

# Embedded Partial Discharge Classification System Using Transformer Neural Networks on Raspberry Pi

Theerayod Wiangtong<sup>1†</sup>, Chanin Boonlaksananusorn<sup>2</sup>, Aung Ye Thway<sup>1</sup>,  
Siwakorn Jeenmuang<sup>1</sup>, and Norasage Pattanadech<sup>1</sup>, Non-members

## ABSTRACT

Partial Discharge (PD) represents a critical indicator of insulation deterioration in high-voltage equipment. This paper presents an embedded PD classification system using a Raspberry Pi platform with transformer-based neural network architecture. The system classifies four PD types, namely corona, surface, internal, and noise using phase-encoded signal data based on IEC 60270 standards. PRPD patterns are converted to sequential vectors through phase encoding; each discharge pulse is mapped to a discretized phase index with its associated charge value. The Raspberry Pi platform integrates an SSD1306 OLED display (128×64 pixels) and five LED indicators for real-time visual feedback. Experimental results demonstrate that the transformer architecture achieves 94.8% validation accuracy compared to CNN baselines at 88.3%. The system enables portable, low-cost condition monitoring without expensive laboratory equipment, demonstrating the viability of embedded artificial intelligence for industrial diagnostic applications.

**Keywords:** Partial Discharge, Transformer Neural Networks, Embedded Systems, Condition Monitoring, Edge Computing, Raspberry Pi

## 1. INTRODUCTION

High-voltage equipment in power systems faces continuous operational stress, and insulation degradation poses significant safety and reliability risks. Partial discharges represent early warning signs of insulation failure, occurring when localized electrical stress concentrations develop in insulation defects such as voids, cracks, or contamination. Detection and classification

of PD types enable predictive maintenance that reduce equipment failures and costs. Phased resolved partial discharge (PRPD) is already a visually classified pattern, and therefore, it can be detected by various classification algorithms. [1-5]

Traditional PD measurement systems require specialized laboratory equipment such as the High voltage transformer, coupling capacitor, Partial Discharge model and Measuring Device such as Omicron MPD 600, limiting their deployment in field monitoring applications. The Raspberry Pi 4B combines sufficient processing power with low cost, making it suitable for distributed sensor networks in industrial environments. This study develops a low-cost, portable partial discharge classification system that integrates three key elements: (1) a Transformer neural network architecture for sequence-based PD signal classification, (2) deployment on resource-constrained Raspberry Pi hardware, and (3) real-time visual feedback through OLED display and color-coded LED indicators.

The remainder of this paper is organized as follows. Section 2 reviews relevant theories. Section 3 neural network architectures. Section 4 describes the system design and implementation methodology. Section 5 presents experimental results with comparative analysis, and section 6 conclude with future work directions.

## 2. BACKGROUND AND THEORY

### 2.1 Partial Discharge Mechanisms

According to IEC 60270 international standards, partial discharges are defined as localized electrical discharges that partially bridge insulation between electrodes [6]. These discharges typically considered as pulses with durations much less than 1 microsecond according to IEC 60270 standard. These are the result of localized electrical stress concentrations exceeding the inception threshold.

Three primary PD types are distinguished by their physical location and mechanism. Corona Discharge occurs in gaseous media surrounding high-voltage conductors when local electric field intensity exceeds air dielectric strength. This discharge type typically arises around sharp points or edges where field concentration is highest [7-10]. Surface Discharge develops on contaminated or moist insulation surfaces where conductive films facilitate partial ionization. Internal Discharge manifests

Manuscript received on December 28, 2025; revised on February 20, 2026; accepted on March 10, 2026. This paper was recommended by Associate Editor Chawasak Rakpenthai.

<sup>1</sup>The authors are with Electrical Engineering Dept., School of Engineering, King Mongkut's Institute of Technology Ladkrabang Bangkok, Thailand.

<sup>2</sup>The author is with Mechatronics Engineering Dept., School of Engineering, King Mongkut's Institute of Technology Ladkrabang Bangkok, Thailand.

<sup>†</sup>Corresponding author: theerayod.wi@kmitl.ac.th

©2026 Author(s). This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License. To view a copy of this license visit: <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

Digital Object Identifier: 10.37936/ecti-ec.2026242.262835

The fundamental relationship governing field-voltage behaviour is:

where  $E$  is electric field intensity,  $V$  is applied voltage, and  $d$  is separation distance. In non-uniform fields, local field magnification can significantly exceed critical field stress, triggering the partial discharge.

The IEC 60270 standard specifies PD measurement procedures using a test circuit comprising a test transformer, coupling capacitor, coupling device, and measuring system [11]. Phase Resolved Partial Discharge (PRPD) patterns are derived by measuring charge-phase relationships across power cycles. The fundamental equation relates current integration to apparent charge:

The integration period is typically one microsecond. By recording apparent charge and phase for every discharge pulse over multiple power cycles using instruments such as the MPD 600 PD measuring unit from Omicron, a characteristic PRPD pattern emerges for each PD type. The measured data is then exported to comma-separated value (CSV) format containing time, charge, and phase information.

PD data is acquired from simulation models at King Mongkut's Institute of Technology (KMITL) HV Laboratory using an MPD 600 PD measuring unit according to IEC 60270 procedures as Fig. 1. Three simulation models represent corona discharge (plane-to-plane electrodes with median gap), surface discharge (two electrodes sandwiched between surfaces), and internal discharge (two plane electrodes with void region) as shown in Fig. 2 [12]. All tests are conducted at approximately 6 kV applied voltage.

Because the raw PRPD data vary markedly across discharge types, feature extraction must reduce this variability while preserving class-specific structure. The raw PD data exhibits significant variability in representation:

- [illegible]



Figure 1: Data Preprocessing. The diagram illustrates the workflow from raw data to CNN input. On the left, a heatmap of a battery cell is shown with a color scale from 0.0 to 0.1. A green line graph is overlaid on the heatmap. Below the heatmap, two stacked bars represent 'Charge x 1200' (purple) and 'Phase x 1200' (green). A bracket below these bars is labeled 'Standardized to Fixed Length per cycle'. On the right, the 'Number of Discharge per Cycle' is listed for Corona (~80), Surface (~800), and Internal (~800). Below this, the 'Input to CNN' is shown as a table with 'Charge' and 'Phase' rows, each with a 'Zero Padding' column. A pink arrow points from the heatmap to the 'Surface' row, and another pink arrow points from the 'Charge x 1200' and 'Phase x 1200' bars to the 'Input to CNN' table.

- internal discharge generates up to 800 events per cycle

To standardize this variability, the preprocessing algorithm creates a fixed-size array of 1200 elements suitable for all discharge types, padding with zeros where necessary. This generates a two-dimensional array of shape [2, 1200], where dimension 1 represents phase and charge values, and dimension 2 represents the 1000 data points as illustrated in Fig. 3.

The conversion maps phase values in the range 0–0.999 to discrete indices with corresponding charge values, yielding a one-dimensional representation of the PRPD pattern.

For transformer processing with 50 Hz power cycles, data is accumulated over 50 cycles to create 1-second data batches. The resulting data shape becomes [2, 60,000] for raw transformer input, which is subsequently processed through the embedding and attention mechanisms.

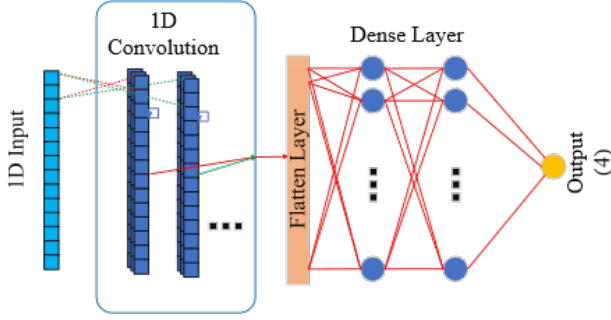


Fig. 4: One Dimensional Convolutional Neural Network.

### 3. NEURAL NETWORK ARCHITECTURES

Artificial Neural Networks provide nonlinear transformation through layered neuron structures. Classical ANNs establish baseline performance but lack spatial/temporal feature learning required for complex pattern recognition.

#### 3.1 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a deep learning architecture developed to autonomously learn hierarchical feature representations from input data using small, trainable filters [13]. This model is widely utilized for tasks such as image recognition and classification due to its ability to capture spatial dependencies effectively.

Phase-resolved data are typically complex and non-linear; therefore, a one-dimensional CNN (1D-CNN) is commonly employed to extract informative patterns from the sequential input due to minimal computational complexity as illustrated in Fig. 4 [14]. CNNs may be formulated in one-, two-, or three-dimensional forms depending on the structure and dimensionality of the data. They are generally composed of multiple convolutional and pooling layers, after which the learned feature maps are flattened and passed to fully connected (dense) layers. As the network depth increases, it can learn progressively more abstract and discriminative features. The rectified linear unit (ReLU) is typically used as the activation function in convolutional layers, whereas a SoftMax activation is applied at the output layer to facilitate multiclass classification.

#### 3.2 Transformer Neural Networks

The Transformer is a neural network architecture that relies on a multi-head attention mechanism to map input data into high-dimensional vector representations [15]. It incorporates positional encoding to preserve information about the order of elements within a sequence. This architecture is widely used in natural language processing tasks and forms the foundation of models such as GPT.

The input sequence is fed to the embedding layer and then positionally encoded. Embedding layer uses alternating sine and cosine functions of different frequencies,

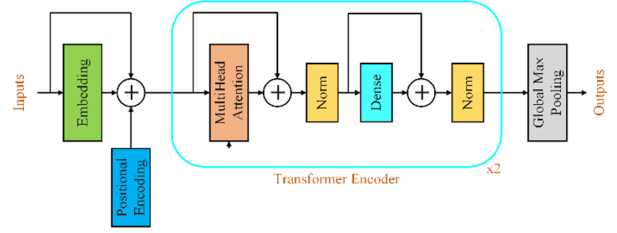


Fig. 5: Transformer Encoders.

to provide the input token with unique vector with lower dimension. Positional encoding is important since the attention mechanism itself does not capture positional structure of the sequence. Residual connections are applied to both the attention and feed-forward layers, and each is followed by layer normalization as presented in Fig. 5.

The multi-head attention mechanism enables the model to assign importance weights to different sequence positions based on their relevance to the classification task:

$$\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V \quad (3)$$

where  $Q, K, V$  are query, key, and value vectors respectively. The attention mechanism identifies whether data element at specific position should pay attention to that of other positions.

### 4. SYSTEM DESIGN AND IMPLEMENTATION

#### 4.1 System Architecture and Data Flow

Measured PRPD data is exported to CSV format containing time, charge, and phase information. From this raw data, time values are removed, leaving phase and charge as the critical features.

The phase encoding algorithm operates as follows:

1. Initialize a 2D array of shape  $[2, 1200]$ .
2. Discretize phase values into 1200 sections (0.001 increments from 0 to 0.999).
3. For each discharge pulse, identify the nearest phase index using the equation below.

$$\text{index} = \underset{i \in \{1, 2, \dots, 1000\}}{\text{arg min}} |\phi_{\text{index}} - \phi_i| \quad (4)$$

4. Store the corresponding charge value in that phase index.
5. Pad with Zeros for the remaining positions.

The process of positional encoding for PRPD data is presented in Fig. 6.

This converts phase-charge relationships into sequential 1D vectors representing the physical discharge distribution across one complete AC cycle.

#### 4.2 Data Compression for Edge Computing

Modern IoT and smart power systems generate messages or real-time data that transmit either raw information or pre-processed information to edge computers

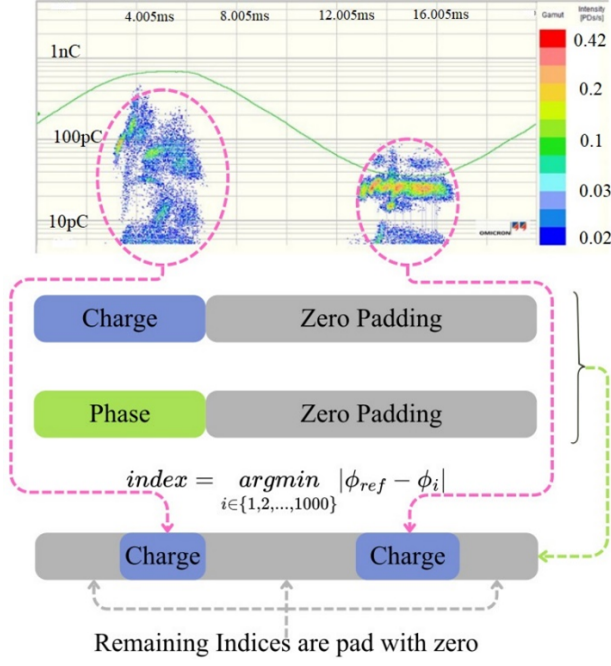


Fig. 6: Positional Encoding for PRPD Data.

or cloud servers. Traditional methods require significant computational resources and time. Therefore, data compression plays a crucial role in minimizing bandwidth usage, and lowering latency.

In edge computing, where processing occurs close to the data source, compression has a significant impact on both performance and energy consumption. In the case of Partial Discharge Classification systems, the data consists of phase and charge values. Without using any traditional lossless compression techniques, these two arrays can be directly compressed into a single array by using a positional encoding method as shown in Fig. 7. Phase Encoding method encodes phase positions into specific locations within the array that correspond to their respective values.

We can achieve at least a 50% compression rate without affecting the performance of the classification algorithms. The method achieves at least 50% compression without measurable loss of classification performance, so the one-dimensional data can be transmitted efficiently as discussed in section 4.1. After compression, the system can transmit the one-dimensional data efficiently.

During decompression, the system reconstructs the original information by finding index and masking the input data, essentially splitting the combined array back into two separate arrays using the following set of equations.

The data received from the edge device is:

$$x = [x_1, x_2, \dots, x_{1200}] \quad (5)$$

The index of the presented charge value can be calculated as follows.

$$index = \{i | (x_i \neq 1)\} \quad (6)$$

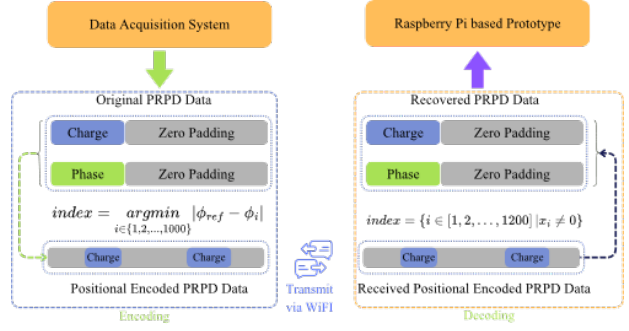


Fig. 7: Data Compression and Recovery.

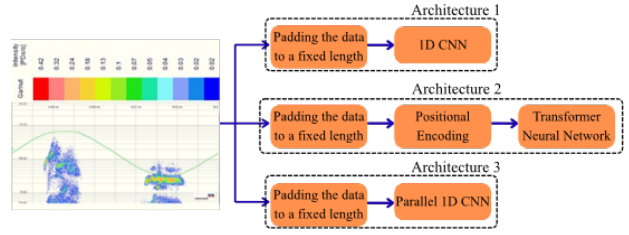


Fig. 8: Overview of the three methods.

The reference phase value is stored in look up table for faster processing and can be calculated as,

$$LUT, L_i = \frac{i}{1200}, i = \{1, 2, \dots, 1200\} \quad (7)$$

Finally, the phase array can be recovered using masking.

$$M_i = \begin{cases} L_i, & i \in index \\ 0, & i \notin index \end{cases} \quad (8)$$

The preprocessing algorithm enables recovery of the original phase and charge data without degrading the classification performance.

### 4.3 Comparative Model Evaluation

The architecture and performance of 1D CNN, Transformer, and parallel 1D CNN are compared. The architecture 1 is 1D CNN with dense layer before output layer. The input data is transformed into  $50 \times [1200, 2]$  for compatibility with 1DCNN. Therefore, for 1 second of data for clear and precise classification, final input shape is  $[60000, 2]$ . For architecture 2, transformer input shape is  $[50, 1200]$  since transformer can process data sequence [1]. For architecture 3, input data is the same as that of architecture 1. The model is slightly modified so that the convolution layer is parallel to extract more features from the data. The comparison of the three architectures is illustrated in Fig. 8.

After training the model to convergence, its performance was evaluated under varying noise level. Uniform noise is injected across the entire input array. The noise intensity was systematically varied from 0% to 30% in increments of 5%. For each noise level, three primary performance indicators were measured: classification accuracy, inference time, and the average data transfer

**Table 1:** Accuracy with various noise.

| Noise Level | Architecture 1 | Architecture 2 | Architecture 3 |
|-------------|----------------|----------------|----------------|
| 0%          | 100%           | 100%           | 100%           |
| 5%          | 72.17%         | 94.64%         | 100%           |
| 10%         | 72.17%         | 92.86%         | 100%           |
| 15%         | 72.17%         | 88.39%         | 100%           |
| 20%         | 70.43%         | 79.46%         | 100%           |
| 25%         | 51.3%          | 83.93%         | 100%           |
| 30%         | 51.3%          | 83.93%         | 100%           |

rate required by the model. The accuracy of different architecture is as shown in Tab. 1.

To enable direct comparison among the models, all performance metrics were normalized within the range [0,1]. For metrics where higher values indicated superior performance (e.g., accuracy), a conventional min-max normalization was applied, defined as

$$A_i^{norm} = \frac{A_i - A_{min}}{A_{max} - A_{min}} \quad (9)$$

Conversely, since lower values of inference time and data transfer rate represent better performance, inverse normalization was employed to ensure that higher normalized scores consistently indicated better outcomes across all metrics. The calculation of inference time score and data transfer score is as follow.

$$D_i^{norm} = \frac{D_{max} - D_i}{D_{max} - D_{min}} \quad (10)$$

$$T_i^{norm} = \frac{T_{max} - T_i}{T_{max} - T_{min}} \quad (11)$$

where A is accuracy, D is data transfer rate and T is inference time. The inference time with varying noise levels is shown in table 2, whereas the summary of data transfer in byte is shown in table 3.s

Following normalization, weights were assigned to each metric based on their relative importance, with the total weight summing to unity as follows.

$$W_A + W_D + W_T = 1 \quad (12)$$

In this study, given the focus on partial discharge classification, accuracy was prioritized as the most critical metric and was therefore assigned a weight of  $W_A = 0.6$ . The remaining weights of  $W_D = W_T = 0.2$  each were allocated to inference time and data transfer rate, reflecting their equal but secondary significance. Therefore, model selection considering the accuracy, data transfer rate and inference time becomes:

$$S_i = 0.6A_i^{norm} + 0.2D_i^{norm} + 0.2T_i^{norm} \quad (13)$$

Where,  $A_i^{norm}$ ,  $D_i^{norm}$  and  $T_i^{norm}$  are accuracy, average data transfer rate and inference time of the noise level. The model selection should not be decided over the accuracy only. Therefore, Model selection parameter for

**Table 2:** Inference time.

| Noise Level | Architecture 1 (Seconds) | Architecture 2 (Seconds) | Architecture 3 (Seconds) |
|-------------|--------------------------|--------------------------|--------------------------|
| 0%          | 10.4653                  | 0.7826                   | 1.3905                   |
| 5%          | 5.5668                   | 0.6844                   | 4.9197                   |
| 10%         | 10.5075                  | 0.7622                   | 2.6777                   |
| 15%         | 10.6199                  | 0.7755                   | 1.36                     |
| 20%         | 10.553                   | 0.6996                   | 1.4074                   |
| 25%         | 7.0289                   | 0.6848                   | 1.7353                   |
| 30%         | 10.4361                  | 0.5104                   | 1.3103                   |
| Mean        | <b>9.311071</b>          | <b>0.699929</b>          | <b>2.114414</b>          |

**Table 3:** Average Data Transfer in Byte.

| Architecture 1 | Architecture 2 | Architecture 3 |
|----------------|----------------|----------------|
| <b>1349104</b> | <b>1249108</b> | <b>658103</b>  |

**Table 4:** Model Selection.

| Noise Level | Architecture 1 (Seconds) | Architecture 2 (Seconds) | Architecture 3 (Seconds) |
|-------------|--------------------------|--------------------------|--------------------------|
| 0%          | 0.6                      | <b>0.96714755</b>        | 0.82894236               |
| 5%          | 0                        | <b>0.8515888</b>         | 0.82894236               |
| 10%         | 0                        | 0.813212947              | <b>0.82894236</b>        |
| 15%         | 0                        | 0.716842124              | <b>0.82894236</b>        |
| 20%         | 0                        | 0.550373793              | <b>0.82894236</b>        |
| 25%         | 0                        | 0.76915987               | <b>0.82894236</b>        |
| 30%         | 0                        | 0.76915987               | <b>0.82894236</b>        |

each architecture is calculated using accuracy, inference time and average data transfer of varying noise level as shown in equation 13, and the result is illustrated in table 4. For this PD Classifier prototype, the influence of noise is not very distinct due to the usage of noise free lab simulated data. Therefore, architecture 2 is selected for model implementation because it shows the higher score during early stages of noise injection according to table 4. If the system were implemented in noisy environments, architecture 3 is recommended.

#### 4.4 Embedded Processing

The Raspberry Pi runs a Flask web server listening on port 5000 that receives JSON-formatted PD data via Wi-Fi as shown in Fig. 9. It processes through the loaded model and generates predictions in real time. The sequence of operations is:

1. Initialize GPIO pins and I<sup>2</sup>C OLED display.
2. Establish Wi-Fi connection to predefined network.
3. Load required Python libraries (NumPy, Keras, Flask).
4. Load pre-trained Transformer model from local storage.
5. Activate status LED (GPIO 22, green) confirming successful model load.
6. Start Flask server listening on port 5000.

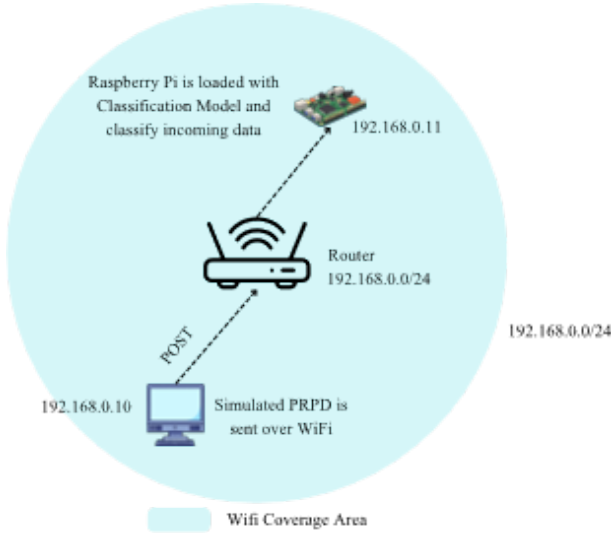


Fig. 9: Working Principle of the system.

7. Receive incoming JSON data containing phase-encoded PD sequence.
8. Convert JSON format to numerical arrays.
9. Execute model prediction on input batch of shape [2, 1200].
10. Classify output (one-hot encoded for four classes).
11. Activate corresponding LED and update OLED display with classification result.
12. Return to listening state.

The flowchart of the PD classification system is demonstrated in Fig. 10.

#### 4.5 Output Feedback

Classification results are displayed on an SSD1306 OLED display (128×64 pixels) showing IP address, Wi-Fi SSID, CPU usage with typical 25% during model initialization. Five GPIO-connected LEDs provide color-coded feedback for operator status indication:

- Status LED is used to indicate when the model is loaded successfully and ready for classification
- Noise LED activates when noise is detected
- Corona LED will turn on when corona discharge is detected
- Surface LED activates when surface discharge is detected
- Internal LED Activates on internal discharge

## 5. RESULTS AND DISCUSSION

The Transformer neural network was evaluated against CNN baselines using the same preprocessed dataset. Validation accuracy was measured over 10 epochs and as highlighted in Fig. 11, transformer model and parallel CNN perform the best with almost 100% accuracy whereas 1D CNN model achieves up to 70% with injection of 15% noise level.

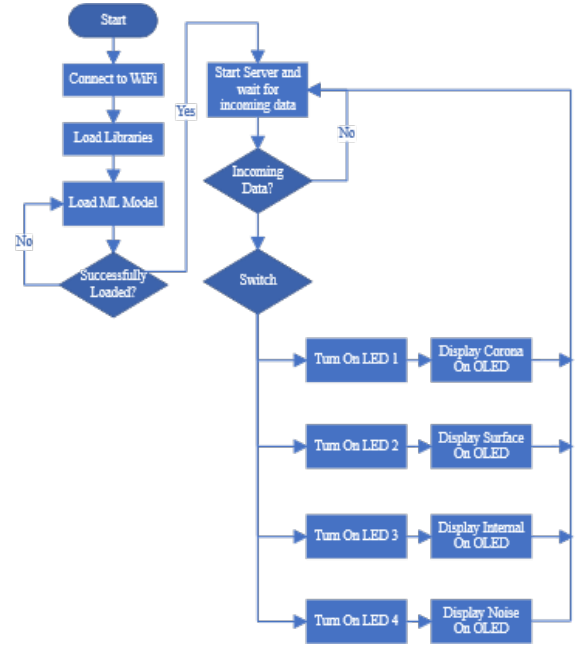


Fig. 10: Flowchart of the PD Classification System.

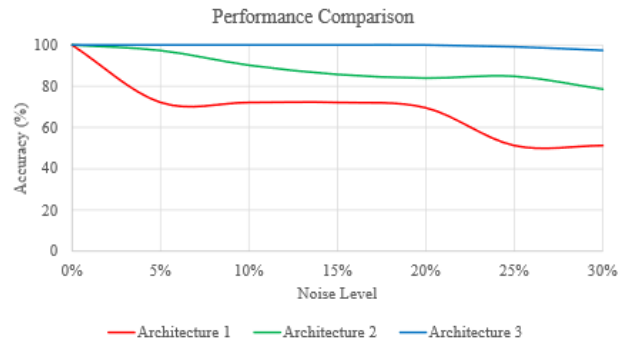


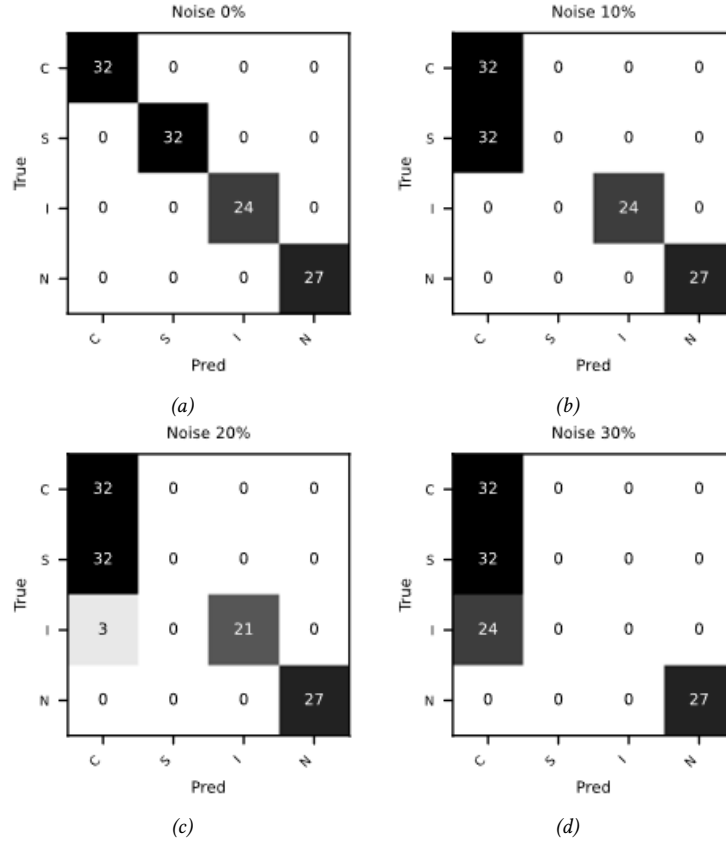
Fig. 11: Model Performance Evaluation.

### 5.1 Model Performance Comparison

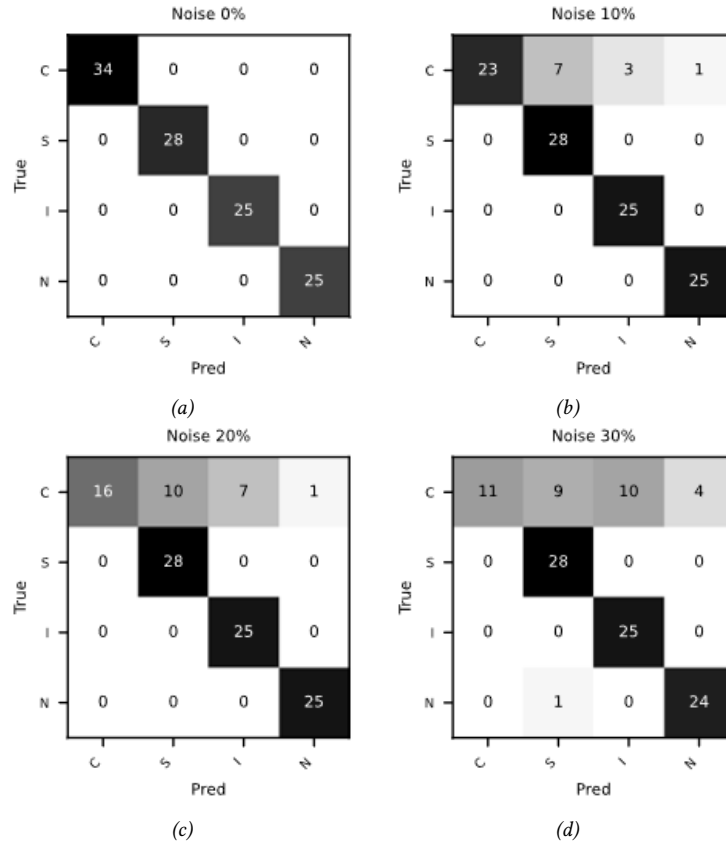
The performance degradation of CNN architectures (Architecture 1) results from their fundamental design. Convolutional layers expect spatially coherent features appearing in consistent locations like an image. In PD classification, discharge features appear at specific phase positions across the 1200-element sequence, violating this assumption which causes misclassification with noise injection as shown in Fig. 12.

Architecture 2 (Transformer) achieves higher accuracy because it captures long-range dependencies within the phase sequence as illustrated in Fig. 12. The multi-head attention mechanism effectively distinguishes subtle variations between discharge types by identifying which phase positions contain discriminative information for each class.

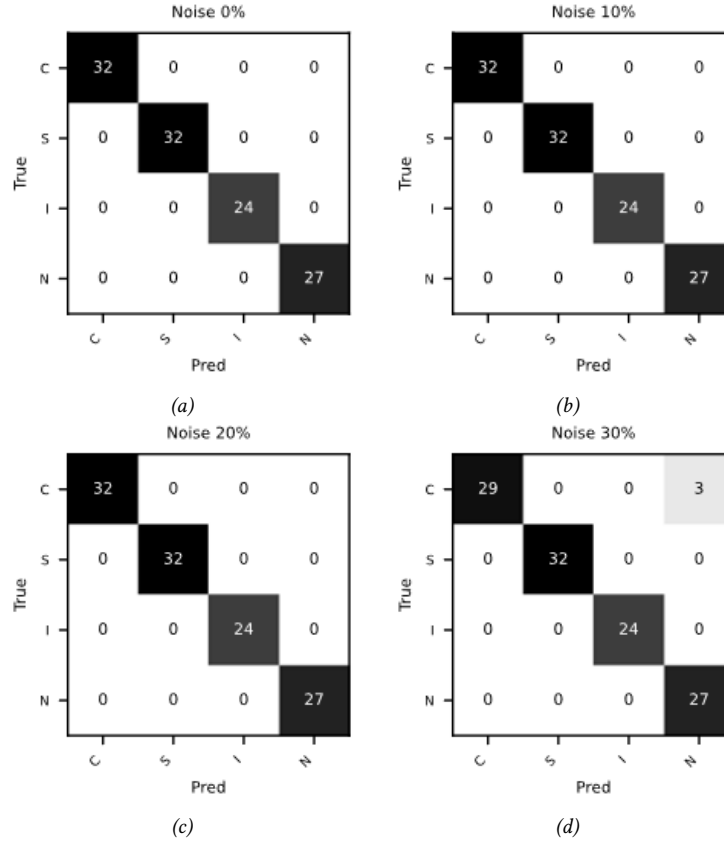
Parallel CNN performs the best because it can extract more underlying features from the input data, resulting in a more robust classification system. Figure 14 demonstrates the performance of Architecture 3 with varying levels of noise injection.



**Fig. 12:** Confusion Matrix of Architecture 1 with noise level(a) 0% (b) 10% (c) 20% (d) 30%.



**Fig. 13:** Confusion Matrix of Architecture 2 with noise level(a) 0% (b) 10% (c) 20% (d) 30%.



**Fig. 14:** Confusion Matrix of Architecture 3 with noise level(a) 0% (b) 10% (c) 20% (d) 30%.

## 5.2 Real-Time Classification Demonstrations

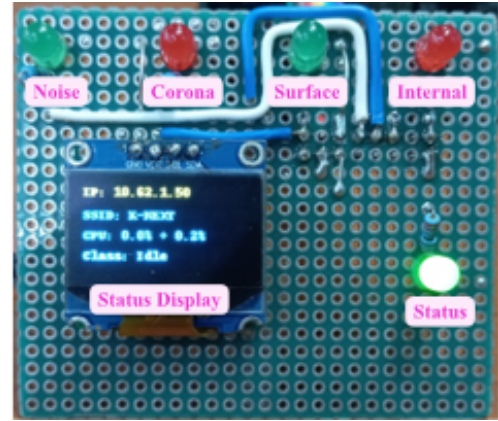
Phase-encoded data for each discharge type was simulated from a PC and transmitted to the Raspberry Pi via JSON over Wi-Fi network. The prototype is designed to classify the incoming PD Data from WIFI [16]. The prototype consists of a raspberry pi and 5 LEDs, assigned for Noise, Corona, Surface and Internal Discharge, and system status. Additionally, an OLED display is attached to display current system IP address, Service Set ID (SSID), CPU usage and PD Class as illustrated in Fig. 15. If the system is ready to classify and waiting for the result, the status LED will light up.

### 5.2.1 Classification of Corona Discharge from our system

Phase encoded PRPD data, which is acquired from conventional partial dischargement measurement method is sent to the PD classification prototype over the Wireless Network. The OLED display will show “Corona”, and second LED will light up as illustrated in Fig. 16.

### 5.2.2 Classification of Surface Discharge from our system

The surface discharge data from test dataset is sent via a computer and the OLED displays “Surface,” and the third LED illuminates, as shown in Fig. 17.



**Fig. 15:** Prototype with LED and OLED Display Outputs.

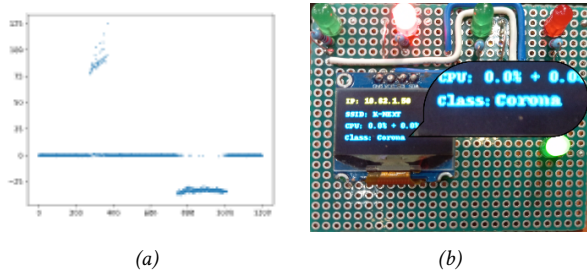
### 5.2.3 Classification of Internal Discharge from our system

Simulated internal discharge data is also labelled as “Internal”, accompanied by the activation of the last LED, as depicted in Fig. 18.

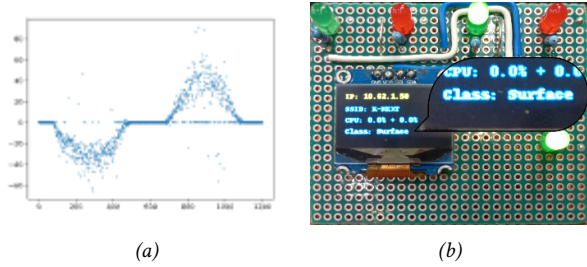
### 5.2.4 Classification of Noise from our system

As illustrated in Fig. 19, if the noise data is injected into the system, the data is labelled as “Noise”, and the first LED will turn on.

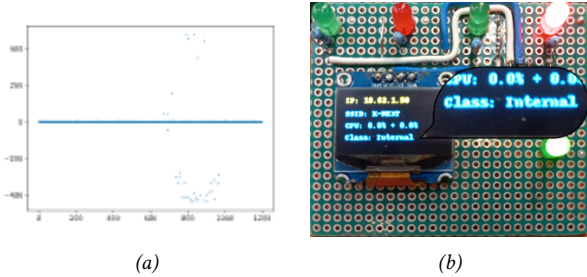
The OLED display successfully update with IP address



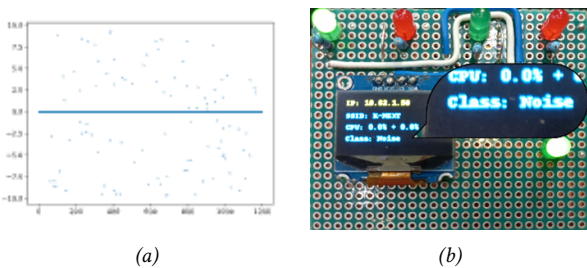
**Fig. 16:** Prepare for PD data for PD Classification Prototype  
(a) Input data, (b) Output from Prototype.



**Fig. 17:** Prepare for PD data for PD Classification Prototype  
(a) Input data, (b) Output from Prototype.



**Fig. 18:** Prepare for PD data for PD Classification Prototype  
(a) Input data, (b) Output from Prototype.



**Fig. 19:** Prepare for PD data for PD Classification Prototype  
(a) Input data, (b) Output from Prototype.

and PD Class, additionally, the classification model typically consumes 0.3% of CPU usage during inference for single client.

## 6. CONCLUSION AND FUTURE WORK

Our system primarily uses PRPD patterns acquired with the MPD 600 in accordance with the IEC 60270

charge-based PD measurement standard. This approach is non-pervasive because it requires electrically attaching the PD acquisition unit to obtain PRPD data, and consequently it is not suitable for online monitoring.

A key limitation is that the HV equipment must be shut down to perform the measurement. This can be addressed by integrating pervasive sensors such as HFCT, UHF, or TEV sensors, which capture PD-related pulses via electromagnetic or surface-current-related phenomena without requiring an electrical connection for conventional apparent-charge measurement.

In the improved workflow, signals would be captured using a custom data acquisition card or an oscilloscope with adequate sampling frequency and resolution to record the time-domain waveform. From the waveform, PD pulses can be extracted and used to construct PRPS, which preserves sequential characteristics of the discharge activity. Under this representation, the advantage of transformer-based neural networks is expected to be more evident because transformers are designed for sequence classification.

## ACKNOWLEDGEMENTS

The author would like to express sincere gratitude to King Mongkut's Institute of Technology Ladkrabang (KMITL) for the continuous academic support throughout this research. Special thanks are dedicated to the DIKA High Voltage Laboratory at KMITL and staff for providing access to essential facilities and for enabling the acquisition of the experimental data used in this study. Appreciation is also given to the School of International and Interdisciplinary Engineering (SIIE) for their administrative and academic guidance. Finally, the author gratefully acknowledges the Department of Electrical Engineering for the foundational knowledge and support that made this work possible.

## REFERENCES

- [1] T. Cheypoca, W. Promphanich, A. Y. Thway, A. P. Hankae, S. Jeenuang and N. Pattanadech, "Partial Discharge Classification with Transformer Neural Networks," *2024 IEEE 14th International Conference on the Properties and Applications of Dielectric Materials (ICPADM)*, Phuket, Thailand, 2024, pp. 93-96, doi: 10.1109/ICPADM61663.2024.10750584.
- [2] V. Chatpattananan, N. Pattanadech and P. Yuthagowith, "Partial Discharge Classification on High Voltage Equipment with K-Means," *2006 IEEE 8th International Conference on Properties & applications of Dielectric Materials*, Bali, Indonesia, 2006, pp. 191-194, doi: 10.1109/ICPADM.2006.284150.
- [3] Pattanadech, N., Nimsanong, P., Potivejkul, S., Yuthagowith, P., Polmai, S., "Partial discharge classification using probabilistic neural network model (2016)," *2015 18th International Conference on Electrical Machines and Systems, ICEMS 2015*, art. no. 7385217, pp. 1176-1180.

- [4] T. Cheypoca, W. Promphanich, A. Y. Thway, A. P. Hankae, S. Jeenmuang and N. Pattanadech, "Partial Discharge Classification With 1D Convolutional Neural Network," *2024 IEEE 14th International Conference on the Properties and Applications of Dielectric Materials (ICPADM)*, Phuket, Thailand, 2024, pp. 89-92, doi: 10.1109/ICPADM61663.2024.10750640.
- [5] Chatpattananan, V., Pattanadech, N., Yutthagowith, P., "Partial discharge classification on high voltage equipment with K-means cc(2006)," *Proceedings of the IEEE International Conference on Properties and Applications of Dielectric*.
- [6] N. Pattanadech, R. Haller, S. Kornhuber and M. Muhr, "Partial Discharges (PD): Detection, Identification and Localization" : Wiley, 2023.
- [7] Pattanadech, N., Pratomosiwi, F., Muhr, M., Baur, M., "The influence of the test methods on the Partial Discharge Inception Voltage value of the mineral oil using the needle-Plane electrode configuration (2012)," *Proceedings of 2012 IEEE International Conference on Condition Monitoring and Diagnosis, CMD 2012*, art. no. 6416215, pp. 597-600.
- [8] Pattanadech, N., Pratomosiwi, F., Wieser, B., Baur, M., Muhr, M., "The study of partial discharge inception voltage of mineral oil using needle - Plane electrode configuration (2012)," *ICHVE 2012 - 2012 International Conference on High Voltage Engineering and Application*.
- [9] N. Pattanadech, F. Pratomosiwi, B. Wieser, M. Baur and M. Muhr, "The study of partial discharge inception voltage of mineral oil using needle - Plane electrode configuration," *2012 International Conference on High Voltage Engineering and Application*, Shanghai, China, 2012, pp. 174-177, doi: 10.1109/ICHVE.2012.6357020.
- [10] N. Pattanadech, S. Potivetkul and P. Yuttagowith, "Corona Phenomena of Various High Voltage Shielding Types," *2006 International Conference on Power System Technology*, Chongqing, China, 2006, pp. 1-6, doi: 10.1109/ICPST.2006.321552.
- [11] IEC Standard IEC-60270, "High Voltage Techniques Partial Discharge Measurement", International Electrotechnical Commission 2000.
- [12] N. Pattanadech, A. A. Kemma, T. F. Sipahutar, F. Pratomosiwi and M. Muhr, "The possibility of using a needle plane electrode for partial discharge inception voltage measurement," *2013 Annual Report Conference on Electrical Insulation and Dielectric Phenomena*, Chenzhen, China, 2013, pp. 1258-1261, doi: 10.1109/CEIDP.2013.6748212.
- [13] Y. LeCun, K. Kavukcuoglu and C. Farabet, "Convolutional networks and applications in vision," *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, Paris, France, 2010, pp. 253-256, doi: 10.1109/ISCAS.2010.5537
- [14] S. Kiranyaz, T. Ince, O. Abdeljaber, O. Avci and M. Gabbouj, "1-D Convolutional Neural Networks for Signal Processing Applications," *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019, pp. 8360-8364, doi: 10.1109/ICASSP.2019.8682194.
- [15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need (2017), doi: 10.48550/arXiv.1706.03762
- [16] N. Pattanadech, "Leveraging AI and IoT for Enhanced Reliability in Smart Electrical Power Systems," *Keynote address at the Int. Conf. on Intelligence of Things (ICIT 2025)*, Bangkok, Thailand, Nov. 17-18, 2025.



**Theerayod Wiangtong** received the B.Eng. degree (first-class honors) in electronic engineering from King Mongkut's Institute of Technology Ladkrabang, the M.Sc. degree in satellite communication from the University of Surrey, and the Ph.D. degree in digital system design, co-design from Imperial College, London, UK. Currently, he is an Associate Professor with the Department of Electrical Engineering, King Mongkut's Institute of Technology Ladkrabang. His research interests include Digital IC design, FPGA, reconfigurable computing platform, embedded system, video/image processing, optimization in hardware/software co-design, computational processing implemented in FPGA, IoTs, and their applications.



**Chanin Bunlaksananusorn** received a Ph.D. degree in electrical engineering from The University of Edinburgh, UK. He is currently with the School of Engineering, King Mongkut's Institute of Technology Ladkrabang (KMUTL). His research area is electrical engineering.



**Aung Ye Thway** is currently pursuing bachelor's degree in electrical engineering from King Mongkut's Institute of Technology Ladkrabang (KMUTL). His research area are partial discharge detection and classification and signal processing for high-voltage insulation monitoring.



**Siwakorn Jeenmuang** received the B.Eng. and M.Eng. degrees in electrical engineering from the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand, in 2021 and 2023, respectively. He is currently Ph.D student at King Mongkut's Institute of Technology Ladkrabang. His research interests mainly include high-voltage measurement techniques, high-voltage equipment monitoring and diagnosis and rotating machines.



**Norasage Pattanadech** received his B.Eng and M.Eng degrees in electrical engineering from the King Mongkut's Institute of Technology Ladkrabang in 1998 and Chulalongkom University, Thailand in 2002, respectively. He is also awarded his Ph.D. degree by the Institute of High Voltage Engineering and System Management, Graz University of Technology, Austria in 2013. Currently, he works as an associate professor at the King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand. His research activities have been mainly involved with partial discharge in insulating liquid, solid insulator characteristics, high-voltage testing and high voltage measurement techniques, high voltage equipment monitoring and diagnostics, lightning and grounding problems and electromagnetic compatibility in high-voltage field.