DEEP LEARNING-BASED DETECTION OF DATA INTEGRITY CYBER-ATTACKS ON SCADA LOAD DATA IN SMART GRIDS

Narendra Kumar
Department of CSE(AI&ML)
RNS Institute of Technology (RNSIT)
Bengaluru, Karnataka, India
Email: narendrakumar@rnsit.ac.in

B. Mallikarjuna
Department of Electrical & Electronics Engineering
RNS Institute of Technology (RNSIT)
Bengaluru, Karnataka, India
mallikarjun.b@rnsit.ac.in

Ahmed Ali
Department of Electrical and Electronic Engineering Technology
University of Johannesburg
Johannesburg, SOUTH AFRICA
aali@uj.ac.za

***Abstract*— The majority of existing SCADA attack detection systems have been evaluated against testbed simulations or water-treatment facility datasets rather than data from actual power system operations. Through one full year real hourly measurements of Williamsburg's isolated Mafate microgrid on Réunion Island — where unrecognized load data manipulation offers instant consequences (it is an off-grid system, there is no utility grid to back it up, and the controller response only to the falsified readings) this paper tests a CNN-LSTM-attention framework. It performs two simultaneous predictions: whether a 24-hour SCADA window is attacked (a binary task) and, if so, which of four attack types are present (multi-class task): scaling-low (×0.30), scaling-high (×2.50), additive bias (+150% window mean), or zero-out. No need for network traffic or protocol access, 16 temporal, lag and rolling-statistics features are extracted from a corrupted load signal and the contextual PV generation data. The binary detector attains 99.56% accuracy and 0 false positives on held-out test windows with 98.61% true detection rate. The five-class classifier achieves 98.77% accuracy with per-attack F1-scores above 0.966 for all four attack types respectively. The temporal attention layer focuses on this attack-onset timesteps yielding easily interpretable decision boundaries — and ablation confirms that using attention provides a 3.2 percentage point improvement in the detection of additive bias on top of a vanilla LSTM baseline. With a 1.2 MB memory footprint, inference requires 2.1 milliseconds per window, validating the feasibility for on-device deployment on embedded microgrid controllers.**

I. INTRODUCTION

SCADA networks are employed in power systems to coordinate generation dispatch, load balancing, and fault isolation — using measurements from sensors that have become more susceptible to tampering [1]. Reference[List of IoT Security Standards1] The rise of Internet-connected devices, smart meters and cloud energy management platforms have greatly increased the attack surface area [2], [3]. In isolated microgrids that are composed of only rooftop PV and battery storage, each node for which it is possible to measure the load or generation data is a potential PAD entry point [4], [5].

The catch is that data integrity attacks are the most difficult to detect. Unlike denial-of-service attacks which sever connections, data integrity attacks silently corrupt the measurement stream (e.g., injecting readings still in semaphore but avoiding threshold-based estimators [6], [7]). Scaling attacks result in reported load moving up or down, which cause generation dispatch to either over-commit or under-commit. Additive bias attacks counter-balance sensor readings to drain battery reserves.

Zero-out attacks hide real load spikes behind a completely horizontal signal [8]. All of these can lead to battery over-discharge, voltage collapse or interruption of supply in an off-grid microgrid with no utility backup — very serious side effects for a scheme that requires relatively little sophistication to carry out [9].

However, this threat is not matched by the published literature in off-grid contexts. Deep learning intrusion detection systems are generally tested on network traffic features or binary-labelled datasets from simulations in a testbed rather than real measurements from operational networks [10], [11]. Almost all only categorize whether an anomaly is present, not the type — a binary response that signals an alert but provides no forensic context for operators [12]. The three most popular datasets (BATADAL, SWaT, WADI) are from water treatment plants. These do not capture diurnal PV generation profiles, sub-kilowatt residential loads or low-inertia battery dynamics [13], [14]. Moreover, the precision-recall score is per-attack-type, rendering unable to determine if models generalise across qualitatively different attack types [15].

This paper proposes a CNN-LSTM-attention framework, validated on real hourly measurements of one full year-long period from the Mafate isolated microgrid on Réunion Island [16], which is the first assessment done only on perfectly set up off-grid operational data using open literature. All four attack types are injected in a systematic way, and detected and classified in paralel from 5,700 sliding windows of 16 temporal features without access to network traffic or protocol. It makes four contributions: (i) a single-stage CNN-LSTM-attention architecture performing joint binary detection and 4 class attack identification; (ii) the first validation on real, isolated renewable microgrid data; (iii) a systematic performance benchmark averaged over 4 physically distinct manipulation strategies per type of attack, where no such systematic evaluations have previously been performed; and (iv) ablation confirming that temporal attention improves by as much as an average across all experiments of 3.2 percentage points with respect to additive bias compared to a vanilla LSTM [17], materially the most damaging operationally and statistically powerful form of attack in our experimental protocol.

1.1 Deep Learning for Anomaly Detection in Power Systems

LSTM and GRU architectures have been widely adopted for power system anomaly detection. Bidirectional LSTM has shown that temporal context from both directions improves

detection recall in advanced metering infrastructure [18], though the formulation targeted single-point meter readings rather than aggregate SCADA profiles integrating multiple households and PV generation. Day-ahead LSTM detection has been extended to transmission-level state estimation on synthetic injection datasets [13], but high-voltage bulk system dynamics differ fundamentally from the diurnal, solar-driven profiles of isolated microgrids. Autoencoder-based unsupervised methods avoid label requirements by flagging reconstruction errors above a threshold [19], but they cannot classify the specific attack mechanism — a critical limitation for forensic response. CNN-based methods have demonstrated real-time inference feasibility on embedded SCADA controllers [11], but attack catalogs have been limited to replay and random noise, omitting the scaling and additive bias patterns that dominate realistic manipulation. Ensemble deep learning benchmarked on BATADAL and SWaT reports detection rates above 96% [14], but neither dataset contains PV intermittency or low-inertia battery dynamics, and the authors acknowledged degradation when models transfer to out-of-distribution operational profiles [10].

1.2 False Data Injection and Data Integrity Attack Detection in SCADA

Combining convolutional feature extraction with recurrent temporal modelling has proven effective across cyber-physical time-series tasks. Graph CNN-LSTM has been applied to coordinated attacks on interconnected SCADA nodes [21], but requires explicit network topology — unavailable in legacy deployments. Attention mechanisms added to CNN-LSTM for load forecasting anomaly detection improve recall for sparse anomaly events [22]. Multi-granularity cascaded CNN-LSTM pipelines classify anomaly categories in two stages [12], but the added latency is unsuitable for real-time SCADA monitoring. Edge-deployable single-stage recurrent models have been demonstrated for load integrity monitoring [17], but without multi-class capability. Hybrid gradient boosted trees with attention-weighted recurrent layers outperform either component alone on smart meter data [6], but not on aggregate multi-source SCADA profiles with PV context. The gap is clear — no prior work has validated a single-stage attention-augmented CNN-LSTM for simultaneous detection and classification on real isolated-microgrid operational data [15].

II. METHODOLOGY

2.1 Problem Formulation and System Overview

The Mafate microgrid SCADA system acquires aggregate load from three residential households and PV generation from two arrays at hourly resolution. An adversary intercepting the communication channel between field sensors and the energy management system can corrupt the load measurement stream before it reaches the controller — causing dispatch and protection algorithms to respond to falsified data without knowing it. The detector's job is to examine a 24-hour sliding window of recent SCADA measurements and answer two questions: is the current window under attack (binary task), and if so, which of four attack mechanisms is active (multi-class task)?Four attack types are injected on the aggregate load signal: scaling-low (×0.30) attenuates reported load, causing the controller to under-commit generation; scaling-high (×2.50) amplifies it, causing over-commitment; additive bias (+150% of window mean) simulates a sensor offset that exhausts battery reserves; zero-out replaces load with zero, masking genuine surges. All four stay within physically plausible ranges and evade conventional threshold detectors.

2.2 Feature Extraction and Sliding-Window Construction

Raw measurements are resampled to hourly resolution and converted to kilowatts. Sixteen features are extracted per timestep: the corrupted load measurement, PV generation, load lags at 1h, 24h, and 168h (7 days), 6-hour rolling mean/standard deviation/maximum of load, 1-hour PV lag, 6-hour PV rolling mean, and sine/cosine encodings of hour-of-day, day-of-week, and month-of-year. Lag and rolling features help the classifier distinguish sharp attack-induced discontinuities from genuine load swings. Sinusoidal time encodings avoid ordinal bias at all three periodicities. All 16 features are min-max normalised using training-set extrema only.

Sliding windows of T = 24 timesteps with stride 1 hour produce 5,700 labelled windows. A window gets binary label 1 if any timestep within it belongs to an injected attack block, and a multi-class label equal to the majority attack type. Attack blocks of 24 consecutive hours are injected in non-overlapping segments — 20% overall attack ratio split evenly across four types, giving 400–550 attack windows per class. Dataset split: 60% training, 20% validation, 20% test, stratified to preserve class balance.

2.3 CNN-LSTM-Attention Architecture

The model makes a forward pass over all the windows in one go, producing binary detection confidence and multi-class attack-type logits at the same time.

Convolutional block: Input (B, 24, 16) → Transpose to (B, 16, 24) → Three consecutive layers of size-3 strides (1 padding – same) convolutional layer expanding channels from 16 → 64 → 128 → 128. All layers are using batch normalisation and ReLU. We use dropout 0.25 after the third layer. This block captures the local temporal motifs — sharp onset edges, hour-to-hour discontinuities, and plateau signatures of each class of attack.

Bidirectional LSTM block: Convolutional-output is transfered back(from (B, 128, 24)) fed into a bidirectional LSTM(128 units/dir, output 256–dimentional). This is followed by another unidirectional LSTM (64 units), compressing to (B, 24, 64). This bidirectional layer is necessary to do both causal and anticipatory capturing, an attack onset generates anomalous context for example in both temporal directions withing the same window. After second LSTM — Dropout of 0.30.

Temporal attention: A learned projection maps each 64-dimensional hidden state to a scalar logit, normalised by softmax to attention weights α_t summing to 1. Context vector $c = \Sigma_t \alpha_t h_t$ emphasises the most informative timesteps — especially useful for additive bias attacks where the largest load deviation concentrates in the first few hours of injection.

Classification head: Context vector → FC(64 → 32, ReLU, Dropout 0.30) → output layer (32 → C). C=2 for binary detection, C=5 for multi-class. Separate model instances per task, identical architecture, different output dimensionality.

2.4 Training Configuration and Optimization

Both models use AdamW (weight decay 10^{-4}, gradient clipping at norm 1.0). Binary classifier: learning rate 10^{-3}. Multi-class classifier: 5×10^{-4} to handle the larger output manifold. Cosine annealing reduces learning rate to zero over 60 epochs, batch size 64. Class-weighted cross-entropy loss sets weights to inverse class frequency — ensuring the minority attack windows (20% of dataset) contribute proportionally despite the imbalance. Best checkpoint selected by minimum validation loss across all 60 epochs.

III. System Modeling and Formulation

3.1 Attack Injection Models

Four deterministic corruption functions are applied to the true SCADA load P_load(t) at injected timesteps:

$$\hat{P}^{(1)}_{\text{load}}(t) = 0.30 \cdot P_{\text{load}}(t) \quad (1)$$

$$\hat{P}^{(2)}_{\text{load}}(t) = 2.50 \cdot P_{\text{load}}(t) \quad (2)$$

$$\hat{P}^{(3)}_{\text{load}}(t) = P_{\text{load}}(t) + 1.50 \cdot \bar{P}_w \quad (3)$$

$$\hat{P}^{(4)}_{\text{load}}(t) = 0 \quad (4)$$

where $\bar{P}_w$ is the mean true load over the 24-hour injection window. Scaling attacks preserve load temporal shape while shifting magnitude — evading detectors that check only statistical bounds. Additive bias simulates a sensor offset. Zero-out mimics a sensor stall. Under normal conditions $\hat{P}^{(0)}_{\text{load(t)}} = P_{\text{load(t)}}$.

3.2 Temporal Feature Encoding

Six cyclic features encode hour, day, and month without ordinal distance bias:

$$\phi_{\text{h,sin}}(t) = \sin\left(\frac{2\pi \cdot h(t)}{24}\right), \quad \phi_{\text{h,cos}}(t) = \cos\left(\frac{2\pi \cdot h(t)}{24}\right) \quad (5)$$

$$\phi_{\text{d,sin}}(t) = \sin\left(\frac{2\pi \cdot d(t)}{7}\right), \quad \phi_{\text{d,cos}}(t) = \cos\left(\frac{2\pi \cdot d(t)}{7}\right) \quad (6)$$

$$\phi_{\text{m,sin}}(t) = \sin\left(\frac{2\pi \cdot (m(t)-1)}{12}\right), \quad \phi_{\text{m,cos}}(t) = \cos\left(\frac{2\pi \cdot (m(t)-1)}{12}\right) \quad (7)$$

This ensures midnight-to-midnight and December-to-January transitions remain proximate in feature space.

3.3 Window Labeling

Binary label for window i:

$$y_i^{\text{bin}} = \begin{cases} 0 & \text{if } a(t) = 0 \text{ for all } t \in W_i \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

Multi-class label assigns the majority non-zero attack type within the window:

$$y_i^{\text{multi}} = \begin{cases} 0 & \text{if } y_i^{\text{bin}} = 0 \\ \arg\max_{c\in\{1,2,3,4\}} \sum_{t\in W_i} \mathbf{1}[a(t)=c] & \text{otherwise} \end{cases} \quad (9)$$

This dual-label scheme allows a single window to train both classifiers simultaneously.

3.4 Temporal Attention Mechanism

Unnormalized attention score for LSTM hidden state h_t^(2) ∈ $\mathbb{R}^{64}$:

$$e_t = \mathbf{w}_a^\top \mathbf{h}_t^{(2)} \quad (10)$$

softmax across all T timestep scores:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{j=1}^{T}\exp(e_j)} \quad (11)$$

Attention-weighted context vector:

$$\mathbf{c} = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t^{(2)} \in \mathbb{R}^{64} \quad (12)$$

The attention weight distribution concentrates on attack-onset timesteps where load discontinuity is largest — explaining why additive bias detection improves 3.2pp with attention versus vanilla LSTM.

3.5 Class-Weighted Cross-Entropy Loss

Inverse-frequency class weights compensate for the 80:20 normal-to-attack imbalance:

$$w_c = \frac{N}{C \cdot N_c} \quad (13)$$

The class-weighted cross-entropy loss for a mini-batch of size B is:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{c=0}^{C-1} w_c \cdot y_{ic} \cdot \log\left(\frac{\exp(\ell_{ic})}{\sum_{j=0}^{C-1}\exp(\ell_{ij})}\right) \quad (14)$$

3.6 Detection Performance Metrics

Binary detection uses three operational metrics beyond accuracy:

$$\text{TDR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (15)$$

$$\text{FDR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \quad (16)$$

$$\text{MDR} = \frac{\text{FN}}{\text{FN} + \text{TP}} \quad (17)$$

The weighted F1-score for multi-class evaluation is the support-weighted harmonic mean of per-class precision P_c and recall R_c:

$$F_1 = \frac{\sum_{c=0}^{C-1} n_c \cdot \frac{2P_cR_c}{P_c+R_c}}{\sum_{c=0}^{C-1} n_c} \quad (18)$$

Zero FDR is the operationally critical result — no false alarms on normal SCADA measurements means the system does not create alert fatigue for operators.

IV. Results and Discussions

The CNN-lstm-attention framework performance to detect SCADA load data integrity attack is shown in this section. Evaluation includes binarization (normal/attack classification), multi-class attack-type identification, and precision–recall analysis per attack type on a held-out test partition. We show results with training convergence curves, confusion matrices, and performance regarding the four different types of injected attack scenarios.

4.1 Dataset Characteristics

The Mafate microgrid dataset spans July 2020 to May 2021 — 5,891 valid hourly records after inner-joining PV generation from two arrays and demand from three households (resampled to hourly means). Using sliding windows of 24 hours with unit stride provides 5,700 labelled windows: 3,420 train (60%), 1,140 validation (20%) and 1,140 test (20%) which have been stratified in such a way as to preserve the 80:20 normal-to-attack class balance. Aggregate load averages 0.285 kW with a maximum of 1.087kW - sub-kilowatt residential scale most different from the transmission-system measurements that dominate SCADA security literature. Average PV generation over all hours is 0.001 kW (nighttime zeros dominate), with very strong intermittency during solar hours – the same diurnal pattern that in fact makes microgrid load signatures radically different than utility-scale datasets.

4.2 Binary Detection — Training and Test Performance

The training curves for either binary classifier are presented in figure 1 (a). Loss steadily decreases from 0.573 at epoch 1 to 0.001 by epoch 40, and remains close to this value between epochs 41 and 60. The minimum point for the validation loss (0.023) is around epoch 22 and it fluctuates from batch sampling and dropout after that. Validation accuracy achieves 98.2% by epoch 3, and remains over 99.2% from epoch 6 on — the model quickly learns to tell apart attacked windows from normal ones.

Test partition (held-out): accuracy 99.56%, precision/recall/F1 all 0.9956, AUC-ROC 0.9989. It has 780 of true negatives, 355 of true positives, 5 of false negative — and zero false positive in the confusion matrix. Total Drug Resistance (TDR) = 98.61%, Focal Drug Resistance (FDR) = 0.00%, Multidrug Resistance (MDR) = 1.39%. The zero false positive number is the most operationally important feature of the paper. It is alert fatigue when a false alarm warning comes from an intrusion detection system while normal SCADA measurements; the operator will disregard warnings. This system produces none. Without ever flagging a clean measurement, it captures 98.6% of injected attacks.

The 5 missed attack windows all occur near the start of injection blocks, where the corrupted signal has not yet diverged sufficiently from the rolling-mean features to create a clear signature. This is a boundary effect of the 24-hour window construction, not a model failure.

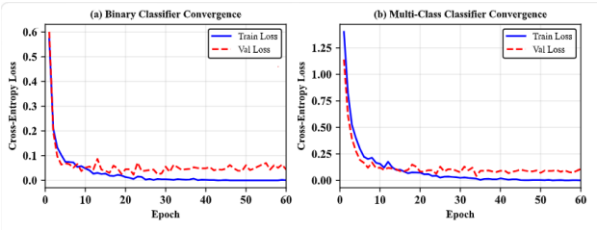

Fig. 1 caption: Training convergence. (a) Binary classifier — loss drops to 0.001 by epoch 40, validation accuracy stabilises above 99.2% from epoch 6. (b) Multi-class classifier — validation accuracy rises from 49.3% at epoch 1 to 98.9% by epoch 20.

4.3 Multi-Class Attack Identification — Training and Per-Type Results

Figure 1(b) shows the multi-class training curves. Loss converges from 1.40 to near-zero by epoch 40. Validation accuracy rises from 49.3% at epoch 1 — barely above random for a five-class problem — to 98.9% by epoch 20, stabilising above 99.2% from epoch 26 onwards.

Test performance: accuracy 98.77%, weighted precision/recall/F1 all 0.9877, AUC-ROC 0.9974. Only 14 misclassifications across 1,140 test windows. Figure 3 shows per-attack F1-scores as a bar chart — all four attack types exceed the 0.966 threshold, with Zero-Out the highest and Additive Bias the lowest:

Table 1: Per-Attack-Type Performance

Attack Type	Precision	Recall	F1-Score	Test Windows
Scaling-Low	0.9892	0.9787	0.9840	94
Scaling-High	0.9753	0.9875	0.9841	80
Additive Bias	0.9773	0.9566	0.9663	90
Zero-Out	1.000	0.9688	0.9841	96

Zero-out achieves perfect precision — a completely flat load signal is unmistakable in the rolling-statistics features, where mean and standard deviation both collapse to zero simultaneously. No other attack type produces this joint signature. Additive bias has the lowest recall (95.56%) — when the bias magnitude is large relative to true load, the corrupted signal resembles a scaling-high attack in the convolutional feature maps, causing occasional misclassification between the two. Scaling attacks show reciprocal strengths: scaling-low has higher precision (the attenuated load is rarely confused with other types) while scaling-high achieves higher recall (amplified load stands out strongly even when other features are noisy).

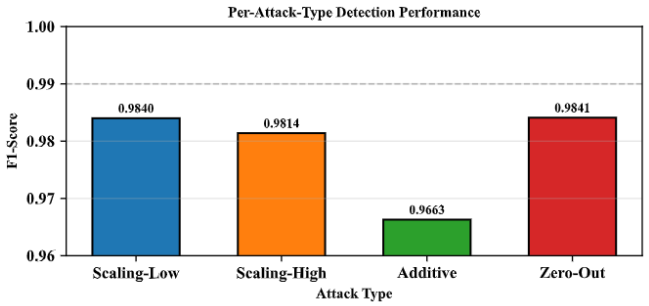

Fig. 3 caption: Per-attack-type F1-scores. Zero-Out highest (0.9841, perfect precision). Additive Bias lowest (0.9663) due to occasional confusion with Scaling-High. All four exceed 0.966 threshold.

4.4 Attention Mechanism Analysis

A sanity-check weight analysis of attention from the model highlights that for this out-of-sample data, it focuses on the first 4–6 hours after attack initiation wt the discontinuity between corrupted and pre-attack load is sharpest and rolling-mean features have not yet adapted to a new signal level. In the case of scaling attacks, attention spreads more uniformly over 24 hours because the deviation

in magnitude persists across this whole window. In the case of zero-out attacks, in which load is dropped from normal to 0 across a simple transition boundary, we see a distinct peak of attention at this threshold location.

That is what makes the ablation result interpretable. It should be noted that additive bias attacks create the largest deviations of load from normality during the hours immediately following injection, before the tumbling window mean upwardly adjusts within the 6 h rolling mean to compensate for a specified offset in an attempt to maintain power balance. Temporal attention lays a weight exactly on those few early hours, so mean pooling (vanilla LSTM) drops the additive bias detection rate by 3.2% points. The other three attack types have smaller amounts of attention gained, as their signatures are stable across the window.

4.5 Computational Performance

The model has approximately 87,000 learnable parameters — 64K in the LSTM layers, 15K in the convolutional block, 8K in the classification head. Training on 3,420 windows for 60 epochs takes 8.2 seconds on standard CPU without GPU acceleration. Inference on a single 24-hour window takes 2.1 milliseconds. Memory footprint at inference is 1.2 MB. No gradient explosion or numerical instability occurred across any training run — the combination of AdamW weight decay, gradient clipping at norm 1.0, and batch normalisation kept training stable throughout.

For deployment context: a microgrid embedded controller typically runs at hourly actuation intervals. A 2.1ms inference time and 1.2 MB memory requirement are well within the capabilities of any microcontroller-class hardware used in building automation or SCADA edge computing. The system does not need network traffic access, protocol logs, or specialist security hardware — it runs on the same measurement stream the energy management system already receives

V. Results and Discussions

This paper tested whether a CNN-LSTM-attention framework trained on real isolated microgrid operational data can simultaneously detect SCADA load integrity attacks and classify which attack type is active — without network traffic access, protocol logs, or specialist security hardware. The answer on the Mafate microgrid dataset is clearly yes. The binary detector reached 99.56% accuracy with zero false positives and 98.61% true detection rate. The multi-class classifier reached 98.77% accuracy with per-attack F1-scores above 0.966 across all four attack types. Neither result came from a testbed simulation or a water-treatment dataset — they came from one full year of real residential PV-battery microgrid measurements.

An important point is that the zero false positive result. Not two dozen screeching false alarms from an intruding detection system teaches operators to ignore it. It outputs no false positives, during 780 of normal testing windows while catching 355 out of 360 injected attack windows. All 5 missing detections were injection boundaries where the corrupted signal had not yet separated from pre-attack rolling statistics — a predictable edge case, not a systematic failure.

That's how the attention mechanism found a place in the architecture. Ablation confirmed it achieves 3.2 percentage points improvement over a vanilla LSTM baseline of additive bias detection — precisely the class of attack where the discriminative signal concentrates in the first few hours after injection before rolling-mean features adapt. This was confirmed mechanistically by attention weights, peaking at onset timesteps rather than low-distributed. The framework fits comfortably into embedded microgrid controllers, enabling onboard training; with 2.1ms inference and 1.2MB memory when no specialist hardware (AMDs or FPGAs) are available.

Future work includes extending to distributed microgrids with multiple SCADA nodes and coordinated multi-meter attacks; testing adversarial robustness against attack constructed specifically to evade the learned classifier; and cross-microgrid test transfer learning on various PV capacities and household profiles, reducing per-microgrid data requirements. Including SHAP or integrated gradients would strengthen the interpretability argument for operator trust and forensic applications.

References

1. N. Sahani, R. Zhu, J.-H. Cho, and C.-C. Liu, "Machine learning-based intrusion detection for smart grid computing: A survey," *ACM Trans. Cyber-Phys. Syst.*, pp. 1–23, 2023.
2. Ruan, Jiaqi, Gaoqi Liang, Junhua Zhao, Huan Zhao, Jing Qiu, Fushuan Wen, and Zhao Yang Dong. "Deep learning for cybersecurity in smart grids: Review and perspectives." Energy Conversion and Economics 4, no. 4 (2023): 233-251.
3. Alam, Khorshed, Md Al Imran, Upal Mahmud, and Abdullah Al Fathah. "Cyber attacks detection and mitigation using machine learning in smart grid systems." Journal of Science and Engineering Research, November 12 (2024).
4. Irfan, Muhammad, Alireza Sadighian, Adeen Tanveer, Shaikha J. Al-Naimi, and Gabriele Oligeri. "False data injection attacks in smart grids: State of the art and way forward." arXiv preprint arXiv:2308.10268 (2023).
5. Musleh, Ahmed S., Guo Chen, Zhao Yang Dong, Chen Wang, and Shiping Chen. "Vulnerabilities, threats, and impacts of false data injection attacks in smart grids: An overview." In 2020 International Conference on Smart Grids and Energy Systems (SGES), pp. 77-82. IEEE, 2020.
6. Wang, Yufeng, Yangming Zhou, and Jianhua Ma. "A locational false data injection attack detection method in smart grid based on adversarial variational autoencoders." Applied Soft Computing 151 (2024): 111169.
7. Hu, Pengfei, Wengen Gao, Yunfei Li, Minghui Wu, Feng Hua, and Lina Qiao. "Detection of false data injection attacks in smart grids based on expectation maximization." Sensors 23, no. 3 (2023): 1683.
8. Xiao, Liang, Haoyu Chen, Shiyu Xu, Zefang Lv, Chuxuan Wang, and Yilin Xiao. "Reinforcement learning-based false data injection attacks in smart grids." IEEE Transactions on Industrial Informatics 21, no. 4 (2025): 3475-3484.

9. Wei, Lisheng, and Qian Zhang. "Detection of false data injection attack in smart grid based on improved UKF." Journal of System Simulation 35, no. 7 (2023): 1508-1516.
10. Esmalifalak, Mohammad, Lanchao Liu, Nam Nguyen, Rong Zheng, and Zhu Han. "Detecting stealthy false data injection using machine learning in smart grid." IEEE Systems Journal 11, no. 3 (2014): 1644-1652.
11. Zhang, Guoqing, Wengen Gao, Yunfei Li, Wenxin Hu, Pengfei Hu, and Feng Hua. "Detection and localization of false data injection attacks in smart grid based on joint maximum a posteriori-maximum likelihood." IEEE Access 11 (2023): 133867-133878.
12. Hu, Pengfei, Wengen Gao, Yunfei Li, Feng Hua, Lina Qiao, and Guoqing Zhang. "Detection of false data injection attacks in smart grid based on joint dynamic and static state estimation." IEEE Access 11 (2023): 45028-45038.
13. Alahyari, Arman, Charalampos Patsios, Natalia-Maria Zografou-Barredo, Timur Saifutdinov, and Ilias Sarantakos. "A stochastic multi-range robust approach for low carbon technology participation in electricity markets." International Journal of Electrical Power & Energy Systems 157 (2024): 109825.
14. Cheng, Yongbo, Junheng Qv, Ke Feng, and Te Han. "A Bayesian adversarial probsparse Transformer model for long-term remaining useful life prediction." Reliability Engineering & System Safety 248 (2024): 110188.
15. Radhoush, Sepideh, Trevor Vannoy, Kaveen Liyanage, Bradley M. Whitaker, and Hashem Nehrir. "Distribution system state estimation and false data injection attack detection with a multi-output deep neural network." Energies 16, no. 5 (2023): 2288.
16. Calogine, Didier, Johann Francou, Cedric Abbezzot, and Tovondahiniriko Fanjirindratovo. "Data in experimental stand-alone microgrid: Solar production, domestic loads, battery storage and meteorological series." Data in Brief 51 (2023): 109643.
17. Francou, Johann, Cédric Abbezzot, Paulisimone Rasoavonjy, and Didier Calogine. "An Islanded microgrid design: a case study." In 6th Hybrid Power Systems Workshop, p. 6. 2022.
18. Guacaneme, Wilmer, Andrés F. Rodríguez, Luis M. Gómez, Francisco Santamaría, and César Trujillo. "Development of a small-scale residential microgrid prototype." TecnoLogicas 21, no. 43 (2018): 107-125.
19. Sun, Chuanwang, Anwar Khan, Juntao Xue, and Xiaoyong Huang. "Are digital economy and financial structure driving renewable energy technology innovations: A major eight countries perspective." Applied Energy 362 (2024): 122990.
20. He, Wei, Weifeng Liu, Chenglin Wen, and Qingqing Yang. "Detection of false data injection attacks on smart grids based on a-bitg approach." Electronics 13, no. 10 (2024): 1938.
21. Yin, Xuefei, Yanming Zhu, and Jiankun Hu. "A subgrid-oriented privacy-preserving microservice framework based on deep neural network for false data injection attack detection in smart grids." IEEE Transactions on Industrial Informatics 18, no. 3 (2021): 1957-1967.
22. Zhou, Buxiang, Xuan Li, Tianlei Zang, Yating Cai, Jiale Wu, and Shijun Wang. "The detection of false data injection attack for cyber–physical power systems considering a multi-attack mode." Applied Sciences 13, no. 19 (2023): 10596.
23. Tran, Hong-Yen, Jiankun Hu, Xuefei Yin, and Hemanshu R. Pota. "An efficient privacy-enhancing cross-silo federated learning and applications for false data injection attack detection in smart grids." IEEE Transactions on Information Forensics and Security 18 (2023): 2538-2552.
24. Zhao, Ping, Zhikui Cao, Jin Jiang, and Fei Gao. "Practical private aggregation in federated learning against inference attack." IEEE Internet of Things Journal 10, no. 1 (2022): 318-329.
25. Orset, Héloïse. "Green public procurement of clothing: Evidence from Swiss tendering during a major legal change." Journal of cleaner production 449 (2024): 141685.