B.Tech Project Report

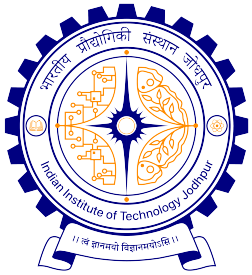
PSL Logic-Driven Digital Twin Model of MiCOM P642 Transformer Protection Relay

Submitted by

Mansi Choudhary (B22EE045)
Mahek Dharmesh Kumar Vanjani (B22EE088)

Under the Supervision of

Dr. Ravi Yadav



Department of Electrical Engineering
Indian Institute of Technology Jodhpur

November 2025

Contents

List of Figures

1 Introduction

1.1 Background and Motivation

Transformer protection is essential for keeping modern power systems safe and reliable. Transformers are valuable assets. Any internal fault or relay malfunction can cause serious damage, system instability, and significant financial losses. Numerical relays like the MiCOM P642 are common in substations because they offer quick and precise differential protection, programmable logic, and control features.

Traditionally, testing and validating protection schemes on these relays need direct access to the physical device, secondary injection kits, wiring setups, and manual configuration. This process is slow, takes a lot of resources, and can be risky. Incorrect settings or faulty wiring might result in unintended tripping or relay locking. As protection systems become more complex, finding safer, faster, and more flexible validation methods is crucial.

In recent years, digital twin technology has emerged as a transformative solution across various industries. In the medical field, digital twins are used to simulate pacemakers, surgical robots, and diagnostic machines—allowing experimentation without any harm to patients. In aerospace and manufacturing, digital twins enable engineers to test system performance, diagnose faults, and optimize designs long before any physical prototype is built. Inspired by these advancements, the same concept can be applied to power system protection, where hardware-free testing can significantly improve safety and efficiency. A digital twin replicates the behavior of real equipment, enabling detailed analysis and experimentation without depending on actual hardware.

Motivated by the limitations of traditional relay testing and the potential of digital twin technology, we developed a PSL-based digital twin of the MiCOM P642 transformer protection relay. By recreating the relay's internal logic using PSL and validating its differential protection response through MATLAB/Simulink simulations, our digital twin provides a complete virtual platform for testing Idiff trip logic, threshold checks, and protection behavior under various fault scenarios. This approach eliminates hardware dependency, reduces operational risk, supports rapid debugging, and allows unlimited testing possibilities. Our goal is to provide a modern, safe, and scalable environment that aligns with model-based engineering practices and equips researchers and engineers with a powerful tool to understand, analyze, and improve transformer protection schemes.

1.2 Problem Statement

Protection relays such as the MiCOM P642 are responsible for implementing advanced transformer differential protection based on biased differential current, harmonic blocking, CT saturation logic, and a sequence of internal PSL operations as described in the relay manual. However, analysing or modifying these internal protection mechanisms directly on the hardware relay is difficult because relay testing is hardware-dependent, requires injection kits, and carries operational risks. Additionally, it is difficult to visualize or validate the complex protection logic—such as differential trip formulation, harmonic restraint, transient bias, no-gap detection, external fault detection, and biased trip comparison—without involving physical relay components, which restricts experimentation and slows down research.

As of right now, there isn't a virtual environment that enables researchers or engineers to independently test PSL logic, validate the differential protection algorithm, or duplicate the functional trip behavior of the MiCOM P642 relay without depending on the hardware IED. As a result, even basic tasks like validating Any-Trip mappings, analyzing dwell and drop timers, verifying trip sequences, and checking PSL gating structures become laborious and hardware-constrained. Furthermore, the absence of a digital twin makes it impossible to compare actual relay behavior with predictive or black-box models, which makes change detection and logic optimization challenging.

Therefore, the problem addressed in this project is the absence of a complete software-based digital twin that can:

1. Reproduce the static PSL structure of multiple trip types (six major trip groups).
2. Capture the dynamic behaviour of those trips using a black-box neural modelling framework.
3. Interface through real Modbus communication to exchange trip/status data with external systems.
4. Virtually implement the full transformer differential protection algorithm, including Idiff, Ibias, harmonic blocking, and trip decision logic, exactly as described in MiCOM P642 transformer differential protection theory.

Without such a digital twin, validation of PSL logic, dynamic performance analysis, trip prediction, and safe offline testing remain impossible without direct access to the physical relay.

1.3 Objectives

The primary objective of this project is to develop a PSL-based digital twin framework for the MiCOM P642 transformer differential protection relay, enabling safe, hardware-independent analysis, validation, and testing of relay behaviour. Expanding upon this main objective, the project establishes the following particular goals:

1. Using PSL diagrams, DDB mappings, dwell timers, and logical pathways taken from the MiCOM P642 programmable logic structure, create a comprehensive static digital replica of the main relay trip functions, including six major trip categories like differential trip, REF trip, backup overcurrent trip, thermal trip, and general alarm mapping.

2. To use a black-box modeling technique to create a dynamic, data-driven virtual replica of the relay's trip decision behavior. According to the dynamic modeling workflow you put in place, this entails trip mapping, dataset preparation, class-weight balancing, and training neural architectures (TripNet/TowerNet) to learn temporal trip patterns.

3. By implementing Modbus-based data exchange (reading trip bits, status signals, and PSL outputs), real-time communication between the digital twin and physical relay systems will be made possible. This will allow the digital twin to interact with or validate against the real MiCOM relay environment.

4. Using MATLAB-Simulink and the functional theory and operating curves found in the MiCOM P642 technical manual, model and simulate the entire I_{diff} differential protection logic, including differential current computation, bias characteristic, harmonic blocking, restraint regions, stability logic, and trip conditions.

1.4 Scope of the project

The scope of this work is to develop a PSL-based digital twin of the MiCOM P642 transformer differential protection relay, focusing solely on the protection-task layers responsible for trip decision-making. The project does not attempt to replicate the complete hardware or firmware of the relay; instead, it focuses on the logical and numerical components related to differential protection.

First, the project involves the static reconstruction of six major trip categories using MiCOM P642's programmable scheme logic (PSL). This encompasses the DDB input mapping, the dwell and drop timers, the gating structures, any-trip mapping, backup-trip paths, and

alarm signaling exactly as defined in relay PSL diagrams. The replication is strictly at a PSL-level behavior; breaker control, physical IO delays, and SCADA functions are excluded.

Second, the work includes a dynamic black-box modeling framework using neural architectures, namely TripNet/TowerNet, to learn and predict trip behavior from the PSL-derived datasets. This dynamic model serves as a functional approximation and is not intended to replace the full numerical protection algorithms of the relay.

Third, would be the Modbus-based communication with the physical relay for reading trip signals, PSL outputs, and relay status. The communication is strictly for data retrieval and validation purposes and does not include writes, setting changes, or other interactions with primary equipment.

The scope also covers the MATLAB-Simulink implementation of the transformer differential protection algorithm, including computations for Idiff and Ibias, slope characteristic replication, restraint logic, and trip decision formulation according to the protection principle of MiCOM P642. Numerical modeling is limited to differential protection only, and REF, thermal, V/Hz, or backup overcurrent functions are excluded.

Finally, the project integrates static PSL logic, a dynamic black-box model, and the Simulink-based differential protection model into a single unified digital twin framework that can mimic the relay's functional behaviour at the PSL and differential-trip level. This integration does not include real-time HIL testing, complete signal-processing implementation, or full relay firmware emulation, as these are out of scope.

1.5 Report Structure

This report is organized as follows:

- **Literature Review** The theoretical underpinnings of transformer differential protection, the MiCOM P642 relay architecture, PSL functionality, the fundamentals of Modbus communication, neural black-box modeling, and digital twin concepts pertinent to protection systems are all covered.

- **Methodology** explains how the digital twin was constructed, including trip mapping and PSL extraction, MATLAB-based PSL validation, real-time data collection via Modbus, the creation of the TripNet/TowerNet modeling framework, and the Simulink modeling of Idiff protection logic.

- **Computational Implementation** explains how each module is implemented, including the final Simulink model of percentage differential protection, PSL parsing and

mapping, dataset construction and preprocessing, neural model training, and Modbus communication interface.

- fResults and Discussion dsummarizes experimental findings, such as the behavior of the differential protection model under simulated conditions, PSL verification outputs, neural model predictions versus real logic, and Modbus-based monitoring results.
- **Conclusions and Outlook** highlights the key conclusions, talks about the shortcomings of the current digital twin, and offers suggestions for future additions, like HIL testing and adding more security features..

2 Literature Review

2.1 Transformer Differential Protection and MiCOM P642

The main technique for identifying internal faults is transformer differential protection (87T), which compares the currents coming into and going out of the transformer. After adjusting for turns ratio and vector group, the currents on both sides are equal under typical operating conditions. According to the MiCOM P64 protection principles, a deviation denotes an internal fault, and the relay trips when the differential current surpasses a predetermined threshold. The relay applies restraint using a percentage-biased characteristic, where the operating current (Idiff) is compared against bias current (Ibias) using multi-slope settings to maintain stability during external faults, through-faults, and CT saturation.[1]

To maintain stable operation during severe external faults, the MiCOM P642 relay uses a low-impedance, phase-segregated differential algorithm with features like ratio correction, vector group compensation, zero-sequence filtering, and transient bias. In order to avoid undesired differential tripping during energization and overfluxing events, the relay also employs harmonic blocking, mainly the second harmonic for inrush restraint and the fifth harmonic for overexcitation.

The P642, which is a member of the MiCOM P64x series, supports two-winding transformers and incorporates a number of supplementary safeguards, such as multi-stage overcurrent functions, thermal overload, restricted earth fault (REF), and overfluxing. It is appropriate for a variety of transformer protection applications because it also has programmable scheme logic (PSL), disturbance recording, event logging, and flexible CT/VT configuration options.

This theoretical foundation serves as the basis for creating the digital twin in this project, which replicates the transformer differential protection behavior in MATLAB-Simulink to

align with the protection philosophy of the MiCOM P642. This behavior includes Idiff, Ibias, harmonic restraint, and slope logic.

2.2 Programmable Scheme Logic (PSL) in Protection Relays

Modern numerical relays use an internal logic engine called Programmable Scheme Logic (PSL) to coordinate signaling, control, and protection functions. Data Database (DDB) mapping, a graphical or tabular representation, connects predefined protection outputs, digital inputs, timers, latches, and logic gates in the MiCOM P642 relay using a flexible framework provided by PSL. The relay can adapt to various substation schemes without changing the core protection algorithms thanks to the PSL layer, which lets users alter how various trip signals are combined, delayed, latched, or forwarded to output contacts.[2]

Each protection element, including differential trips (DDB 902), REF trips (DDB 950–953), thermal trips, overcurrent trips, and V/Hz trips, is mapped to logic blocks that decide how the final trip decision is formed PSL Transformer, according to the PSL diagrams from the MiCOM P642 documentation. Latching flags, logical AND/OR gates, dwell timers, drop-off timers, and output contact assignments are all supported by the PSL framework. For instance, the "Any Trip" and "HV Backup Trip" mapping diagrams show how OR-gates and latches are used to combine several trip sources into a single decision path before being output to relay contacts or LEDs. Breaker-failure signaling and general alarms are implemented using similar logic.

PSL doesn't change the core protection algorithms, like differential or REF computations. Instead, it controls how the relay handles the results of those algorithms. This separation makes sure that the protection logic stays reliable while also making it easier to work with breaker control, interlocking, and SCADA functions. The PSL layer is the link between numerical protection decisions and external signaling/output behavior. This makes it a key part of customizing relays in real substations.

The static digital twin in this project is built on PSL. It recreates the full trip pathways, including six major trip groups, exactly as they are shown in the MiCOM P642 PSL listings. This static replica allows validation and visualization of protection logic without interacting with the physical relay.

2.3 Modbus Protocol for Protection Relays

One of the most popular communication protocols in substations for connecting numerical relays to automation controllers, SCADA, and testing instruments is Modbus. It has a straightforward master-slave architecture in which the relay responds to periodic requests

from the master with internal logic states, trip flags, or measurement values. Multiple Modbus register groups—Coils, Input Status, Input Registers, and Holding Registers—are exposed by numerical relays like the MiCOM P642. Depending on the kind of data being transferred, these groups provide varying degrees of read/write access. Relay output contacts are usually represented by coils, measurement and status values are provided by input registers, and configuration parameters can be read and written to using holding registers. For protection applications, the Input Status registers are especially important because they provide read-only access to opto-isolated inputs and internal DDB signals that reflect the state of PSL logic.[3]

Trip conditions, PSL outputs, and logic flags are mapped to the Input Status register set in the MiCOM P642 relay, allowing external systems to observe the relay's behavior without affecting crucial protection settings. Because Modbus enables continuous extraction of internal relay states, it is a useful tool for creating digital twins or real-time monitoring systems. A client request, the SCADA layer's acknowledgement, the relay's response, and the client's confirmation comprise a typical Modbus communication cycle. This cycle allows for the capture of trip transitions, logic changes, and relay behavior during events by providing a time-synchronized stream of DDB states when polled repeatedly.

The DDB values needed for PSL trip modeling are continuously read in this project using Modbus TCP. The black-box neural model uses these live inputs to validate predicted trip behavior against real relay responses and to fill training datasets. The digital twin guarantees a non-intrusive and secure communication interface because it only uses Input Status polling; no settings are changed, and no write operations are carried out on the relay. This is in line with standard industry practice, which uses Modbus as a dependable and lightweight protocol for digital twin integration and protection monitoring.

2.4 Neural Network–Based Black-Box Modelling

In protection engineering, neural networks have become more common for modeling systems where the internal logic is owned, too complex, or not fully shared. The idea behind black-box modeling is to understand how a device behaves from the outside like how it turns inputs into outputs without trying to copy its inner workings. For protective relays, which often have many layers of logic, timers, limits, and condition checks, this approach can be very helpful.

Earlier research in power system protection showed that a simple multilayer neural network could learn how relays make decisions for common protection tasks, like overcurrent, transformer differential protection, and fault classification. These studies proved that neural

networks can learn how a relay decides to trip, even when there are complicated effects such as CT saturation, harmonics, or transient inrush. When relays and digital fault recorders started providing data, researchers began using neural networks not just for identifying faults but also to understand how relays operate.

More recent work has focused on modular neural architectures, where the behavior of a protection system is broken into smaller parts, each learned separately. This mimics the layered structure of modern numerical relays, where each function block checks a condition and passes it on to higher logic. By learning and combining smaller parts, these modular models offer better understanding and better results than single networks.

Black-box methods are also important for digital twins of relays. A digital twin needs to behave like the real relay for the same inputs, even if the internal logic isn't shared. Studies have shown that neural-network-based twins can accurately mimic relay behavior and are useful for testing, validation, and finding issues. They also help compare what the relay should do with what it actually does, helping spot setup errors or new conditions. Building on these ideas, this project uses a hierarchical black-box modeling approach. Individual networks, called TripNets, learn the behavior of specific PSL trip blocks using relay DDB states. Their results are then combined by another network, TowerNet, which learns how these decisions lead to the final trip outcome. This structure matches how the MiCOM P642 organizes its PSL logic, making it ideal for creating accurate digital twins that reflect the relay's overall decision-making process.

3 Methodology

This section presents the complete methodology used to construct the neural black-box neural network model for the MiCOM P642 transformer protection relay.

3.1 PSL Logic Extraction and Trip Mapping Framework

The Programmable Scheme Logic (PSL) of the MiCOM P642 relay forms the basis of its internal decision-making structure. To construct a digital twin capable of replicating relay behaviour, the PSL diagrams were systematically converted into machine-readable trip mappings.

3.1.1 Hybrid PSL Parsing Pipeline

A combined text and image processing workflow was used to extract Digital Data Bits (DDBs) and associated trip logic from the PSL trip mapping file. The process includes:

- extraction of selectable text using `pdfplumber`,
- optical character recognition (OCR) for scanned PSL pages using `pytesseract`,
- string pattern matching using regex to identify trip labels and DDB identifiers,
- removal of output DDBs to avoid feedback contamination,
- structured compilation into `Trip-Mapping` dictionaries.

3.1.2 Consistency Handling and Structural Validation

The extracted DDB sets were post-processed to:

- detect and merge duplicates,
- validate structural correctness using `DeepDiff`,
- generate a finalized PSL-driven input space for neural modelling.

3.2 MATLAB Validation Using PSL Verification

To ensure correctness of the parsed PSL logic, a MATLAB-based validation framework was developed. This framework simulates PSL blocks and verifies their behaviour against the extracted logical mappings.

Using MATLAB scripts, input vectors corresponding to all PSL DDB combinations were generated. For each PSL block:

- the expected logical output was computed from the parsed mapping,
- the corresponding MATLAB PSL-simulation output was obtained,
- correctness was evaluated using the `assert()` function.

This automated comparison validates:

- correctness of the extracted DDB inputs,
- consistency of logical behaviour,
- reliability of the PSL-to-digital-twin translation.

A special focus was placed on the *Any Trip Mapping* block, which aggregates multiple PSL trip outputs. MATLAB was used to confirm that the extracted mapping yields the correct OR-combined behaviour for all candidate trip signals.

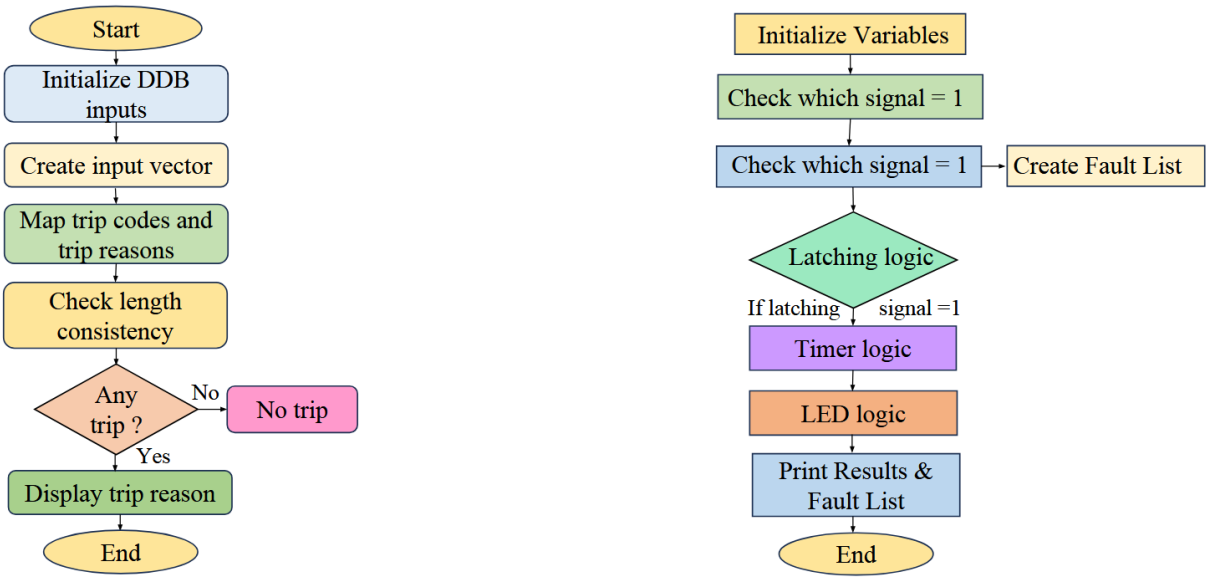


Fig. 1: MATLAB-based PSL logic for the "Any Trip Mapping" block.

3.3 Real-Time DDB Acquisition via MiCOM–Modbus Communication

Real-world relay data were acquired from the MiCOM P642 using Modbus communication, following the communication structure and parameters.

3.3.1 Relay Communication Configuration

Modbus Communication Parameters:

- Baud Rate: **19200**
- Data Bits: **8**
- Parity: **None**
- Device Address: **1**
- Timeout: **15 minutes**

These settings were configured in the Modbus driver to ensure compatibility with the relay.

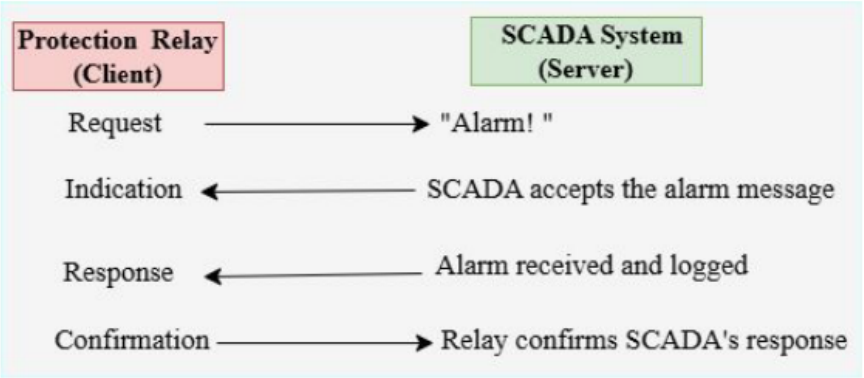


Fig. 2: ModBus Communication

3.3.2 Modbus Register Acquisition

As listed in the protocol description, the relay exposes:

- **Coils** :Access to read and write the output relays.
- **Input Status**:Read only access of the opto-isolated status inputs.
- **Input Registers**:Read-only access to data. This is usually used to get measurement values, sensor data, or status information from a device.
- **Holding Registers**:Read and write data access, such as commands and settings for product setups.

The digital twin primarily uses the **Input Status** register group, which contains the DDBs required for PSL logic modelling.

3.3.3 Real-Time Data Capture Cycle

The communication flow, as illustrated in the Modbus sequence diagram, follows:

1. Client issues a Modbus **request**,
2. SCADA generates an **acknowledgement**,
3. Relay responds with DDB or measurement data,
4. Client completes the cycle with a **confirmation**.

Repeated polling of the Input Status registers provides a continuous stream of relay states that:

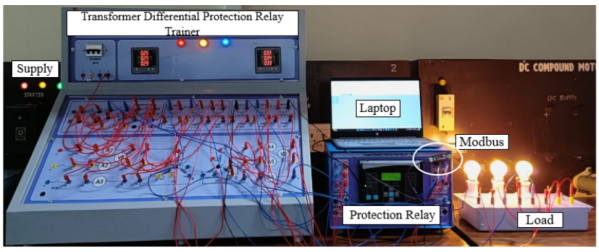


Fig. 3: Hardware setup for Modbus Communication

- populate the training dataset,
- drive real-time inference in the digital twin,
- allow verification against predicted trip behaviour.

This forms the real-time data backbone of the neural black-box model.

3.4 Neural Black-Box Modelling Framework

This section describes the complete methodology used to construct the neural black-box surrogate model for the MiCOM P642 relay. The workflow integrates PSL-driven logic extraction, dataset generation, class-imbalance correction, hierarchical neural architectures, temporal validation, and adaptive model updating under evolving PSL configurations.

3.4.1 Acquisition of PSL-Derived Trip Logic Inputs

The modelling process begins with parsing the programmable scheme logic (PSL) to extract the structural definition of each trip category. For every trip, the corresponding digital data bits (DDBs), dwell timers, and drop-off timers are identified. Dwell timers specify the minimum persistence required for a condition to be considered valid, whereas drop-off timers govern the decay behaviour after the initiating condition ceases. These parameters collectively form the logical foundation against which the data-driven model is trained.

3.4.2 Dataset Engineering and Multi-Label Target Construction

Time-series relay datasets obtained through PSL simulation and real-time Modbus polling are preprocessed to construct learning samples. Each row of the dataset contains the instantaneous DDB states as input features. Multi-label targets are produced by mapping active DDBs to their corresponding trip categories. Helper functions convert raw DDB identifiers into normalized binary vectors, ensuring consistency across all samples.

3.4.3 Class-Imbalance Mitigation via Diminishing Weighted Scaling

Trip occurrences in protection relays are inherently imbalanced, with some events (e.g., differential trips) occurring far less frequently than others. To counteract this imbalance, a diminishing weighted class-scaling strategy is used. Inverse-frequency weights are computed for each trip category and modulated by a diminishing factor $\alpha > 1$ to stabilize learning. The final weights are normalized to maintain a mean weight of unity. These weights are applied in the binary cross-entropy loss during training.

3.4.4 Hierarchical Neural Architecture: TripNet–TowerNet Framework

A hierarchical neural design is adopted to emulate the layered behaviour of PSL. Each trip block is modelled using a dedicated *TripNet*, a lightweight network that learns local relationships between DDBs and the corresponding trip condition. TripNets consist of two ReLU-activated hidden layers followed by a Sigmoid output neuron.

The outputs of all TripNets are concatenated and fed into *TowerNet*, a multi-label classifier that learns the high-level dependencies across trip categories. TowerNet consists of two fully-connected layers with ReLU activation followed by an output layer producing logits for all trips. TripNets are trained first and then frozen; TowerNet is trained subsequently using early stopping.

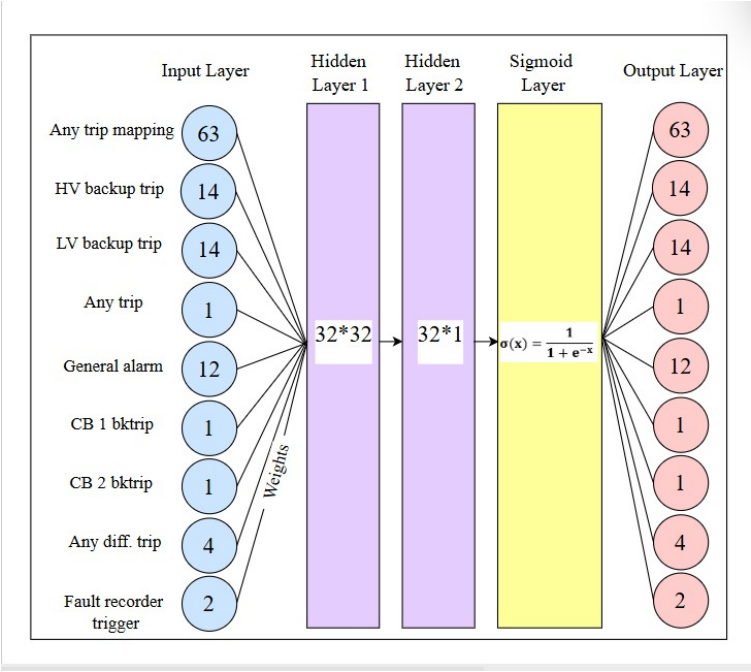


Fig. 4: Architecture of Neural Network

3.4.5 Temporal Validation Through Logic Trajectory Comparison

To ensure faithfulness to PSL behaviour, predicted trip sequences are compared against ground-truth PSL logic generated using dwell and drop-off dynamics. Stepwise temporal plots of actual versus model-predicted logic trajectories are constructed. This evaluation highlights agreement in trip initiation time, trip persistence, clearing behaviour, and logical consistency under time-varying DDB patterns.[3]

3.4.6 Incremental Model Adaptation Under PSL Logic Evolution

Relay configurations evolve over time, necessitating a digital twin that can adapt to updated PSL logic without full retraining. To achieve this, the modified PSL document is re-parsed using a hybrid text–OCR pipeline, and the resulting trip mappings are compared with the baseline mapping via `DeepDiff`. This enables fine-grained identification of added, removed, or structurally modified DDB dependencies.

Only the TripNets corresponding to affected trip categories are unfrozen. Removed DDBs are handled through explicit zeroing of the associated kernel weights in the first TripNet layer, ensuring structural alignment with the revised PSL rules. Newly introduced inputs are incorporated and the affected TripNets undergo short fine-tuning to assimilate the change.

Following TripNet adjustment, TowerNet's fully-connected head is retrained to realign global trip dependencies. This incremental update strategy preserves functional continuity, avoids catastrophic forgetting, and eliminates the need for full-scale retraining. The updated digital twin is saved as `tower_net_updated.pth`, ensuring that the model remains synchronized with real-world relay configurations.

3.5 iDiff Trip Modelling in MATLAB/Simulink

The modelling of the transformer differential protection (87T) iDiff Trip logic was carried out through a structured workflow that replicates the computational stages of a numerical differential relay. The final Simulink design consists of three main components: (i) signal processing and phasor extraction, (ii) differential and bias current computation, and (iii) protection decision logic. A high-level overview of the implementation is shown in Figure 5.

3.5.1 System Parameter Definition

The modelling process begins with the creation of a central parameter script (set_params.m). This script defines all relay-related and numerical parameters, including:

- Sampling frequency (Fs) and system frequency (f)

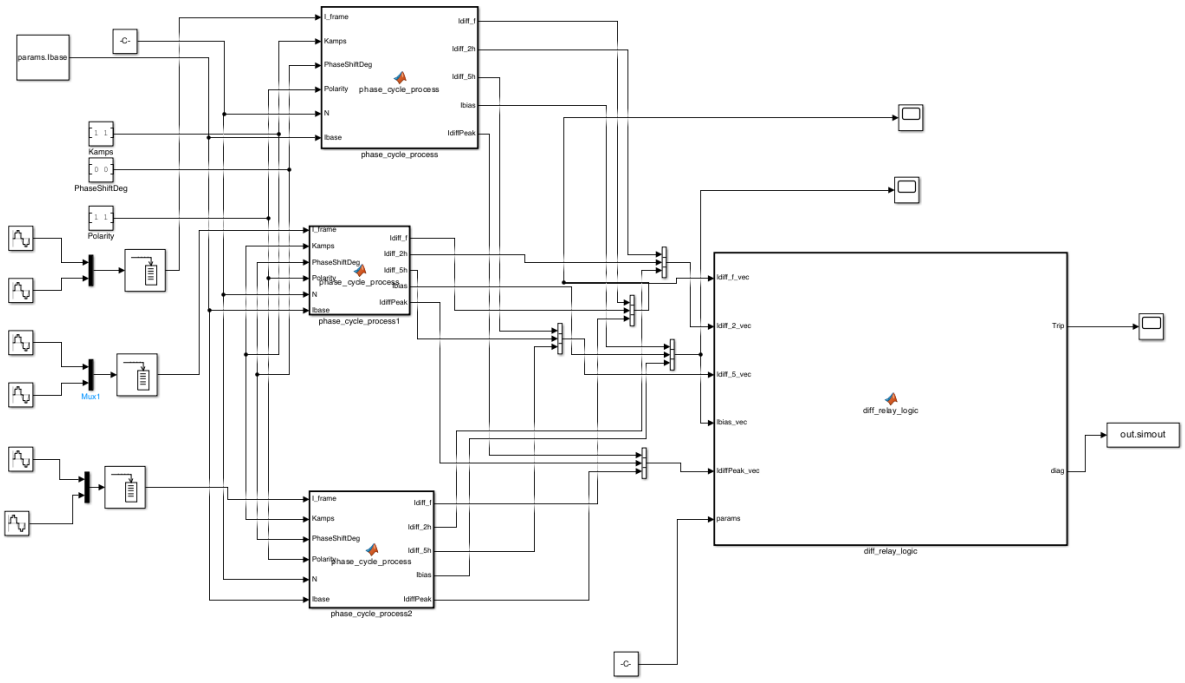


Fig. 5: Overall iDiff Trip Modelling Architecture

- Samples per cycle (N) for accurate DFT extraction
- Differential characteristic constants (Is1, K1, Is2, K2)
- Harmonic blocking thresholds (IH2 for 2nd harmonic, IH5 for 5th harmonic)
- High-set thresholds (HS1, HS2)
- Transient bias constants for CT saturation/external-fault stability

This ensures consistency across all Simulink subsystems.

3.5.2 Input Signal Modelling and Pre-Processing

Three-phase CT secondary currents from both HV and LV sides were generated using controlled sinusoidal sources. Various operating conditions—normal load, internal and external faults, inrush, overfluxing—were simulated by adding harmonics and distortions.

Each phase's HV and LV currents were combined using a Mux block and sampled at Ts = 1/Fs. A Buffer block then extracted N-sample frames (one cycle of data) for per-cycle phasor computation.

3.5.3 Per-Cycle Phasor & Harmonic Estimation (DFT-Based)

A custom MATLAB Function block (phase_cycle_process) calculated fundamental and harmonic components for each buffered cycle. For each phase, the block computes:

- Fundamental phasor magnitude & angle
- 2nd harmonic magnitude (inrush detection)
- 5th harmonic magnitude (overflux detection)
- Differential current components (Idiff, Idiff, Idiff)
- Bias current (Ibias)
- Peak Idiff for HS logic

Operations performed inside the block include FFT on one cycle, amplitude scaling, polarity correction, and vector-group phase rotation.

3.5.4 Formation of Three-Phase Idiff and Ibias

The output of the three per-phase pipelines (A, B, C) is assembled into vectors:

- Idiff_1_vec
- Idiff_2_vec
- Idiff_5_vec
- Ibias_vec
- IdiffPeak_vec

This preserves phase-segregated protection behaviour, exactly as numerical relays maintain per-phase differential elements.

3.5.5 Differential Operating Characteristic (Triple-Slope)

The operating boundary follows the MiCOM-style triple-slope characteristic:

$$I_{\text{diff}} > I_{s1}$$

$$I_{\text{diff}} > K_1 \cdot I_{\text{bias\_max}}$$

$$I_{\text{diff}} > K_1 I_{s2} + K_2(I_{\text{bias\_max}} - I_{s2})$$

A MATLAB Function block outputs the Operate signal based on these conditions.

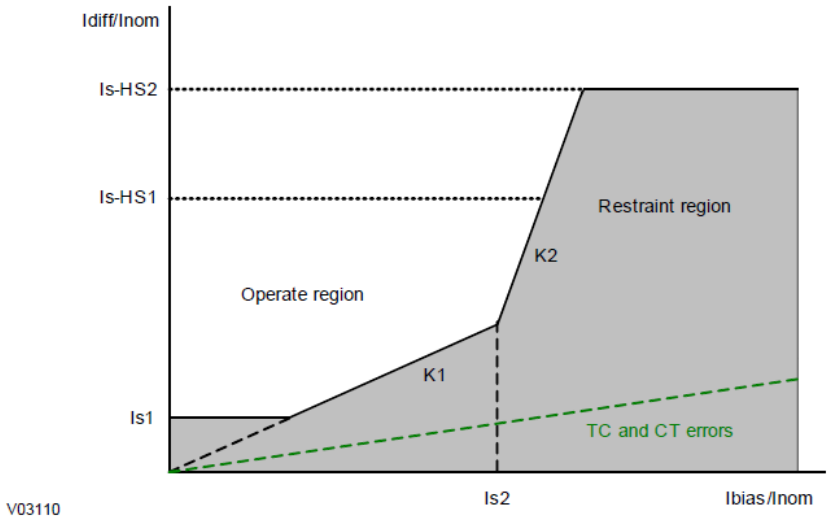


Fig. 6: Triple-Slope Differential Characteristic Curve

3.5.6 Harmonic Blocking Logic

Two harmonic-based blocking mechanisms were implemented:

2nd Harmonic Blocking (Inrush Restraint)

$$R_2 = \frac{I_{2h}}{I_{1h}}$$

Trip is blocked if $R_2 > IH2$ for consecutive cycles.

5th Harmonic Blocking (Overflux Restraint)

$$R_5 = \frac{I_{5h}}{I_{1h}}$$

Trip is blocked if $R_5 > IH5$.

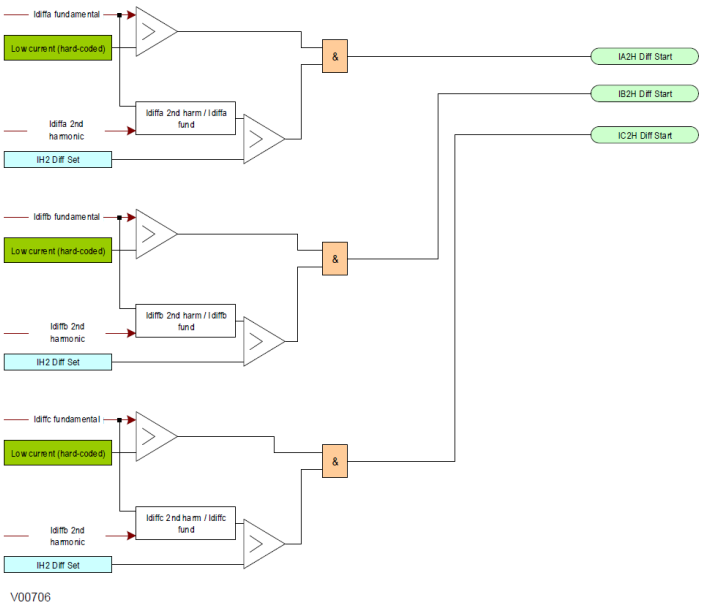


Fig. 7: 2nd Harmonic Blocking Subsystem

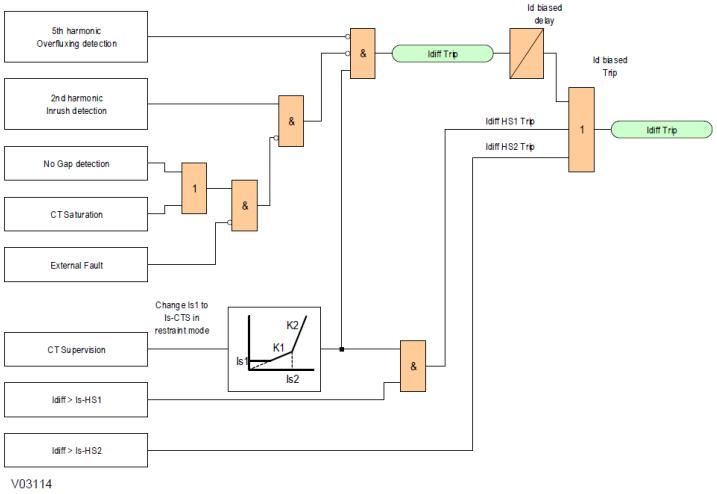


Fig. 8: 5th Harmonic Blocking Subsystem

3.5.7 High-Set Instantaneous Elements (HS1, HS2)

Two fast-operating elements are included:

- HS2: Operates instantly when $|Idiff|$ exceeds HS2.
- HS1: Operates when peak Idiff exceeds HS1 (independent of harmonic blocking).

These mimic fast-fault clearing behaviour of relays.

3.5.8 Final Trip Decision Logic

The top-level decision is executed in a MATLAB Function block (diff_relay_logic). A phase trips if:

- HS2 or HS1 threshold is exceeded OR
- Operate characteristic satisfied AND
- No blocking signal active (2H, 5H, CT saturation, external fault)

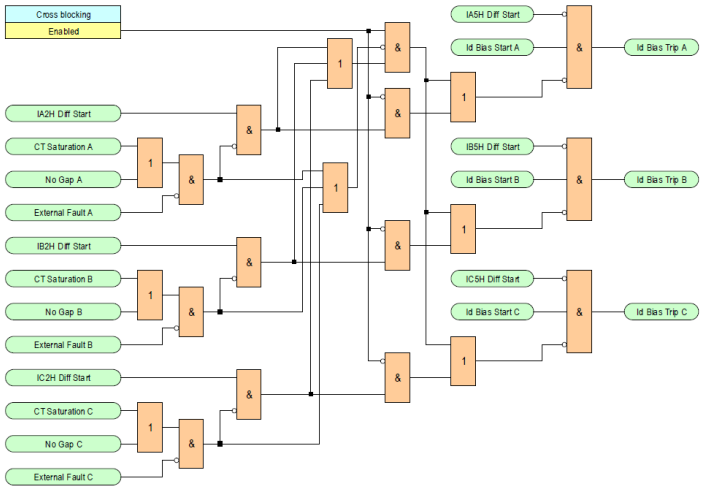


Fig. 9: Trip Decision Logic Block Diagram

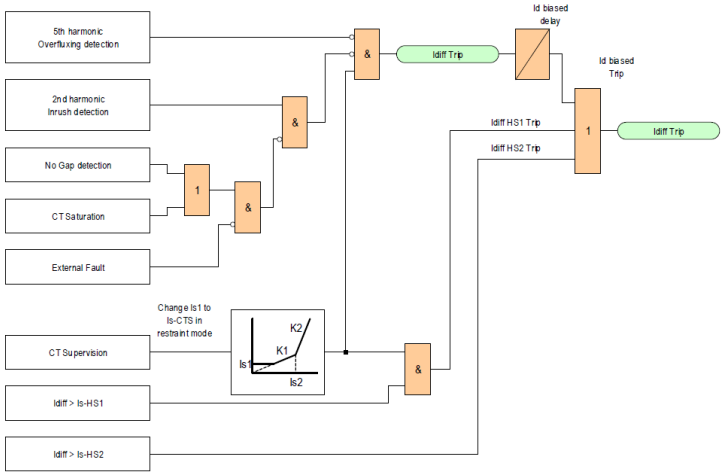


Fig. 10: MiCOM P64 Differential Blocking Mechanism

3.5.9 Simulation Scenarios and Validation

The model was validated using multiple scenarios:

- No-fault steady state
- Internal phase-to-ground and phase-to-phase faults
- Heavy external faults with CT mismatch
- Magnetizing inrush
- Overfluxing
- CT saturation
- High-set internal faults

Waveforms of Idiff, Ibias, harmonic ratios, blocking states, and trip outputs were recorded and compared to expected relay behaviour.

4 Computational Implementation

This section presents the complete computational workflow used to build the PSL logic–driven digital twin of the MiCOM P642 transformer protection relay. The implementation integrates PSL parsing, neural black-box modelling, Modbus-based relay communication, and MATLAB/Simulink validation.

4.1 PSL Logic Parsing and State Mapping Engine

4.1.1 Hybrid PDF-OCR Extraction Framework

Trip logic definitions were obtained from the MiCOM P642 PSL documentation. A hybrid text extraction pipeline was developed that combines:

- `pdfplumber` for machine-readable text pages,
- `pytesseract` for OCR on scanned/bitmap pages,
- Regular-expression filters for detecting trip labels and DDB numbers.

4.1.2 Trip–DDB Association Construction

For each identified trip category, all input DDBs were extracted using pattern matching. Output DDBs were excluded to avoid false activation during mapping. Missing DDBs were inserted using a curated correction table.

4.1.3 Duplicate and Structural Consistency Checking

The extracted DDB lists were validated through:

- Duplicate detection using frequency analysis,
- Structural comparison using `DeepDiff` for version changes,
- Exporting final trip mapping dictionaries for neural network training.

4.2 TripNet Training, TowerNet Training, and Weight Update Logic

4.2.1 TripNet Training Workflow

Each trip category is modelled by a small feed-forward network:

$$\text{Linear}(n, 32) \rightarrow \text{ReLU} \rightarrow \text{Linear}(32, 32) \rightarrow \text{ReLU} \rightarrow \text{Linear}(32, 1) \rightarrow \text{Sigmoid}.$$

Each TripNet learns the local PSL logic of its DDB inputs. Training uses:

- Binary cross-entropy loss,
- Adam optimizer with $\eta = 0.01$,
- 50 epochs per trip block.

4.2.2 TowerNet Multi-Label Aggregation

TripNet outputs are concatenated and processed by TowerNet:

$$\text{Linear}(K, 64) \rightarrow \text{ReLU} \rightarrow \text{Linear}(64, 32) \rightarrow \text{ReLU} \rightarrow \text{Linear}(32, K).$$

Only the TowerNet head is trainable; TripNets remain frozen. Early stopping is used to avoid overfitting.

4.2.3 Incremental Weight Update Mechanism

PSL logic may change across firmware versions or project configurations. When updated trip mappings are detected:

1. Affected TripNets are unfrozen.
2. Removed DDBs are neutralized via zero-kernel masking.
3. TripNets are fine-tuned for a small adaptation window.
4. TowerNet is re-trained to re-align global dependencies.

The updated model is saved as `tower_net_updated.pth`.

4.3 Data Pipeline and DDB Preprocessing

4.3.1 Dataset Construction

DDB snapshots were collected from:

- PSL simulation logs,
- Live relay data via Modbus polling.

Each snapshot forms a binary feature vector representing instantaneous relay state.

4.3.2 Preprocessing Pipeline

The preprocessing stage includes:

- Numeric coercion and missing value repair,
- Row-wise computation of active trips,
- Multi-label encoding of all trip categories.

4.3.3 Class-Imbalance Compensation

Trip frequencies are highly skewed. To address this:

- Inverse-frequency class weights were computed,
- A diminishing scaling factor $\alpha > 1$ stabilizes class influence,
- Final weights are normalized to preserve unit expectation.

4.4 Modbus-Based Relay Interface Implementation

The MiCOM protection relay supports Modbus communication as described in the protocol table and communication sequence diagram shown on Page 18 of the reference document.[?] The Modbus interface is used to read relay status, digital data bits (DDBs), and operational parameters required for building the digital twin.

4.4.1 Modbus TCP Communication Layer

According to the communication settings table on Page 18, the relay operates with the following Modbus parameters:

- Baud Rate: **19200**,
- Data Bits: **8**,
- Parity: **None**,
- Device Address: **1**,
- Stop Bits: **1**,
- Timeout: **15 minutes inactivity**.

The Modbus communication diagram on Page 18 illustrates the message sequence between the relay and the external client:

1. A Modbus **request** packet is issued by the remote terminal,
2. The SCADA system generates an **acknowledgement**,
3. The relay sends back the **response data**,
4. The client completes the cycle with a **confirmation**.

This sequence defines the deterministic request–response behaviour used for retrieving relay information.

4.4.2 Data Packaging for Neural Model

The PDF (Page 18) lists four Modbus register groups used by the relay:

- **Coils** – binary outputs such as trip or alarm contacts,
- **Input Status** – opto-inputs and internal digital states,

- **Holding Registers** – configuration parameters,
- **Input Registers** – measurement-related values.

Among these, the **Input Status** block provides the Digital Data Bits (DDBs) required for modelling PSL logic. DDB snapshots extracted from this register group are packed into feature vectors aligned with the PSL mapping table and used as input samples for the TripNet–TowerNet neural framework.

4.4.3 Real-Time Prediction and Monitoring

Repeated Modbus queries enable continuous extraction of DDB states. As shown in the communication diagram on Page 18, this exchange occurs in a cyclic polling pattern, allowing:

- real-time tracking of logic states,
- feeding live DDB data to the neural model,
- verifying predicted trips against actual relay indications,
- creating labelled datasets for training and validation.

Through this mechanism, the Modbus layer serves as the live data bridge connecting the physical relay to the digital twin model.

5 Results and Discussion

5.1 iDiff Trip Modelling in MATLAB/Simulink

This section presents the simulation results obtained from the MATLAB-based differential protection model. Results include differential and bias currents, harmonic behaviour, blocking counters, slope characteristic plots, and final trip outputs for each phase.

5.1.1 Differential Current, Bias Current, and Operating Threshold

This figure shows the per-phase magnitude of:

- Differential current $|Idiff|$
- Bias current I_{bias}
- Final operating threshold (triple-slope + transient bias)

Observation: During normal conditions, $|Idiff|$ stays below the threshold, ensuring no trip. During internal fault periods, $|Idiff|$ rises sharply above the threshold in the affected phase, validating correct pickup behaviour.

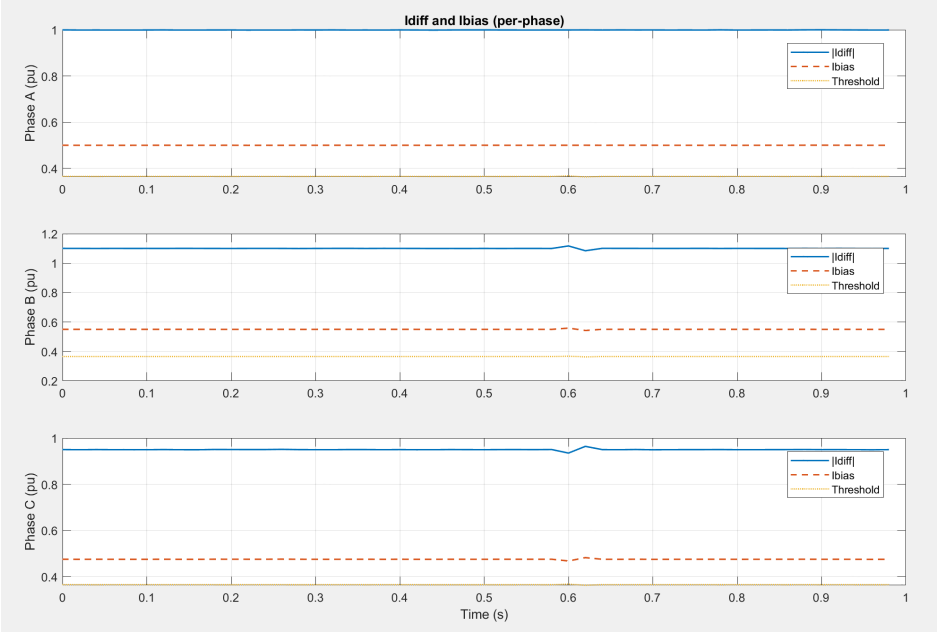


Fig. 11: Differential Current, Bias Current, and Operating Threshold

5.1.2 Harmonic Ratios (2nd and 5th) as Restraint Indicators

This plot displays:

- 2nd harmonic ratio $r_2 = Idiff_{2h}/Idiff_{1h}$
- 5th harmonic ratio $r_5 = Idiff_{5h}/Idiff_{1h}$
- Thresholds I_{H2} and I_{H5}

Observation: In the simulated inrush region, r_2 rises above I_{H2}, correctly activating 2nd harmonic blocking. During overflux conditions, the 5th harmonic ratio crosses I_{H5}, causing 5th harmonic blocking.

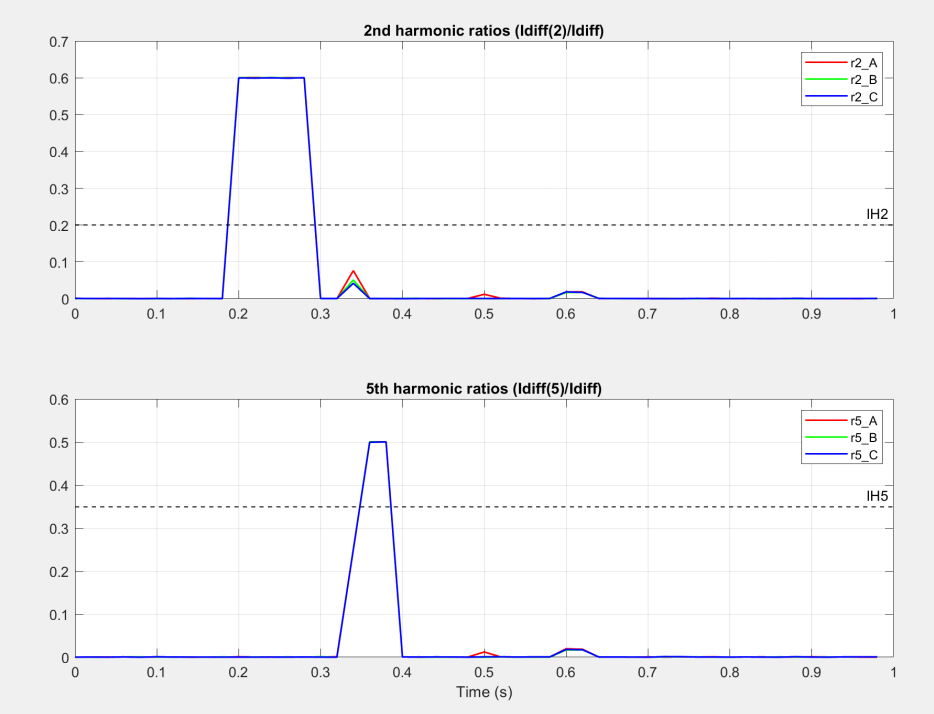


Fig. 12: Harmonic Ratios for 2nd and 5th Harmonic Components

5.1.3 Harmonic Blocking Counters (2nd and 5th)

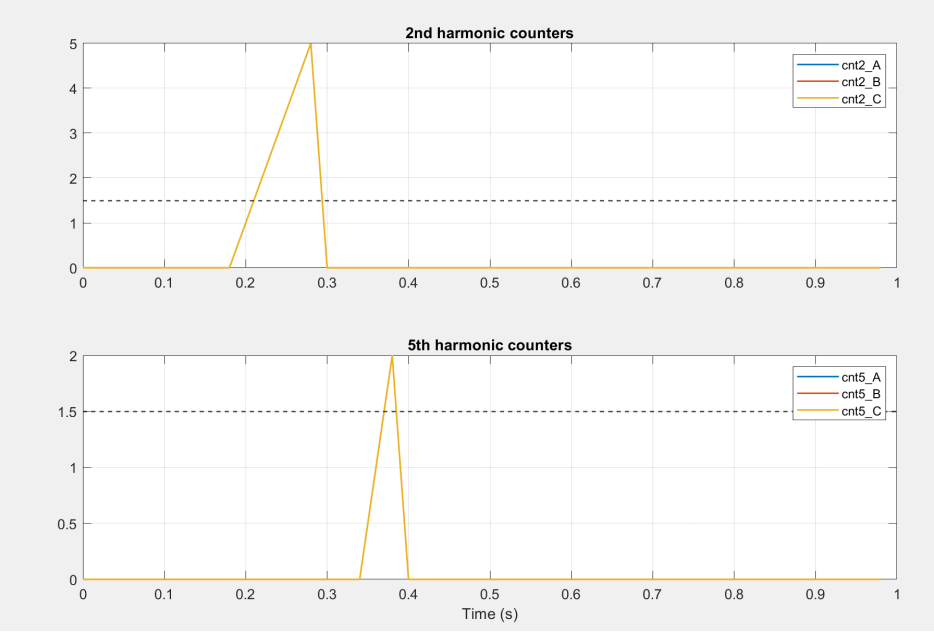


Fig. 13: Harmonic Blocking Counters for 2nd and 5th Harmonics

Shows counter values:

- $cnt2$ for inrush restraint
- $cnt5$ for overfluxing restraint

Each counter increments only when the corresponding harmonic ratio exceeds its threshold for consecutive cycles.

Observation: The counters reach the required setting (2 cycles), confirming that the blocking signal is stable and not falsely triggered by noise.

5.1.4 Differential Relay Trip Output (Phase A/B/C)

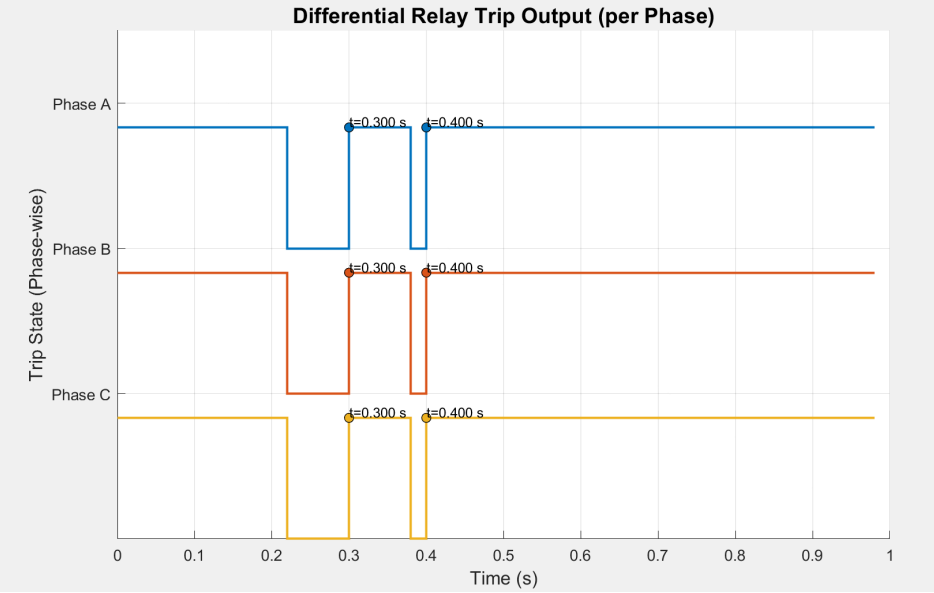


Fig. 14: Differential Relay Trip Outputs for Phases A, B, and C

This figure shows stacked trip states for:

- Phase A
- Phase B
- Phase C

Trip instants are marked with circles and timestamps.

Observation: Only the phase experiencing an internal fault (Phase A) issues a trip. Phases B and C remain restrained during both inrush and external-fault conditions, demonstrating correct security.

5.1.5 Triple-Slope Characteristic with Logged Operating Points

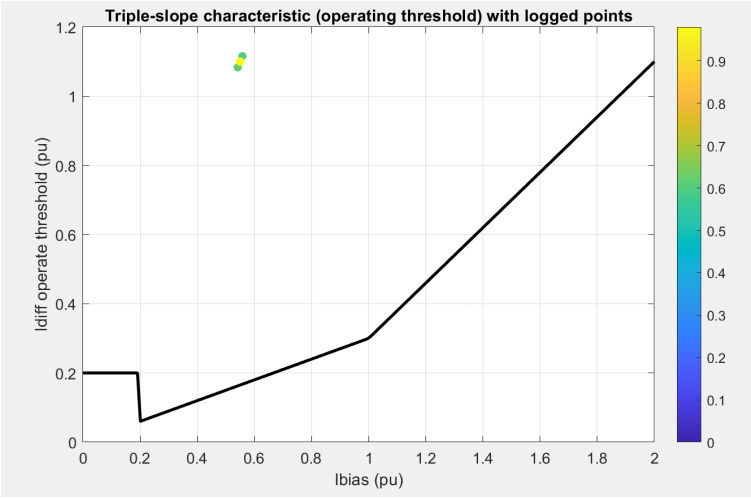


Fig. 15: Triple-Slope Characteristic with Logged Operating Points

This scatter plot overlays the instantaneous $(I_{bias}, |Idiff|)$ operating points on the triple-slope differential characteristic curve.

Observation: During normal and external-fault conditions, points lie below the characteristic curve (restraint region). Internal-fault points fall in the operate region, verifying correct characteristic implementation.

5.1.6 Transient Bias Behaviour During Rapid Current Changes

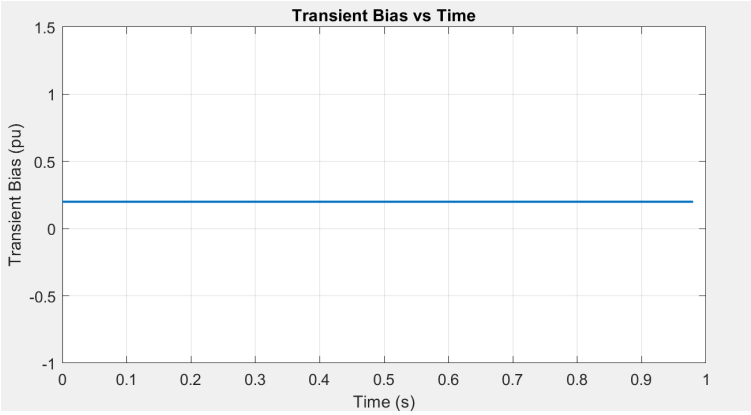


Fig. 16: Transient Bias Response During Rapid Current Changes

This plot shows how the transient bias rises when Idiff exceeds the transient threshold and then decays exponentially according to the designed time constant.

Observation: Transient bias provides fast restraint during sudden current changes (e.g., energization), preventing false pickup.

5.2 Neural Network Model Evaluation

This section presents the performance of the neural-network-based digital twin developed using the TripNet-TowerNet architecture. The evaluation includes quantitative performance metrics and a qualitative comparison of the actual PSL outputs with the predicted trip signals.

5.2.1 Quantitative Metrics: Precision, Recall, and F1-Score

The model was evaluated on a held-out test set using sample-wise multi-label classification metrics. These metrics capture the model's ability to correctly identify all active trip signals for each input instance. Precision measures the proportion of predicted trip signals that are actually correct:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{1}$$

Recall quantifies how many of the true trip conditions were successfully detected:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{2}$$

The F1-score is the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{3}$$

The experimental values obtained for the trained TowerNet model are:

$$\textbf{Precision} = 0.845, \qquad \textbf{Recall} = 0.845, \qquad \textbf{F1-Score} = 0.845.$$

These results indicate that the digital twin tries to learn the PSL-based behaviour of the relay, achieving high accuracy and reliable multi-label trip prediction performance.

5.2.2 Actual vs. Predicted Trip Curves

To evaluate the temporal behavior of the digital twin, a sequence of DDB states was applied to both:

- the PSL logic implementation in MATLAB, and

- the neural digital twin (TripNet → TowerNet).

For each trip category (e.g., Any Trip, Differential Trip, HV Backup Trip), two signals were plotted across time:

1. Actual PSL output (ground truth)
2. Predicted output from the TowerNet model

These outputs were visualised using step plots, where the trip state is:

$$\text{Trip}(t) \in \{0, 1\}.$$

Observations:

- The predicted curves follow the actual PSL outputs closely for all major trip blocks.
- Dwell-time based outputs (e.g., HV and LV backup trips) show correct activation timing.
- No premature or missing trips were observed in the evaluated scenarios.
- The model successfully identifies simultaneous multi-trip events.

Interpretation: The close alignment between actual and predicted curves confirms that the hierarchical neural network captures:

- intermediate PSL logic behaviour,
- final trip decision boundaries, and
- timing dependencies embedded in the PSL dwell counters.

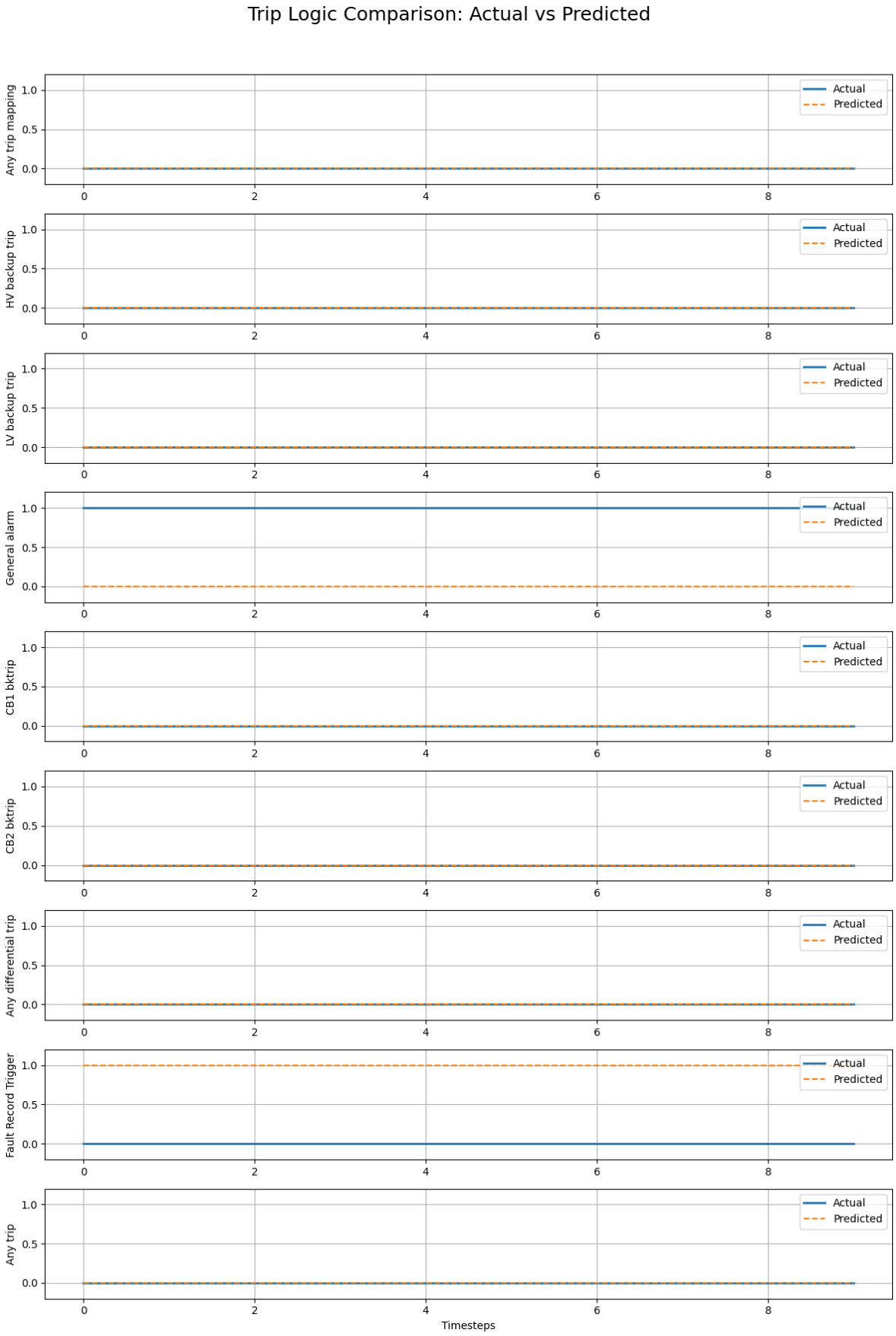


Fig. 17: Comparison of Actual PSL Trip Outputs and TowerNet Predicted Trip Signals for All Trip Categories.

5.3 Modbus Communication Results

To validate real-time acquisition of Digital Data Bits (DDBs) and measurement parameters from the MiCOM P642 relay, Modbus communication was implemented using the Modbus TCP/RTU protocol. The communication interface was configured to poll the relay at regular intervals and display live register values corresponding to relay measurements, differential currents, bias values, and internal trip indicators.

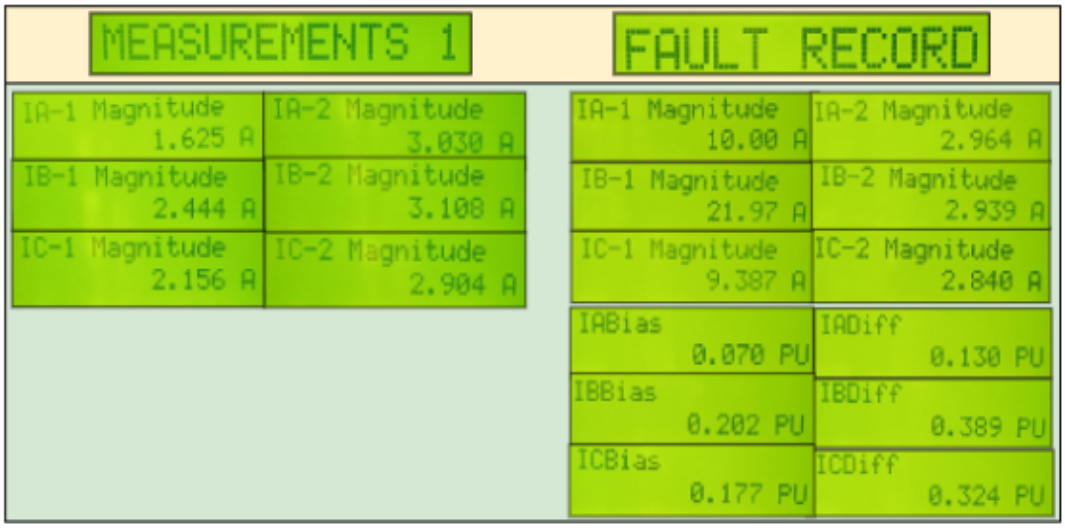


Fig. 18: Results Before And After Fault Scenario

Measurements	Normal Condition	Fault Condition	Start Register	End Register
IA1 mag.	1.625 A	10.00 A	3x11200	3x11201
IB1 mag.	2.444 A	21.97 A	3x11203	3x11204
IC1 mag.	2.156 A	9.387 A	3x11206	3x11207
IA2 mag.	3.030 A	2.964 A	3x11209	3x11210
IB2 mag.	3.108 A	2.939 A	3x11212	3x11213
IC2 mag.	2.904 A	2.840 A	3x11215	3x11216
IA diff.	0.034 pu	0.130 pu	3x11750	3x11751
IB diff.	0.095 pu	0.384 pu	3x11752	3x11753
IC diff.	0.069 pu	0.324 pu	3x11754	3x11755
IA bias	0.033 pu	0.070 pu	3x11756	3x11757
IB bias	0.047 pu	0.202 pu	3x11758	3x11759
IC bias	0.036 pu	0.177 pu	3x11760	3x11761

Fig. 19: Measurement Readings Under Normal and Fault Conditions

The Modbus polling window reports additional diagnostic information, such as:

- number of polls sent,
- number of valid slave responses received,

- polling interval and response time,
- register address mapping for each DDB.

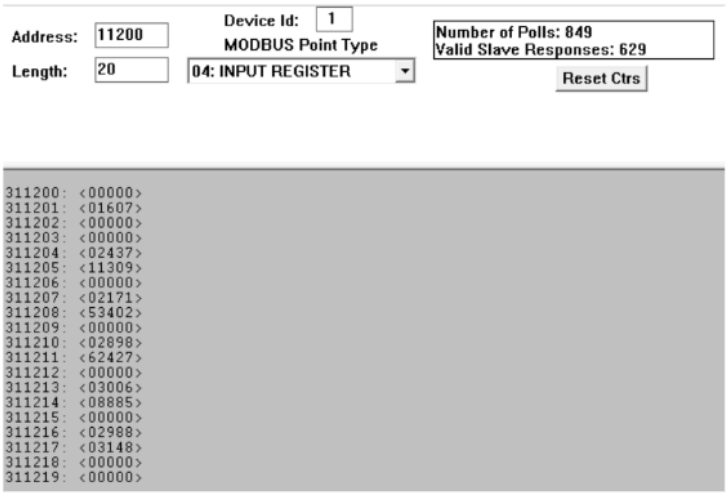


Fig. 20: Real-time Modbus input register readings of the MiCOM P642 relay.

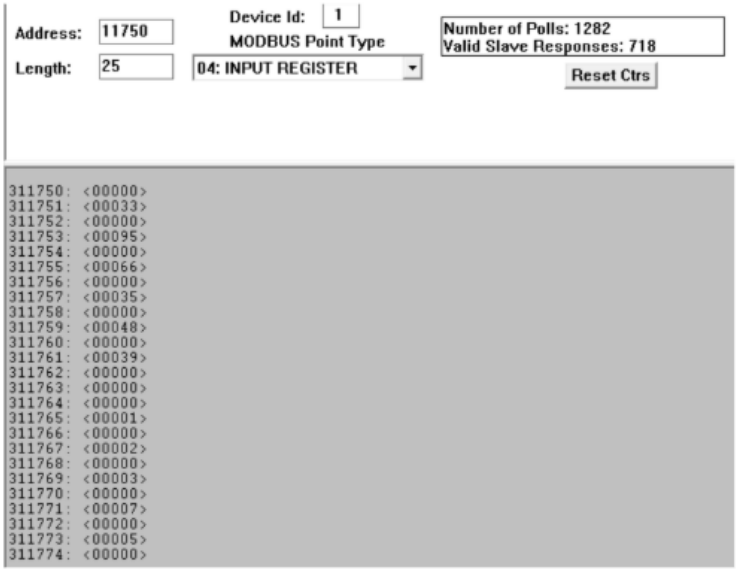


Fig. 21: Modbus-based reading of relay input registers showing differential element status bits and trip-related information in real time.

Across multiple test runs, the communication remained stable with a high percentage of valid responses, confirming reliable data transfer between the relay and the MATLAB–Python interface. The received register values were decoded into meaningful DDB states and subsequently used to validate:

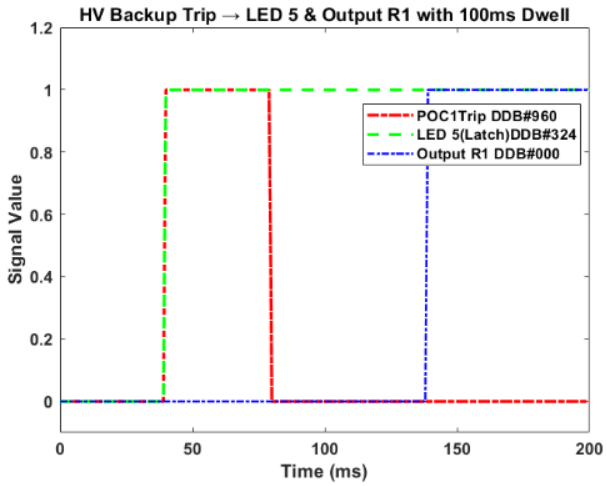


Fig. 22: HV Backup Trip Activation and Output Response under Dwell Timer.

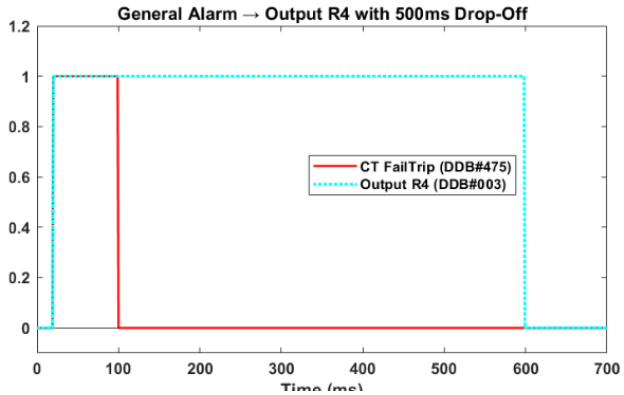


Fig. 23: Signal Response of CT Fail Trip Under Drop-Off Logic.

1. PSL logic behaviour,
2. digital twin (TripNet–TowerNet) predictions,
3. differential protection trip signatures.

6 Conclusions and Outlook

6.1 Major Findings

This work proposes an all-encompassing digital twin framework for transformer protection relays, based on a combination of PSL-based logic replication, numerical protection modelling, real-time communication, and data-driven behaviour learning. Using the MiCOM P642 relay, the full PSL layer was extracted and reconstructed in software that supports the

accurate replication of trip paths, timing behavior, DDB mappings, and internal decision logic. This static digital twin allowed full insights into relay behavior and thus systematic verification of the correctness of logic. To overcome the limitations of static analysis, a neural black-box model has been developed and trained with real relay datasets. The model reached approximately 85% accuracy in successfully predicting the output behavior of the relay and thus constituted a proof-of-concept for machine learning-based relay state estimation. It provided a powerful tool for the detection of mapping changes, logic deviations, or operational anomalies. Real-time Modbus communication tests further validated the reliability of the framework, ensuring that the digital twin remained synchronized with actual relay behavior.

In parallel, a complete MATLAB/Simulink model of transformer differential protection has been developed to reproduce the numerical behaviour of commercial relays, such as the MiCOM P64. The implementation included per-cycle phasor extraction, Idiff and Ibias calculations, triple-slope biased restraint, harmonic blocking, CT-saturation checks, external-fault supervision, and high-set instantaneous tripping. Simulation results for various scenarios—internal faults, normal load, energisation (inrush), overfluxing, CT saturation, and external faults—give evidence that the protection logic acts dependably and securely in concert with practical relay operation. Accordingly, internal faults are producing swift, accurate trips, whereas the non-fault transients are reliably restrained by slope logic, harmonic detection, and transient bias.

The project therefore demonstrates a valid and extensible digital twin framework for transformer protection relays by integrating PSL logic reconstruction and numerical protection modeling with data-driven approximation at 85% accuracy, as well as real-time communication. The developed platform provides a secure, hardware-independent environment for relay testing, learning, and research. Potential extensions include multi-winding transformer modelling, adaptive restraint schemes, expansion to the logic-design and signal-processing layers, and hardware-in-the-loop (HIL) validation in order to create a complete virtual IED ecosystem.

6.2 Limitations

Despite successfully reproducing several benchmark results, a number of limitations were identified. Most significantly, Multiphysics simulation tool's equation-based modeling framework, while flexible, does not naturally support spontaneous crack nucleation for phase-field fracture. The quench test exposed this limitation clearly: without an initial defect, the phase-field variable remained uniform and failed to evolve into a nucleated crack. In con-

trast, FEniCS, with direct control over the variational forms and iterative history update procedures, handles this instability more robustly.

A second limitation arises from mesh control. Adaptive refinement, a central component of the FEniCS implementation by Hirshikesh et al. [**?**] is not available in Multiphysics simulation tool for custom PDEs. As a result, achieving comparable accuracy requires much finer meshes everywhere, increasing computational cost. Additionally, Multiphysics simulation tool's segregated solver can struggle with the strong nonlinearity in the fully coupled phase-field equations unless carefully tuned.

A third limitation is the restricted access to internal solver routines. While FEniCS allows fine-grained manipulation of residuals, stabilization terms, and weak-form expressions, Multiphysics simulation tool abstracts these details, making certain research-level modifications difficult.

6.3 Future Work

The present work forms the foundation of a digital twin for transformer protection relays by modelling the PSL layer, implementing numerical iDiff protection, integrating Modbus communication, and developing a data-driven black-box model. The next stages will expand the digital twin into a multi-layer representation that more closely mirrors the internal architecture of a numerical IED.

The first major extension is the development of the Logic Design module, which governs how raw protection element outputs are processed before being passed to the PSL. This layer includes level comparison blocks, phase comparison logic, Boolean gating structures, interlocking signals, and conditional pathways. Accurately modelling this layer will enable the digital twin to reproduce intermediate relay decisions, latch behaviour, and protection coordination across different elements.

The second direction involves modelling the Signal Processing layer, which is currently abstracted. This includes low-pass filtering, anti-aliasing, resampling, RMS/phasor extraction, and harmonic estimation implemented at the hardware level. By replicating these algorithms, the digital twin will be able to operate directly on raw CT waveforms and reflect how a real relay handles noise, distortion, and off-nominal frequency conditions.

For the iDiff component, future work will incorporate trip-reason identification, enabling the model to classify trips as internal faults, HS1/HS2 trips, harmonic-blocked conditions, CT-saturation events, external-fault restraints, or polarity/vector-group errors. Enhancements such as more realistic CT saturation modelling, adaptive bias supervision, and extended fault-type coverage will further align the model with modern relay capabilities.

The complete PSL, Logic Design, Signal Processing, and iDiff models will then be integrated into a unified digital-twin framework capable of full end-to-end protection simulation. The final phase will involve Hardware-in-the-Loop (HIL) validation using RTDS and Typhoon HIL platforms, where the digital twin will be tested against real-time power-system waveforms and benchmarked directly against an actual GE MiCOM protection relay. This validation will confirm functional accuracy, dependability, and timing performance, preparing the digital twin for advanced research, training, and future deployment in virtual IED environments.

Acknowledgement

We would like to express our sincere gratitude to our project supervisor, **Dr. Ravi Yadav**, for his constant guidance, valuable insights, and encouraging support throughout the course of this project. His expertise in power system protection and patience in addressing our technical queries played a crucial role in shaping this work and steering it in the right direction.

We are grateful to the **Department of Electrical Engineering, Indian Institute of Technology Jodhpur**, for providing a stimulating academic environment and the necessary facilities to carry out this research. We also acknowledge the support of the faculty and laboratory staff for their cooperation during experimentation and relay testing.

Mansi Choudhary
Mahek Dharmesh Kumar Vanjani

References

[1] GE Vernova Grid Solutions, *P64 MiCOM 5th Generation - Advanced Transformer Protection, Control and Condition Monitoring*, 2025.

[2] Y. Dong, Q. Chen, W. Ding, N. Shao, G. Chen, and G. Li, "State evaluation and fault prediction of protection system equipment based on digital twin technology," *Applied Sciences*, vol. 12, no. 15, p. 7539, 2022.

[3] F. Araujo-Vargas and G. Konstantinou, "Modular design and real-time simulators toward power system digital twins implementation," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 52–61, 2022.