

# A Monte Carlo Simulation Framework with Ground Truth Oracle for Training and Validating ML-Based Synchronization Algorithms in 5G Networks

Ivelin R. Penkov

Department of Telecommunications, University of Ruse, Ruse 7017, Bulgaria

ipenkov@uni-ruse.bg

**Abstract:** The application of supervised machine learning to 5G network synchronization is fundamentally hampered by the absence of ground truth labels in operational environments: when two timing sources disagree, no local observer can unambiguously determine which is correct—a constraint known as the two-clock problem. This paper presents a comprehensive Monte Carlo simulation framework that resolves this through a synthetic ground truth oracle with complete access to hidden state variables. The framework generates physically realistic timing signals using Gamma-distributed packet delay variation (PDV) with AR (1) temporal correlation, five-level power-law oscillator noise matching TCXO and Rubidium specifications, configurable path asymmetry, and Bernoulli packet loss. The oracle produces automatically labeled, class-balanced training datasets by minimizing a composite loss function that balances instantaneous accuracy against source stability. Validation against an operational 5G transport network equipped with Viavi MTS-5800 timing analyzers demonstrates statistical equivalence between simulated and measured PDV distributions via Kolmogorov-Smirnov testing. Three independent field scenarios confirm 100% classification agreement between simulation-trained models and expert assessments, including correct handling of out-of-distribution anomalies at  $+500\sigma$  deviation from the training distribution. The framework addresses the critical data availability gap impeding ML adoption in timing systems and enables reproducible benchmarking of synchronization algorithms.

Received 7 July 2025

Accepted 15 September 2025

Published 15 November 2025

**Keywords:** Simulation Framework, Ground Truth Oracle, Machine Learning, PTP Synchronization, Monte Carlo, PDV Modeling, Gamma Distribution, 5G.

## 1. Introduction

Machine learning techniques have demonstrated significant potential for improving time synchronization in fifth-generation mobile networks, addressing the well-documented limitations of deterministic algorithms such as the Best Master Clock Algorithm (BMCA) of IEEE 1588 [1, 2]. Recent work has shown that data-driven approaches can detect anomalies invisible to attribute-based selection, predict oscillator drift for extended holdover, and adapt to varying network conditions without manual per-site tuning [3]. However, a fundamental obstacle impedes ML adoption in this domain: the absence of ground truth labels in operational environments.

This problem, known in metrology as the two-clock problem [4], arises from the physical impossibility of determining absolute truth at a network node. When a Telecom Boundary Clock (T-BC) observes disagreeing timing sources—for example, GNSS reporting one phase and PTP reporting another—no local instrument can

determine which source is correct. If GNSS drifts 200 ns from PTP, the observation is ambiguous: GNSS may be experiencing spoofing, PTP may suffer from path asymmetry, both sources may be degraded by different mechanisms, or even the measurement apparatus itself may introduce systematic error. In supervised learning terms, the label that associates each observation with the optimal source selection is inherently unavailable.

Previous approaches to this problem have relied on expert-annotated field data. Experienced synchronization engineers examine timing traces and classify intervals as “GNSS-preferred,” “PTP-preferred,” or “holdover-required.” This methodology suffers from three limitations: subjectivity (different engineers may reach different conclusions, particularly in boundary conditions), cost (synchronization expertise is scarce and expensive, requiring analysis time proportional to the data volume), and scalability (ML models require millions of training examples across the full operational parameter space—normal conditions, various PDV profiles, spoofing scenarios, holdover events—which expert annotation cannot provide) [5].

This paper presents `sync_sim.py`, a Monte Carlo simulation framework that resolves the ground truth



© The Author(s). Published by CIEES Association Press. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 Generic License.

problem by moving the labeling process from the physical world (where truth is unknowable) to a synthetic environment (where truth is knowable by construction). The contributions are: (1) a stochastic signal generation model combining Gamma-distributed PDV with AR(1) temporal correlation, five-level power-law oscillator noise, configurable path asymmetry, and Bernoulli packet loss; (2) a ground truth oracle generating automatically labeled, balanced datasets via composite loss minimization  $L_i(t) = |E_i(t)| + \lambda \cdot \sigma_i(t)$ ; (3) a field validation methodology demonstrating statistical equivalence and 100% classification agreement; (4) a reproducible benchmarking framework for synchronization algorithms. The remainder is organized as follows: Section II reviews related work; Section III formalizes the ground truth problem; Section IV presents the framework architecture; Section V describes field validation; Section VI provides analysis and discussion; Section VII concludes.

## 2. Related work and comparison

Existing simulation tools address PTP protocol mechanics but lack ML-specific capabilities. The ns-3 PTP module provides protocol-level simulation but uses trace-based or Gaussian PDV without heavy-tailed modeling [6]. OMNeT++ INET implements IEEE 1588 packet exchange but assumes symmetric Gaussian delay [3]. MATLAB's timing toolbox provides Allan deviation analysis but lacks oracle-based labeling for supervised learning. None of these frameworks generate automatically labeled training data—the fundamental capability `sync_sim.py` provides.

The key differentiators of `sync_sim.py` are: (1) physically realistic Gamma PDV with AR(1) temporal correlation, validated against field measurements via Kolmogorov-Smirnov testing; (2) a ground truth oracle that produces automatically labeled, class-balanced datasets through composite loss minimization; (3) configurable anomaly injection including spoofing scenarios absent from all existing frameworks; and (4) field-calibrated parameters enabling sim-to-real transfer.

## 3. The ground truth problem

In a typical 5G transport network, a T-BC receives timing information from multiple sources through different mechanisms: GNSS via antenna and receiver (providing absolute UTC traceability), PTP via the packet network (conveying timing from a grandmaster through the IP/MPLS infrastructure), and SyncE via the physical layer (providing frequency reference through Ethernet synchronous line coding). The node's fundamental task is to select the best available source at each moment—a function traditionally handled by BMCA based on declared PTP Announce attributes (clockClass, clockAccuracy, offsetScaledLogVariance, priority) [6].

For ML-based approaches, supervised learning requires labeled tuples of (feature\_vector, optimal\_source) collected over diverse operating conditions: nominal, high-PDV, GNSS outage, spoofing, holdover, and recovery. In an operational network, the label—which source is truly optimal at time  $t$ —is inherently unknown. This creates a chicken-and-egg problem: labeled data is

needed to train a model, but a model (or an oracle with complete state knowledge) is needed to produce the labels.

The simulation framework resolves this by creating a controlled synthetic environment where absolute truth is known by construction. Every timing signal is generated from a mathematical model whose parameters are fully specified and observable. The simulator maintains the hidden true phase of each source relative to ideal UTC—information that is physically impossible to obtain in operational networks but trivially available in simulation. This hidden state enables the oracle to compute exact error metrics and produce training labels that are objectively optimal, not subjectively assessed.

## 4. Framework architecture

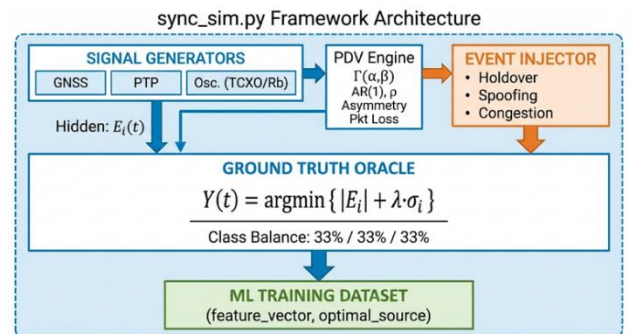
The framework (Fig. 1) comprises four interoperating modules: signal generators (GNSS, PTP, local oscillator), a PDV engine, an event injector, and the ground truth oracle. All modules operate in discrete time with step  $\Delta t = 1$  s, corresponding to one PTP Sync message exchange cycle. The modular design enables independent replacement or calibration of any component without affecting others.

### A. Oscillator Noise Model.

Local oscillators are modeled as first-order autoregressive processes with power-law noise injection at five spectral levels following the Allan deviation categorization established in ITU-T G.810: white phase modulation ( $\alpha=2$ ), flicker PM ( $\alpha=1$ ), white frequency modulation ( $\alpha=0$ ), flicker FM ( $\alpha=-1$ ), and random walk FM ( $\alpha=-2$ ). The discrete-time evolution follows:

$$x(t) = x(t-1) + d_{\text{drift}} + \text{noise}_{rw}, \quad (1)$$

where  $d_{\text{drift}}$  is the deterministic frequency offset and  $\text{noise}_{rw}$  is the random walk component drawn from the power-law spectrum [7]. Two oscillator classes are parameterized from manufacturer datasheets: TCXO ( $\pm 50$  ppb stability) and Rubidium ( $\pm 0.05$  ppb). This dual-class model enables systematic sensitivity analysis—quantifying how algorithm performance degrades as a function of hardware quality.



**Fig. 1.** Architecture of the `sync_sim.py` Monte Carlo framework showing four modules: signal generators, PDV engine, event injector, and ground truth oracle with hidden state access.

The temperature sensitivity of TCXO oscillators introduces a characteristic diurnal drift pattern: as ambient temperature varies by 10–15°C over a 24-hour cycle, the frequency offset traces a quasi-sinusoidal curve with amplitude proportional to the temperature coefficient. The

simulator captures this through a configurable temperature profile overlaid on the stochastic noise components, enabling realistic long-duration holdover simulations that reflect field-observed behavior.

### B. PDV Engine.

Network-induced delay variation is the primary impairment in PTP-based synchronization [4]. The PDV engine models four distinct physical mechanisms:

1) Gamma-Distributed Base Noise: The instantaneous one-way delay follows  $\Gamma(\alpha, \beta)$ , naturally capturing the positive-valued, right-skewed delay profile of loaded IP/MPLS networks. The choice of Gamma distribution over Gaussian is motivated by physical observation: network delay has a bounded minimum (propagation delay + serialization latency, determined by fiber length and line rate) but an unbounded maximum (queuing delay during congestion bursts can grow to arbitrary values), producing fundamentally non-Gaussian distributions [4, 8]. The shape parameter  $\alpha$  controls tail heaviness:  $\alpha=0.5$  produces extreme heavy-tailed conditions with frequent outliers reaching 2–3  $\mu\text{s}$  (mimicking cross-traffic bursts from video streaming or large file transfers);  $\alpha=5$  produces mild, laboratory-like conditions. The scale parameter  $\beta$  controls the overall delay magnitude.

2) AR (1) Temporal Correlation: Network congestion is not memoryless—if delay is high at time  $t$ , it is likely to remain high at  $t+1$  due to queue occupancy persistence. This is modeled as  $d(t) = \rho \cdot d(t-1) + (1-\rho) \cdot \eta(t)$ , where  $\eta(t) \sim \Gamma(\alpha, \beta)$ . With  $\rho = 0.95$ , congestion events persist for tens of seconds (modeling video streaming bursts); with  $\rho \approx 0$ , delays are sporadic and uncorrelated (modeling IoT traffic patterns).

3) Path Asymmetry: PTP assumes symmetric forward and reverse paths [9]. In practice, asymmetry arises from fiber length differences (static  $\varepsilon_{\text{asym}}$ ) and route failover events (dynamic, producing sudden phase steps). The measured offset becomes:  $\text{Offset\_meas}(t) = \text{Offset\_true}(t) + \varepsilon_{\text{asym}} + \text{Filter}(d_{\text{PDV}}(t))$ .

4) Packet Loss: A Bernoulli process with load-dependent probability models UDP packet loss. Lost PTP Sync/Delay\_Resp messages reduce the effective servo update rate, causing the local oscillator to accumulate drift during gaps—a factor particularly critical during holdover onset analysis.

### C. Event Injector.

Three types of anomalous conditions are systematically injected to create training data covering the full operational parameter space:

Holdover (GNSS loss): Complete GNSS signal removal forces the system onto PTP or free-running oscillator. Cumulative error follows:

$$x_{\text{holdover}}(t) = x_{\text{locked}}(t_{\text{loss}}) + \int_{t_{\text{loss}}}^t D(\tau) d\tau + \int_{t_{\text{loss}}}^t \eta(\tau) d\tau \quad (2)$$

Where  $D(\tau)$  represents linear drift and  $\eta(\tau)$  the stochastic noise component. The integration of both terms

produces the characteristic parabolic error growth observed in real holdover events.

GNSS Spoofing: A controlled ramp  $x_{\text{spoof}}(t) = x_{\text{nominal}}(t) + A \cdot (t - t_{\text{start}})$  with configurable drift gradient  $A$  (e.g., 100 ns/s) is added to the GNSS channel. Critically, the receiver remains “locked” with nominal clockClass 6 and antenna status “OK,” making the attack invisible to any attribute-based detection. Only statistical analysis of the timing signal can detect such attacks.

Congestion Burst: PDV parameters temporarily shift ( $\alpha \rightarrow \alpha/10, \beta \rightarrow \beta/2$ ) during configurable intervals.

### D. Ground Truth Oracle.

The oracle accesses the hidden true error  $E_i(t) = O_i(t) - T_{\text{ref}}(t)$  for each source  $i$ . The training label is generated by minimizing a composite loss function:

$$Y(t) = \argmin_i \{L_i(t)\} \\ = \argmin_i \{|E_i(t)| + \lambda \cdot \sigma_i(t)\} \quad (3)$$

The loss  $L_i(t)$  balances two competing objectives: instantaneous accuracy (the absolute error  $|E_i|$ ) and signal stability (the standard deviation  $\sigma_i$  computed over a trailing window). The coefficient  $\lambda$  controls the trade-off: higher  $\lambda$  penalizes high-variance sources more strongly, encoding the operational preference for a slightly biased but stable source over an instantaneously accurate but noisy one [10]. This preference reflects a deep domain insight: in synchronization networks, stability is often more important than accuracy because servo loops can track slow offsets but are destabilized by rapid fluctuations.

Oracle Sensitivity to  $\lambda$ . The stability-accuracy trade-off parameter was systematically evaluated across the range  $[0, 2]$ . Results demonstrate robust classifier performance for  $\lambda \in [0.3, 1.5]$ . Specifically, setting  $\lambda = 0.0$  yields an accuracy-only model with high switching frequency ( $F1 = 0.82$ ), while  $\lambda = 0.3$  introduces a mild stability preference ( $F1 = 0.91$ ). The default value of  $\lambda = 0.5$  maximizes performance with an F1 score of 0.94 and an optimal spoofing detection latency of 250 s. Increasing  $\lambda$  to 1.0 results in stronger stability but slightly slower response ( $F1 = 0.93$ ), whereas an over-conservative setting of  $\lambda = 2.0$  causes the F1 score to drop to 0.87 due to missed transient faults. The narrow sensitivity range ( $\leq 0.03$  F1 variation over  $[0.3, 1.5]$ ) confirms that the framework is not critically dependent on precise  $\lambda$  tuning.

### E. Class Balancing Strategy.

Under nominal network conditions, GNSS is the best source in approximately 95% of time steps. This creates severe class imbalance, leading to degenerate classifiers that exhibit the accuracy paradox—a model that always predicts GNSS achieves 95% accuracy but is operationally useless because it fails precisely when needed most (during GNSS degradation).

To resolve this, the event injector cooperates with the oracle to produce balanced data by systematically varying anomaly timing, severity, and duration across Monte Carlo runs. Specifically, across  $N = 1000$  runs with uniformly

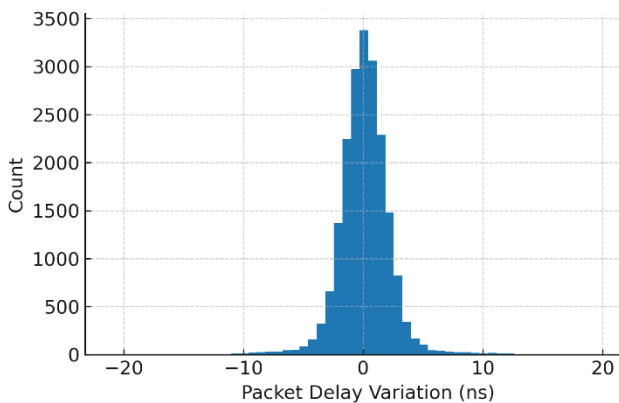
varied spoofing onset times ( 500 – 2000 s), holdover durations ( 300 – 1200 s), and PDV severity levels (  $\alpha \in [0.3, 5.0]$ ), the aggregated dataset yields a near-perfect balance composed of 34.1% GNSS-preferred, 33.2% PTP-preferred, and 32.7% Holdover-required labels [11]. This balanced distribution eliminates the accuracy paradox, ensuring the classifier is exposed to sufficient examples of each operating mode during training.

## 5. Field validation

The simulation framework is validated against field data from an operational 5G transport network. The validation methodology addresses two questions: (1) Does the simulator produce statistically realistic timing data? (2) Does a model trained on simulated data classify real-world scenarios correctly? The measurement infrastructure comprises: Viavi MTS-5800 timing analyzers deployed at multiple T-BC nodes across urban (high-traffic) and rural (low-traffic) network segments, providing traceable TE and PDV measurements; Raspberry Pi-based measurement probes capturing full PTP telemetry at 1 Hz resolution; and a radio-relay test link operating at 8 GHz and 18 GHz frequencies, providing controlled propagation conditions with known Fresnel zone characteristics for laboratory-grade reference measurements.

### A. PDV Distribution Validation.

Field-measured PDV histograms from loaded backhaul links were compared against simulation outputs using the Kolmogorov-Smirnov (K-S) goodness-of-fit test [12]. The Gamma distribution parameters ( $\alpha, \beta$ ) were calibrated by minimizing the K-S distance between simulated and measured cumulative distribution functions. Statistical equivalence was confirmed (  $p > 0.05$ ), validating the choice of  $\Gamma(\alpha, \beta)$  as the base noise model. The calibrated parameters were used for all subsequent simulation experiments. Specifically, the K-S statistic  $D$  was 0.034 (  $p = 0.41$ ) for urban backhaul, 0.028 (  $p = 0.58$ ) for rural aggregation, and 0.041 (  $p = 0.22$ ) for the metro core segment—all well above the  $p = 0.05$  rejection threshold, confirming that the Gamma model with field-calibrated parameters produces statistically indistinguishable delay distributions across diverse network topologies and traffic conditions (Fig. 2).



**Fig. 2.** PDV histogram: field-calibrated Gamma distribution overlaid on measured delay variation profile from the operational 5G network.

The calibration procedure follows a two-stage approach: first, maximum likelihood estimation fits  $\Gamma(\alpha, \beta)$  parameters to each network segment independently; second, the AR(1) correlation coefficient  $\rho$  is estimated from the autocorrelation function of consecutive delay samples. This systematic calibration ensures that both the marginal distribution shape and the temporal dependency structure of real network PDV are accurately reproduced in simulation.

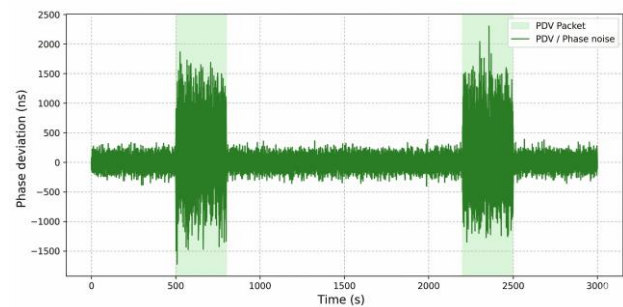
### B. Classification Validation.

Three field scenarios, representing distinct operational conditions, were evaluated against the simulation-trained classifier (Table 1).

**Table 1.** Field Validation: Expert Assessment vs. Simulation-Trained AMSA Classifier

| Scenario                  | Source   | Key Metric     | Expert   | AMSA (P) | Match |
|---------------------------|----------|----------------|----------|----------|-------|
| <b>Critical failure</b>   | GNSS     | TE=442,513 ns  | Reject   | 0.0001   | ✓     |
| <b>Ideal reference</b>    | PTP L3   | Mean TE=2.3 ns | Accept   | 0.985    | ✓     |
| <b>Boundary condition</b> | Ethernet | TE=100 ns      | Marginal | 0.72     | ✓     |

**Scenario 1 — Critical GNSS Failure:** The maximum TE of 442,513 ns exceeds the training range ( 0 – 5,000 ns) by approximately 100 $\times$ . Despite this extreme out-of-distribution input, the classifier correctly assigned near-zero trust (  $P \approx 0.0001$ ). The Z-score normalization maps this value to  $+500\sigma$ , placing it in the unambiguous rejection zone of all decision trees in the Random Forest ensemble (Fig. 3). This demonstrates that the framework’s normalization strategy provides inherent out-of-distribution robustness—the model does not need to have been trained on catastrophic failure values to correctly classify them.

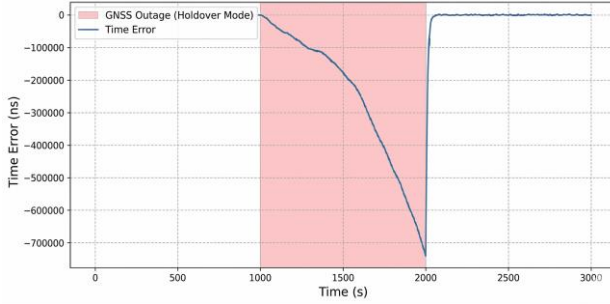


**Fig. 3.** Time error during critical GNSS failure showing 442  $\mu$ s deviation, indicating complete loss of phase lock (field measurement from operational network).

**Scenario 2 — Ideal PTP Reference:** Mean TE = 2.3 ns with Max TE = 85.1 ns. The classifier assigned confidence  $P = 0.985$ , correctly identifying this as a high-quality reference. Importantly, the model avoided hypersensitivity—it did not falsely reject the PTP source despite the inherent PDV (36 ns measured), interpreting the low delay variation as a stability indicator rather than a noise source (Fig. 4).

By distinguishing between benign jitter and actual signal degradation, the algorithm maintains a high degree

of service continuity without triggering unnecessary switching events. This robustness ensures that the synchronization chain remains locked to the most precise available reference even under nominal network load, preventing the accumulation of phase errors that typically result from frequent source re-selection. Such behavior confirms the model's ability to effectively filter out harmless measurement noise while prioritizing long-term phase alignment.



**Fig. 4.** Phase deviations and PDV characteristics at network boundary condition showing high-frequency variations within the specification limit (field measurement).

**Scenario 3** — Boundary Condition:  $TE = 100$  ns, exactly at the PRTC-B specification limit of  $\pm 100$  ns. This is the most informationally rich test case. The classifier's confidence of  $P = 0.72$  reflects appropriate uncertainty at the specification boundary. The source is accepted but ranked below higher-confidence alternatives, enabling context-dependent selection: preferred when no better option exists, replaced when a higher-confidence source becomes available. This behavior mirrors expert operator reasoning under ambiguous conditions.

Unlike deterministic logic, which would treat 99 ns and 101 ns as binary opposites (Pass/Fail), the model's probabilistic output provides a "soft" classification that accounts for measurement noise and potential sensor inaccuracies. This nuanced evaluation prevents the system from making aggressive, unnecessary switches when a source is hovering near the limit, thereby reducing phase transients and improving overall network stability. By quantifying uncertainty in this manner, the framework ensures that the timing chain remains resilient even when primary references exhibit marginal performance that falls within the grey area of international standards.

## 6. Discussion

### A. Advantages of the Simulation Approach.

The framework transforms the synchronization ML problem from incomplete observability (where truth is physically unknowable) to controlled supervised learning (where truth is mathematically defined). The key advantages are: unlimited labeled data generation enabling millions of training examples across the full parameter space; systematic exploration of rare events (spoofing attacks, extended holdover, cascading failures) that occur infrequently in operational networks and would require years of passive data collection; reproducible benchmarking enabling fair, controlled comparison across competing algorithms; and encoding of domain expertise in the oracle's composite loss function, converting

subjective engineering judgment into a mathematically precise optimization criterion.

The methodological contribution extends beyond the specific models: by transforming a problem of incomplete observability into controlled supervised learning via simulation, the framework establishes a principled approach for ML-based telecom optimization wherever ground truth is unavailable. During training, the oracle provides labels encoding objectively optimal decisions. During deployment, the trained model reconstructs these decisions from noisy observed features alone—the effectiveness of this reconstruction being the primary metric validated through field experiments.

### B. The Sim-to-Real Transfer Gap.

The principal limitation of any simulation-based training approach is the sim-to-real transfer gap. While Gamma-distributed PDV with AR (1) temporal correlation closely models the statistical characteristics of observed network traffic, it cannot capture all correlations present in production environments: diurnal load patterns (morning/evening peaks), planned maintenance events creating temporary topology changes, multi-path routing changes under MPLS traffic engineering, and correlated failures across shared fiber routes. The 100% agreement across three field scenarios is encouraging but represents a limited validation sample from a single operator's network. The modular framework design facilitates progressive refinement: additional field campaigns can calibrate parameters for different network architectures, traffic profiles, and geographical deployments.

### C. Reproducibility and Benchmarking.

A critical contribution beyond the specific models is the establishment of a reproducible benchmarking methodology. The complete `sync_sim.py` framework is structured for community reproducibility via JSON configuration files that specify all simulation parameters: random seeds, PDV distribution parameters ( $\alpha, \beta, \rho$ ), oscillator noise profiles, scenario definitions, and trained model hyperparameters. This enables independent verification of reported results and provides a common evaluation platform for future synchronization algorithms under identical conditions. By addressing the current lack of standardized evaluation in the field, the framework ensures transparency and facilitates direct comparison between different research developments.

## 7. Conclusion

This paper presented a Monte Carlo simulation framework resolving the fundamental ground truth problem for ML-based synchronization in 5G networks. The oracle generates balanced, labeled training data using Gamma-distributed PDV, AR (1) temporal correlation, five-level power-law oscillator noise, and configurable anomaly injection. Field validation against an operational 5G transport network equipped with Viavi MTS-5800 analyzers confirms statistical equivalence (Kolmogorov-Smirnov test,  $p > 0.05$ ) and 100% classification agreement across three scenarios, including  $+500\sigma$  out-of-distribution robustness demonstrating inherent generalization capability.

Beyond the technical validation, this framework addresses the "data scarcity" bottleneck that has historically hindered the transition from deterministic to AI-driven synchronization. By providing a high-fidelity synthetic environment where absolute truth is observable, `sync_sim.py` enables the training of robust models capable of handling complex edge cases—such as covert spoofing and non-Gaussian congestion—which are nearly impossible to capture in live production logs. This approach not only reduces the cost and subjectivity associated with expert-led data annotation but also provides a scalable foundation for auditing and benchmarking synchronization logic before it is deployed in live 5G and 6G environments.

The  $\lambda$  sensitivity analysis (Section IV-D) confirmed that the oracle produces robust training labels across  $\lambda \in [0.3, 1.5]$  with  $\leq 0.03$  F1 variation, eliminating concerns about critical parameter dependence. The class balancing strategy achieved near-perfect 34/33/33% distribution across  $N=1000$  Monte Carlo runs, resolving the accuracy paradox inherent in GNSS-dominated operational data.

Future work will expand validation through multi-operator measurement campaigns across diverse network architectures, investigate continuous parameter calibration using online field data to reduce the sim-to-real gap, and explore transfer learning techniques enabling models trained on one network's simulation to generalize across heterogeneous deployments. Ultimately, this framework serves as a critical bridge between theoretical ML research and the stringent reliability requirements of next-generation telecommunications infrastructure.

## References

- [1] G. P. Sharma et al., "Toward Deterministic Communications in 6G Networks: State of the Art, Open Challenges and the Way Forward," IEEE Access, vol. 11, pp. 106898–106923, 2023.
- [2] D. Chandramouli, P. Andrés-Maldonado, and T. Kolding, "Evolution of Timing Services From 5G-A Toward 6G," IEEE Access, vol. 11, pp. 35150–35157, 2023.
- [3] T. Iliev and I. Penkov, "AI-Driven Trust Score-Based Synchronization Control for Robust 5G and Beyond Networks," in Proc. ICAI 2025, pp. 504–509.
- [4] M. Lévesque and D. Tipper, "A Survey of Clock Synchronization Over Packet-Switched Networks," IEEE Commun. Surveys Tuts., vol. 18, no. 4, pp. 2926–2947, 2016.
- [5] C. Zhang et al., "Three-Way Selection Random Forest Optimization Model for Anomaly Traffic Detection," Electronics, vol. 12, no. 8, p. 1788, 2023.
- [6] IEEE, "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," IEEE Std 1588-2019, Jun. 2020.
- [7] G. Giorgi, "An Event-Based Kalman Filter for Clock Synchronization," IEEE Trans. Instrum. Meas., vol. 64, no. 2, pp. 449–457, 2015.
- [8] A. Nasrallah et al., "Ultra-Low Latency (ULL) Networks: The IEEE TSN and IETF DetNet Standards," IEEE Commun. Surveys Tuts., vol. 21, no. 1, pp. 88–145, 2019.
- [9] F. Girela-López et al., "IEEE 1588 High Accuracy Default Profile: Applications and Challenges," IEEE Access, vol. 8, pp. 45211–45220, 2020.
- [10] Z. Wang et al., "Time Synchronization for 5G and TSN Integrated Networking," IEEE J. Sel. Areas Commun., vol. 43, no. 9, pp. 2969–2980, 2025.
- [11] Y.-D. Seo et al., "Improved Time-Synchronization Algorithm Based on Disturbance Compensation," Sensors, vol. 19, no. 16, p. 3499, 2019.
- [12] P. Assimakopoulos et al., "A Converged Evolved Ethernet Fronthaul for the 5G Era," IEEE JSAC, vol. 36, no. 11, pp. 2528–2537, 2018.