proposing MoE-TransDLD, a novel framework that integrates cyber, physical, and cross-layer information for high-fidelity attack detection. The main contributions of this work are summarized as follows:

1. A cross-attention mechanism is introduced to extract cross-layer interactions, capturing how cyber attacks propagate into the physical layer.

2. A Mixture of Experts model is developed that fuses cyber-layer, physical-layer, and cross-layer data to achieve both early and accurate detection of cyber-attacks.

3. Each expert processes a specific data layer using a shared Transformer backbone for global context, but independent feedforward heads for domain-specific focus. The gating network also employs a Transformer to enable adaptive expert selection.

4. Using the ACTIVSg2000 system, we demonstrate that MoE-TransDLD outperforms single-layer and fixed-weight MoE baselines in detection accuracy and computational efficiency.

The remainder of this paper is organized as follows: Section II presents the methodology, including data preprocessing and model architecture. Section III provides experimental results and analysis. Section IV concludes the paper with key findings and future directions.

II. PROPOSED METHODOLOGY FOR MOE-TRANSDLD

The MoE method is a combining approach that uses individual learning techniques as experts who specialize in their particular subspace of input space [17]. The MoE method mainly has the following three components: (1) several intermediate experts, which are either regression or classifiers depending upon the given problem domain; (2) a gating function that partitions the input space into the number of subspaces using soft partition boundaries; (3) a probabilistic model to combine the experts and the gate. The proposed MoE-TransDLD enhances the traditional MoE framework by incorporating Transformer capabilities. Specifically, as described in the following subsections.

A. Problem Formulation

Let us represent the data $S = \{\mathcal{M}, \mathcal{D}\}$, $\mathcal{M} = (X_c^t, X_p^t, y^t) \mid t = 1, \ldots, T$, and $\mathcal{D} = \hat{y}^t \mid t = 1, \ldots, T$, where $X_c^t \in \mathbb{R}^{d_n}$ represents the characteristics of the cyber-layer at time t, $X_p^t \in \mathbb{R}^{d_p}$ represent the characteristics of the physical-layer, Y^t is the attack label indicating whether a cyber-attack is occurring, and $\hat{y}^t$ is predicted Y^t.

The goal of MoE-TransDLD is to learn a detection function $f : (X_c^t, X_p^t) \rightarrow \hat{y}^t$ that accurately predicts $\hat{y}^t$, minimizing the detection error:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \ell(y^t, \hat{y}^t) \tag{1}$$

where $\ell(\cdot, \cdot)$ is a loss function, such as binary cross-entropy. Thus, the problem is formulated as a supervised binary class classification problem.

To improve detection robustness, the cross-layer data interaction modeling is incorporated:

$$X_r^t = g(X_c^t, X_p^t) \tag{2}$$

where $g(\cdot, \cdot)$ captures bidirectional dependencies between cyber-layer commands and physical-layer responses.

B. Proposed Methodology

MoE-TransDLD extends the traditional MoE by incorporating Transforme-based experts—dedicated to cyber-layer, physical-layer, and cross-layer data. A shared Transformer architecture provides unified feature extraction, while each expert retains an independent FFN for specialized learning. The gating function also shares the Transformer architecture. Specifically, the architecture is shown in Fig. 1.

1) Assignment of Experts to Each Layer: MoE-TransDLD adopts the MoE framework consisting of three expert models, each based on a shared **Transformer** architecture but focusing on different aspects of the dual-layer data:

- **Cyber-layer Expert**: A Transformer-based model that processes cyber-layer data. Specifically as follows:
  - *Protocol Features*: Function codes, control commands (e.g., DNP3, Modbus operations).
  - *Traffic Pattern Metrics*: Packet rate, inter-arrival times, flow duration, number of active connections.
  - *Anomaly Indicators*: Unexpected function code, snot, or unusual source-destination pairs.

  This expert leverages multi-head self-attention to capture long-range dependencies in communication flows, enabling early recognition of malicious commands or abnormal traffic bursts.
- **Physical-layer Expert**: A Transformer-based model that processes physical-layer data. Specifically as follows:
  - *State Variables*: Real and reactive power flows, bus voltages, line currents, breaker statuses.
  - *Dynamic Behaviors*: Rate of change of frequency (ROCOF), load variations, generator dispatch levels.
  - *Fault/Anomaly Indicators*: Sudden voltage sags, line overloads, abnormal frequency drifts.

  By using attention to learn temporal trends and correlations among power system states, this expert identifies incipient physical disruptions or anomalies caused by cyber-attacks.
- **Cross-layer Interaction Expert**: A Transformer-based model utilizing self-attention to explicitly model the interplay between cyber-layer commands and physical-layer responses.

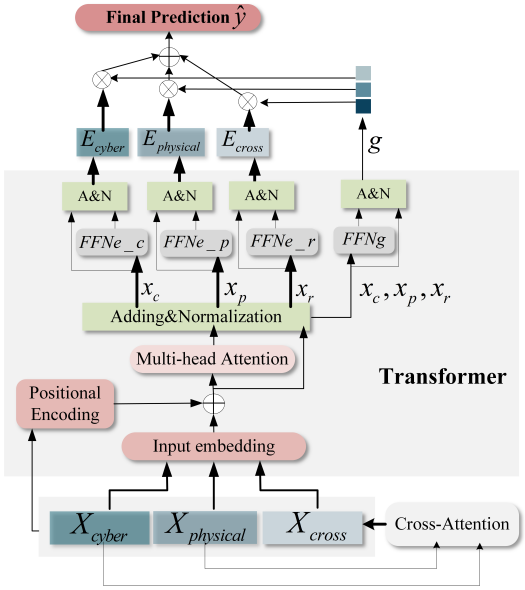

Fig. 1: Overview of the proposed MoE-TransDLD

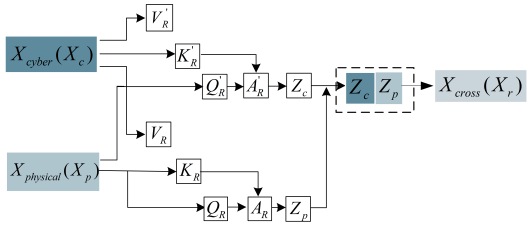

Fig. 2: Cross-layer feature extraction based on cross-attention

2) Cross-Layer Feature Extraction Based On Cross-Attention: To capture cross-layer dependencies, cross-attention between cyber-layer and physical-layer features is applied, as shown in Fig.2. First, the attention of the physical-layer on the cyber-layer is calculated.

$$Q_R = X_c^t W_Q^R, \quad K_R = X_p^t W_K^R, \quad V_R = X_p^t W_V^R \tag{3}$$

$$A_R = \text{softmax}\left(\frac{Q_R K_R^T}{\sqrt{d_k}}\right) \tag{4}$$

$$Z_c = A_R V_R \tag{5}$$

where the cyber-layer X_c is used as the query, the physical-layer X_p is used as the key and value, and the attention of the physical-layer on the cyber-layer; W_Q^R, W_Q^R, and W_Q^R are trainable parameters for cross-attention computation; the generated Z_c represents the cross-features enhanced by the cyber-layer.

Then, the attention of the physical-layer on the cyber-layer is calculated.

$$Q_R{}' = X_p^t W_Q^R, \quad K_R{}' = X_c^t W_K^R, \quad V_R{}' = X_c^t W_V^R \tag{6}$$

$$A_R{}' = \text{softmax}\left(\frac{Q_R{}' K_R^{'T}}{\sqrt{d_k}}\right) \tag{7}$$

$$Z_P = A_R{}' V_R{}' \tag{8}$$

where the physical-layer X_p is used as the query, the physical-layer X_c is used as the key and value, and the attention of the physical-layer on the cyber-layer; the generated X_r represents the cross-features enhanced by the physical-layer.

Finally, the cross-layer feature fusion is performed.

$$X_r = Concat(Z_c, Z_p) W_F \tag{9}$$

where W_F is a trainable linear transformation parameter that adjusts the feature dimension.

3) Expert Network and Gating Network Based On a Shared Transformer: The Transformer-Gating mechanism is directly integrated into the Transformer-based expert network, which extends the functionality from "only assigning expert weights" in traditional MoE to "fusing multi-source information and adaptively selecting experts. " First, the input data from the cyber-layer, physical-layer, and cross-layer data is concatenated:

$$X = [X_c^t, X_p^t, X_r^t] W_E + P \tag{10}$$

where W_E is the embedding matrix, and P is the positional encoding to retain temporal order. The Transformer applies multi-head self-attention (MHSA) to capture dependencies:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \tag{11}$$

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \tag{12}$$

$$H = AV \tag{13}$$

where W_Q, W_K, and W_V are trainable parameters, and A is the attention matrix learning global feature dependencies. For each expert, a FFN is used to compute expert-specific features:

$$E_i = \text{ReLU}(W_{E1} H + b_{E1}) W_{E2} + b_{E2} \tag{14}$$

where W_{E1} and W_{E2} are expert feature extraction parameters. The Transformer-Gating network dynamically computes expert selection weights:

$$\alpha_i^t = \text{softmax}(W_{G1} H + b_{G1}) \tag{15}$$

where W_{G1} is the gating function mapping input into expert weights. Using the expert features and dynamically computed weights, the final anomaly detection output is obtained:

$$\hat{y}^t = \sum_{i=1}^{3} \alpha_i^t E_i \tag{16}$$

The gating network of MoE-TransDLD is no longer just a simple MLP to "soft-partition inputs", but a Transformer-driven deep gating mechanism that can dynamically deploy multi-layer experts, and the shared Transformer architecture not only reduces duplicated computations, but also allows the experts and gates to maintain the same global temporal awareness of the data for better accuracy and adaptability in network attack detection.

III. EXPERIMENTAL SETUP AND RESULTS ANALYSIS

The detection performance and the detection time of the proposed MoE-TransDLD are now evaluated, following the elaboration of the dataset generation and experiment setups.

A. Dataset Generation

This study adopts the synthetic Texas 2000-bus test system [18], a large-scale benchmark model representing the electrical infrastructure of the Texas power grid, to simulate cyber-physical attack scenarios. Both physical-layer and cyber-layer datasets are constructed based on this model.

1) Physical-Layer Data Construction: The physical-layer dataset is generated through iterative power flow simulations using the *case_ACTIVSg2000* model in MATPOWER. A total of 2000 labeled samples are produced, comprising 1000 normal and 1000 attack samples. The data is partitioned into training (80%) and testing (20%) sets.

Normal Samples. Normal conditions reflect both steady-state and contingency scenarios that occur without cyber intrusion. Three types of system modifications are implemented to synthesize operational diversity:

- *Load variation:* Adjustments are made to the active and reactive power demand on selected buses to simulate varying load profiles.
- *Generator dispatch changes:* The power outputs of specific generators are modified to emulate different generation strategies.
- *Planned outages:* Selected transmission lines are temporarily disabled to represent maintenance or fault conditions.

Among the 1000 normal samples, approximately 50% are generated by varying load and generation profiles, while the remaining 50% involve scheduled system disconnections such as branch outages due to maintenance. Specifically, 480 samples are based on load and generation fluctuations, and 520 samples are from scheduled branch disconnection scenarios. This ratio is selected to ensure that the model learns to distinguish between benign and malicious topological changes.

Attack Samples. Cyber-attacks are modeled as deliberate manipulations of the grid topology through unauthorized disconnection of critical transmission lines. These scenarios are implemented by modifying branch connection statuses to simulate stealthy intrusion events. Though structurally similar to fault-induced disconnections, attack cases are distinguished by their malicious intent and target selection strategy.

Following each simulation, system states including voltage magnitudes, phase angles, power injections, and line flows are extracted from the updated bus, branch, and generator matrices. These results form the basis of the physical-layer time-series inputs for the proposed detection model.

2) Cyber-Layer Data Generation: The cyber-layer data generation is based by transmitting the previously generated physical-layer data using the DNP3 protocol. This process involves simulating DNP3 communication between a master station and outstations using the Python 'dnp3' package. During data transmission, Wireshark is employed to capture and log network communication. The captured packets are stored in JSON format. The method for parsing JSON files and extracting relevant cyber-layer features can be found in [12]. The recorded network traffic serves as cyber-layer data.

Normal Data Collection: The normal cyber-layer data is collected by simulating routine DNP3 communication between the master and outstations, where:

- The master transmits normal physical-layer data, including power flow, voltage, and branch status.
- The DNP3 function codes reflect standard operational commands, such as reading and writing data.
- Wireshark captures network traffic during normal system operations.
- The captured packets are stored in JSON format.

Attacked Data Collection: To simulate a cyber-attack scenario:

- The attacker sniffs normal cyber-layer traffic, identifying unusual patterns such as frequent read requests or increased data packet rates.
- The attack is triggered by modifying the DNP3 function code to 5, instructing a branch disconnection at the target outstation.
- The master transmits attacked physical-layer data, reflecting the induced fault.
- Wireshark captures the post-attack cyber-layer data, including:
  - The altered DNP3 function code.
  - The transmission of branch disconnection status from the outstation in the next integer second.
  - The response of other outstations, reflecting overloads and power flow redistributions due to the attack.
- The captured packets are stored in JSON format.

3) Data Preprocessing: To ensure effective learning, the cyber-layer features and physical-layer features are first normalized to ensure numerical stability. Second, due to the substantial sampling frequency gap between the cyber-layer and physical-layer, a simple yet effective approach to align their time series is repetition alignment. Specifically, the cyber-layer data is treated as the primary high-frequency timeline and repeated each physical-layer data point until the next sampling point arrives. Formally, let $\{(t_p, d_p)\}$ be the physical-layer samples, where t_p indicates the p-th sampling time, and d_p is the corresponding measurement (e.g., voltage or power flow). For any network-layer timestamp t_n satisfying $t_p \leq t_n < t_{p+1}$, the physical-layer value d_p is assigned. Thus, until the physical-layer sample is updated at t_{p+1}, the system reuses d_p for each high-frequency timestamp t_n.

B. Training the MoE-TransDLD Model

The MoE-TransDLD framework is implemented in PyTorch. The training process is optimized using the AdamW optimizer [19] with a learning rate of 5e-4 and a weight

decay of 1e-4. To gradually adjust the learning rate during training, a Cosine Annealing scheduler is employed with T_max = 20. The model is trained with a batch size of 16 for 50 epochs. The cross-entropy loss function is used to optimize the classification task. During training of the MoE-TransDLD model, in the last epoch, accuracy reaches 99.52%.

C. Evaluation and Analysis

The comparative analysis of the proposed MoE-TransDLD framework is presented. Multiple comparative experiments are performed: (1) evaluating Transformer-based models on both single-layer and dual-layer datasets to highlight the advantages of collaborative dual-layer data fusion, and (2) comparing MoE implementations with different deep learning architectures as expert networks to demonstrate the superiority of incorporating Transformers into the MoE framework. The trained MoE-TransDLD model is evaluated using:

- Accuracy (ACC): The proportion of correctly classified instances (both attack and normal) among all samples.
- Precision (PR): The proportion of predicted attacks that are indeed actual attacks.
- Recall (RE): The proportion of actual attacks that are correctly identified as attacks.
- F1-Score (F1): The harmonic mean of Precision and Recall, reflecting the overall balance of these two metrics.
- Detection Time (DT): The average time required to detect an ongoing attack from the moment it appears.
- Confusion Matrix: A tabular layout showing the distribution of predicted vs. actual classes (TP, TN, FP, FN)

Figure 3 presents the confusion matrices for different methods. Figure 4 compares the detection time across different methods, where 0s represents the moment when an attack triggers a circuit disconnection.

1) Single-Layer Transformers vs. MoE-TransDLD: Figure 5 compares MoE-TransDLD with the following three baselines based on single-layer datasets. All baselines adopt a standard Transformer architecture trained on the specified input. It can be seen that the MoE-TransDLD achieves the best detection results in terms of four metrics: ACC, PR, RE, and F1.

- Cyber-Only (Transformer).
- Physical-Only (Transformer).
- Cyber-Physical Concatenation (Transformer).

Besides, as shown in Fig. 3, due to the lack of physical-layer verification, the cyber-only model has a high false positive rate and often misclassifies normal samples as attacks. On the other hand, due to the lack of cyber-layer support, the physical-only model is difficult to distinguish between normal system failures in non-attack scenarios and actual attacks, leading to challenges in accurate classification.

The results in Fig.4 show that physical-only can only detect an attack after the physical disconnection occurs, causing a noticeable delay in attack identification. In contrast, all methods that incorporate cyber-layer data successfully

Actual cases	Classified cases	
	Attacked	Normal
Attacked	95%	5%
Normal	21.7%	79.3%

(a) Cyber-Only (Trans)

Actual cases	Classified cases	
	Attacked	Normal
Attacked	97.5%	2.5%
Normal	19.3%	81.7%

(b) Physical-Only (Trans)

Actual cases	Classified cases	
	Attacked	Normal
Attacked	96.3%	3.7%
Normal	7.5%	92.5%

(c) Cyber-Physical (Trans)

Actual cases	Classified cases	
	Attacked	Normal
Attacked	99.5%	0.5%
Normal	0.8%	99.2%

(d) MoE-TransDLD

Actual cases	Classified cases	
	Attacked	Normal
Attacked	97.5%	2.5%
Normal	3.8%	96.2%

(e) MoE-MLPDLD

Actual cases	Classified cases	
	Attacked	Normal
Attacked	96.9%	3.1%
Normal	1.1%	98.9%

(f) MoE-LSTMDLD

Actual cases	Classified cases	
	Attacked	Normal
Attacked	97.9%	2.1%
Normal	97.7%	2.3%

(g) MoE-CNNDLD

Fig. 3: Confusion matrices (in percentages) for the seven evaluated methods.

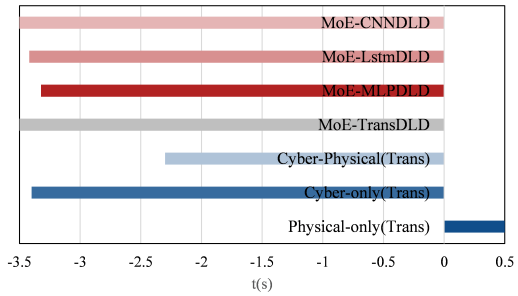

Fig. 4: Comparison of detection time

achieve preemptive detection, demonstrating the critical role of cyber-layer in timely attack identification.

2) MoE-Based Dual-Layer Methods: MLP, LSTM, CNN vs. Transformer: Next, MoE variants based on other deep learning algorithms are compared, with each method integrating both network and physical data. As shown in Fig. 6, MoE-TransDLD consistently outperforms other MoE variants, highlighting the advantages of Transformer-based experts in capturing long-range temporal dependencies and cross-layer interactions.

- MoE-MLPDLD: Each expert is an MLP, capturing only low-level feature interactions without explicit temporal modeling.
- MoE-LSTMDLD: Each expert is LSTM-based, capable of modeling short-term temporal dependencies but limited in capturing long-range dependencies and cross-layer interactions. The gating network is also LSTM-based, dynamically assigning expert weights based on recent historical trends rather than a global attention mechanism.

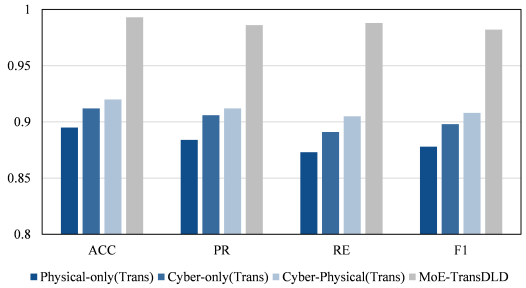

Fig. 5: Comparison of MoE-TransDLD and other dataset base on Transformers.

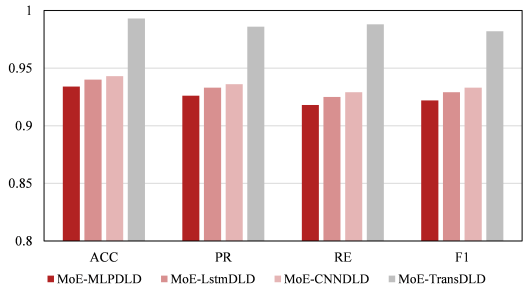

Fig. 6: Comparison of MoE-TransDLD and MoE variants.

- MoE-CNNDLD: Each expert is a CNN, effective for spatial feature extraction but lacking explicit temporal modeling capabilities, which may limit its performance in sequential data processing.

IV. Conclusion

This paper presented MoE-TransDLD, a Transformer-driven MoE framework that integrates cyber-layer and physical-layer data. By leveraging early attack signals in the cyber-layer, system validation in the physical-layer, and cross-layer interactions via a cross-attention mechanism, MoE-TransDLD mitigates the limitations of single-layer approaches. Each expert network and the gating network share a common Transformer architecture yet maintain independent FFN, achieving consistent global feature extraction and specialized learning, which can save computational resources while improving the accuracy of gating decisions.

Extensive experiments on the synthetic Texas 2000-bus power system demonstrate that MoE-TransDLD outperforms existing methods, attaining higher accuracy, reduced false alarms, and faster detection. The Transformer-based gating mechanism adaptively allocates expert weights, enabling robust performance across diverse attack scenarios. Future work may explore sparse gating or load-balancing strategies to optimize computational overhead.

References

[1] S. Xu, Y. Xia, and H.-L. Shen, "Analysis of malware-induced cyber attacks in cyber-physical power systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 12, pp. 3482–3486, 2020.

[2] N. Saxena, L. Xiong, V. Chukwuka, and S. Grijalva, "Impact evaluation of malicious control commands in cyber-physical smart grids," *IEEE Transactions on Sustainable Computing*, vol. 6, no. 2, pp. 208–220, 2018.

[3] C.-W. Ten, K. Yamashita, Z. Yang, A. V. Vasilakos, and A. Ginter, "Impact assessment of hypothesized cyberattacks on interconnected bulk power systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4405–4425, 2017.

[4] Y. Hu, A. Yang, H. Li, Y. Sun, and L. Sun, "A survey of intrusion detection on industrial control systems," *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, p. 1550147718794615, 2018.

[5] M. Kordestani and M. Saif, "Data fusion for fault diagnosis in smart grid power systems," in *2017 IEEE 30th Canadian conference on electrical and computer engineering (CCECE)*. IEEE, 2017, pp. 1–6.

[6] J. Valenzuela, J. Wang, and N. Bissinger, "Real-time intrusion detection in power system operations," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1052–1062, 2012.

[7] B. Zhu, A. Joseph, and S. Sastry, "A taxonomy of cyber attacks on scada systems," in *2011 International conference on internet of things and 4th international conference on cyber, physical and social computing*. IEEE, 2011, pp. 380–388.

[8] M. M. Hassan, S. Huda, S. Sharmeen, J. Abawajy, and G. Fortino, "An adaptive trust boundary protection for iiot networks using deep-learning feature-extraction-based semisupervised model," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2860–2870, 2020.

[9] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 92–100.

[10] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[11] N. Chen, J. Zhu, and E. Xing, "Predictive subspace learning for multi-view data: a large margin approach," *Advances in neural information processing systems*, vol. 23, 2010.

[12] A. Sahu, Z. Mao, P. Wlazlo, H. Huang, K. Davis, A. Goulart, and S. Zonouz, "Multi-source multi-domain data fusion for cyberattack detection in power systems," *IEEE Access*, vol. 9, pp. 119 118–119 138, 2021.

[13] S. Masoudnia and R. Ebrahimpour, "Mixture of experts: a literature survey," *Artificial Intelligence Review*, vol. 42, pp. 275–293, 2014.

[14] A. Omer, S. S. Rathore, and S. Kumar, "Me-sfp: A mixture-of-experts-based approach for software fault prediction," *IEEE Transactions on Reliability*, 2023.

[15] J. Li, J. Jin, H. Chen, and W. Huang, "Attention mechanism with adaptive normalization and mixture of experts for estimating battery capacity degradation," in *2024 IEEE 17th International Conference on Signal Processing (ICSP)*. IEEE, 2024, pp. 204–208.

[16] J. Gong, Z. Chen, C. Ma, Z. Xiao, H. Wang, G. Tang, L. Liu, S. Xu, B. Long, and Y. Jiang, "Attention weighted mixture of experts with contrastive learning for personalized ranking in e-commerce," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 3222–3234.

[17] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE transactions on neural networks and learning systems*, vol. 23, no. 8, pp. 1177–1193, 2012.

[18] P. Wlazlo, K. Price, C. Veloz, A. Sahu, H. Huang, A. Goulart, K. Davis, and S. Zounouz, "A cyber topology model for the texas 2000 synthetic electric power grid," in *2019 Principles, Systems and Applications of IP Telecommunications (IPTComm)*. IEEE, 2019, pp. 1–8.

[19] I. Bello, B. Zoph, V. Vasudevan, and Q. V. Le, "Neural optimizer search with reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 459–468.