ELECTRICITY THEFT DETECTION USING DEEP REINFORCEMENT LEARNING IN SMART POWER GRIDS

GUIDED BY:

Mr. Harikrishnan G.R
Asst. Professor
Dept. of CSE

PRESENTED BY:

Shadma Subair CP
S7 CSE-B
MES20CS102

Acknowledgement

This presentation have my consent and approval to be presented.

Student name and signature:

Shadma Subair CP

S7 CSE-B

Mentor's name and signature:

HARIKRISHNAN N R

18/10/23.

OVERVIEW

01 Introduction

02 Related works

03 What are the preliminaries used for this

04 How data can be prepared

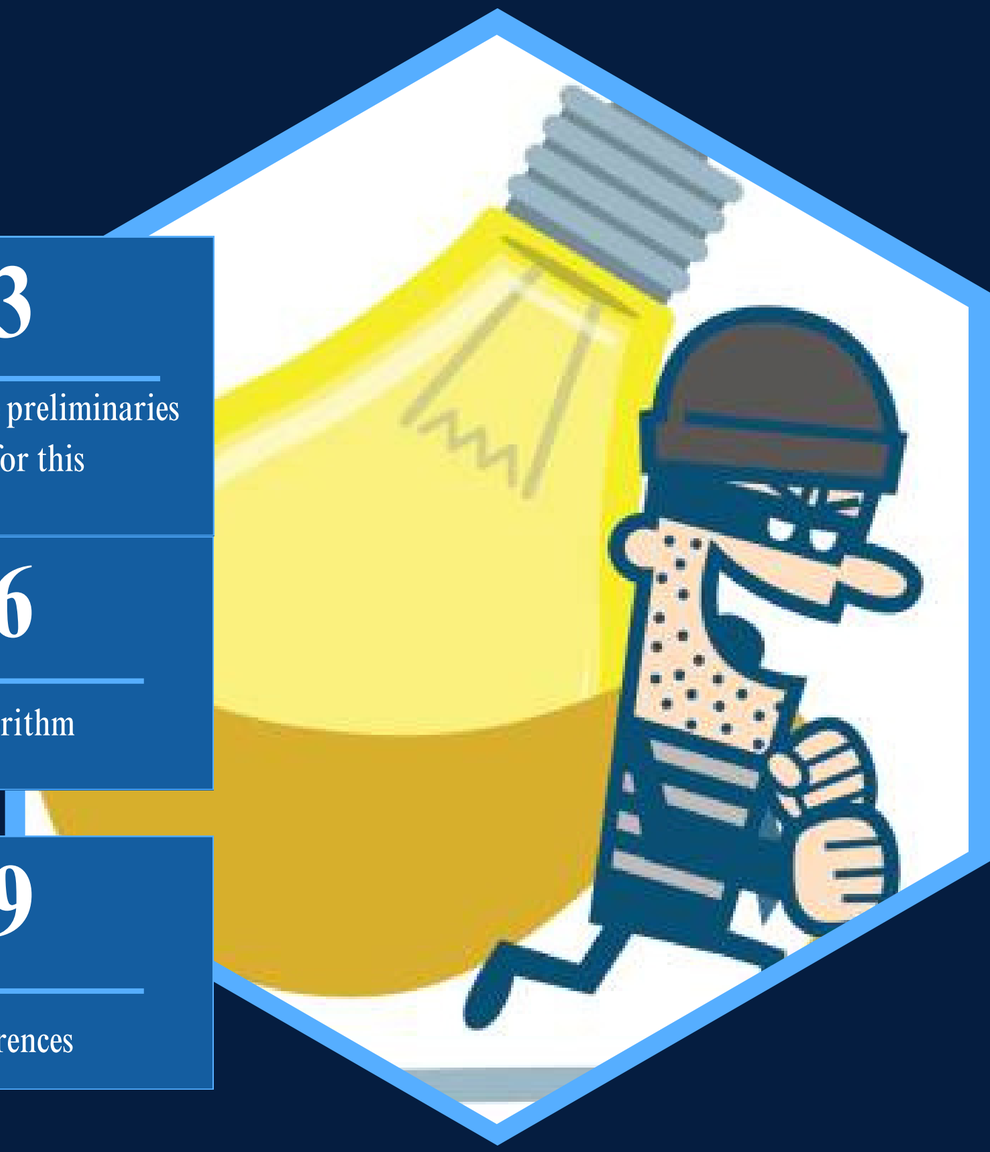
05 The proposed RL models

06 Algorithm

07 How to evaluate performance

08 Conclusion

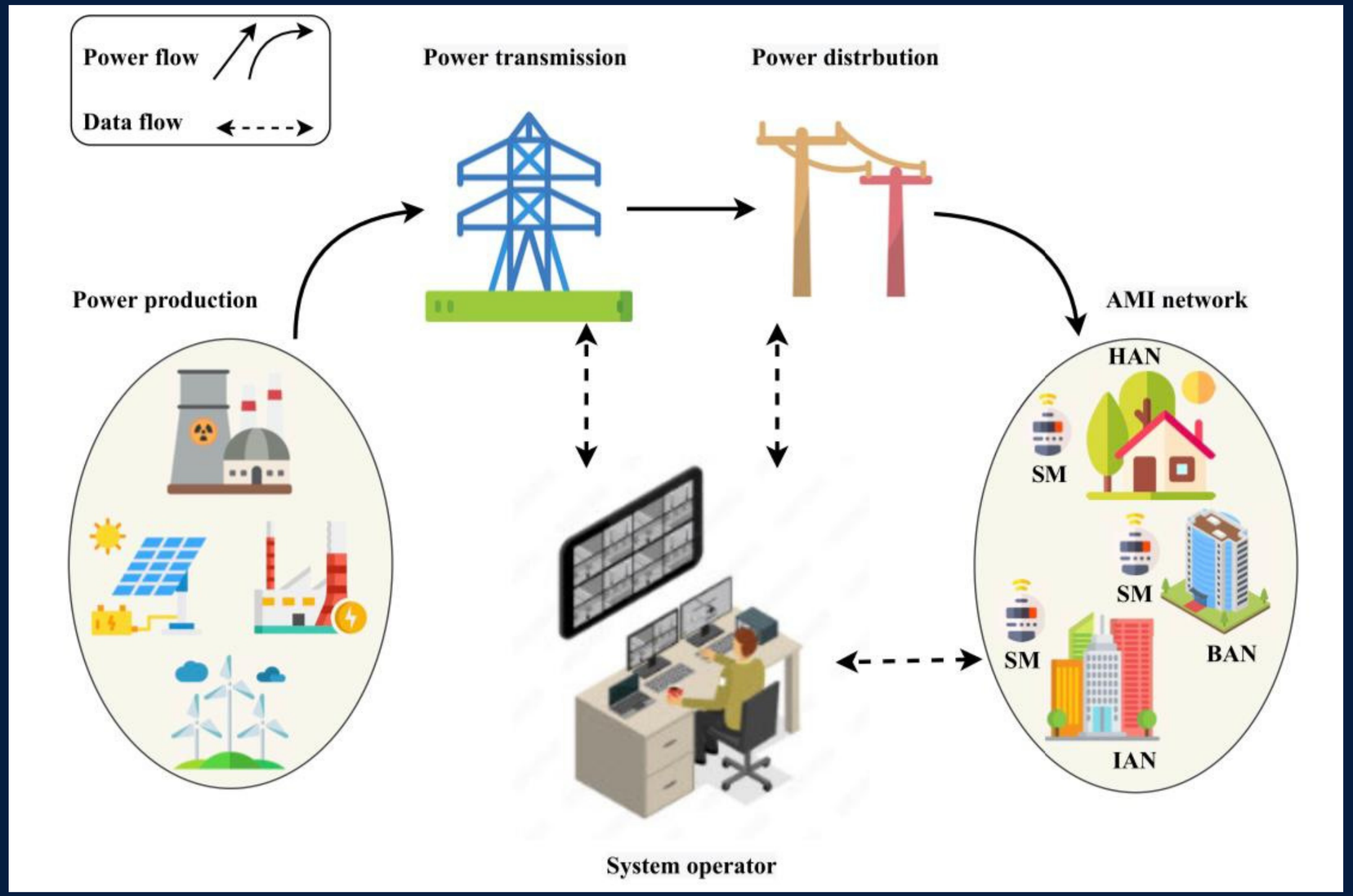
09 References



INTRODUCTION

- The Smart Grid (SG) is an advanced system for efficient electricity delivery and monitoring.
- Advanced Metering Infrastructure (AMI) enables communication between Smart Meters (SMs) and the System Operator (SO). Electricity theft via SM tampering is a major issue globally.
- AI and ML have been used for theft detection but have limitations.
- It's tested in four scenarios: a general detection model, a customized one, adapting to changing usage patterns, and defending against new attacks.
- This study represents a novel application of RL in cybersecurity.

Smart grid model architecture



RELATED WORKS

Related works

To detect electricity theft, multiple methods have been proposed for the detection of electricity theft. These methods can be categorized as

1. **Hardware-based methods,**
2. **Statistical and game theory methods,**
3. **Data-driven methods.**

WHAT ARE THE PRELIMINARIES USED FOR THIS?

A. REINFORCEMENT LEARNING (RL)

- RL = Trial-and-error learning.
- Agent interacts with environment (s, a, r).
- Goal: Maximize cumulative rewards.
- Sequential: s_t -> a_t -> s_{t+1}.
- Value function Vπ(s) evaluates performance.
- Optimal policy π^* , $V^*(s)$ maximizes rewards.

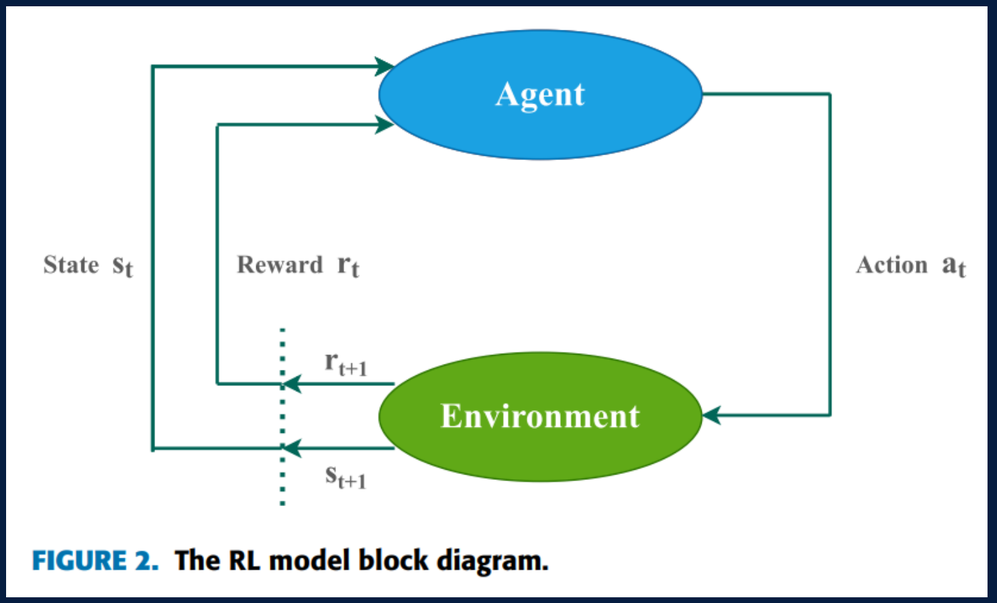


FIGURE 2. The RL model block diagram.

B. DEEP LEARNING (DL)

- Deep Learning (DL): Uses multi-layered neural networks for applications like voice and face recognition.
- Manuscript Objective: Detecting electricity theft through DL classification of consumption data. Detecting electricity theft by classifying consumption readings using DL architectures like FFNNs, GRUs, and CNNs.
- DL Model Training: Involves optimizing parameters (weights and biases) using labeled data samples.

1. FEED-FORWARD NEURAL NETWORK (FFNN)
2. CONVOLUTIONAL NEURAL NETWORK (CNN)
3. GATED RECURRENT UNIT (GRU) NEURAL NETWORK

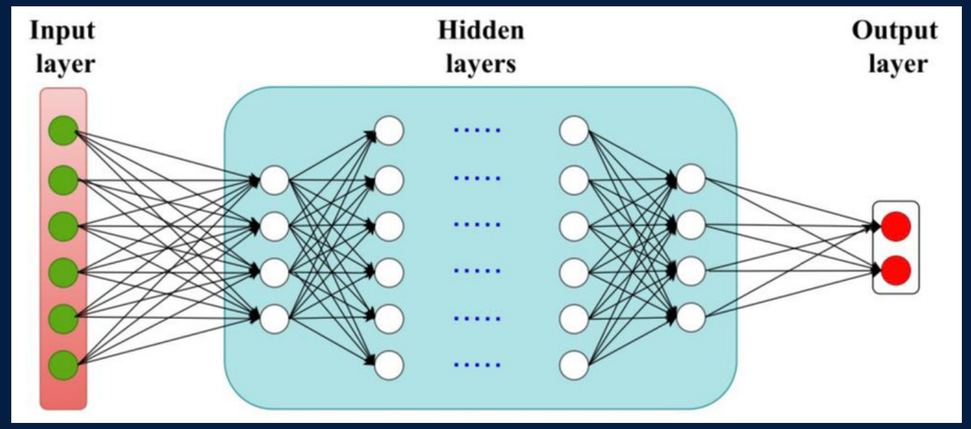


FIGURE 3. The typical architecture of FFNN

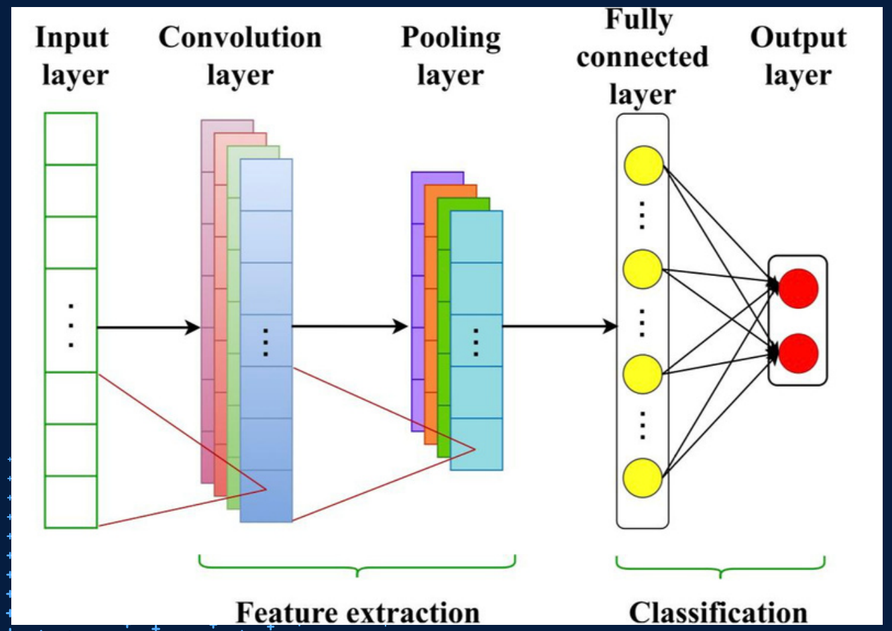


FIGURE 4. The typical architecture of CNN

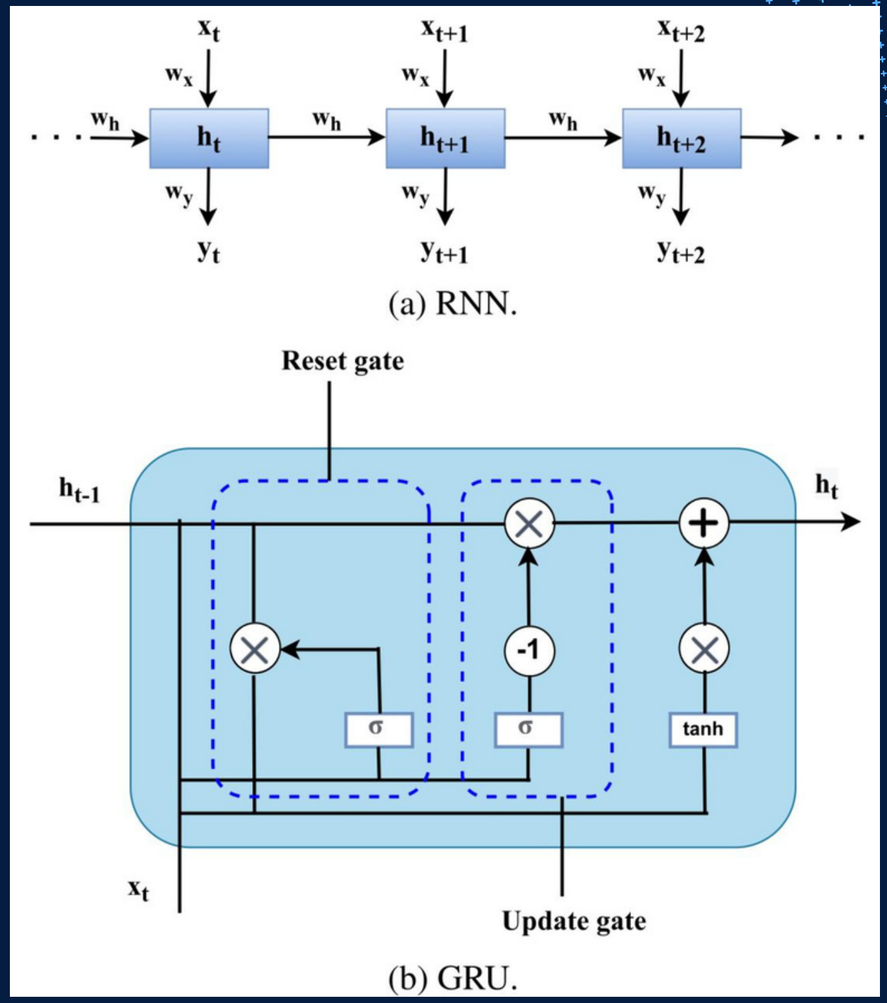


FIGURE 5. The typical architecture of (a) RNN and (b) GRU.

HOW DATA CAN BE PREPARED?

A. BENIGN SAMPLES

1. Paper uses Irish smart energy dataset for training and evaluation.
2. Dataset from Electric Ireland and Sustainable Energy Authority (2012).
3. Focuses on 130 randomly selected customers, 69,680 benign samples with detailed daily readings.

TABLE 1. Irish dataset characteristics.

Description of data	Value
Data consumption time frame	536 days
No. of customers	3600+
Fine-grained interval	30 Min
No. of readings per day (R)	48
No. of the considered customers	130

B. MALICIOUS SAMPLES

1. Detector uses cyber-attacks to imitate electricity theft.
2. Attacks modify benign samples to create malicious ones.
3. Six attack targets:
   - Reducing true consumption randomly.
   - Dynamically reducing it over time.
   - Reporting mean consumption values.
   - Dynamically reducing it by a time-varying factor
   - Reporting zero consumption for a period.
   - Reporting higher consumption during low-price periods.
4. These attacks help test the detector's effectiveness.

TABLE 2. Cyber attack functions.

Attack No	Attack function
1^{st}	$f_1(x_i(t)) = \beta x_i(t)$
2^{nd}	$f_2(x_i(t)) = \beta_t x_i(t)$
3^{rd}	$f_3(x_i(t)) = mean(x_i)$
4^{th}	$f_4(x_i(t)) = \beta_t mean(x_i(t))$
5^{th}	$f_5(x_i(t)) = \begin{cases} 0 & t \in [t_s, t_e] \\ x_i(t) & t \notin [t_s, t_e] \end{cases}$
6^{th}	$f_6(x_i(t)) = x_i(R - t)$

C. DATA PROCESSING

- Attack functions are applied to generate malicious samples with parameters β, βt, ts, and te.
- The initial dataset is imbalanced, with a 1:6 ratio of benign to malicious samples.
- ADASYN is used to oversample the minority class, resulting in a balanced dataset of 836,160 samples. It's split into 2:1 for training and testing, with 557,440 training samples and 278,720 testing samples.

THE PROPOSED RL MODELS

1. Implements mini-batch training, utilizing samples with features and labels representing the environment state and action for effective model training.
2. Constructs mini-batches through random sampling of the training dataset, comprising sequential samples organized by time slot index, including the current time slot state, action, and the subsequent time slot state for each sample.

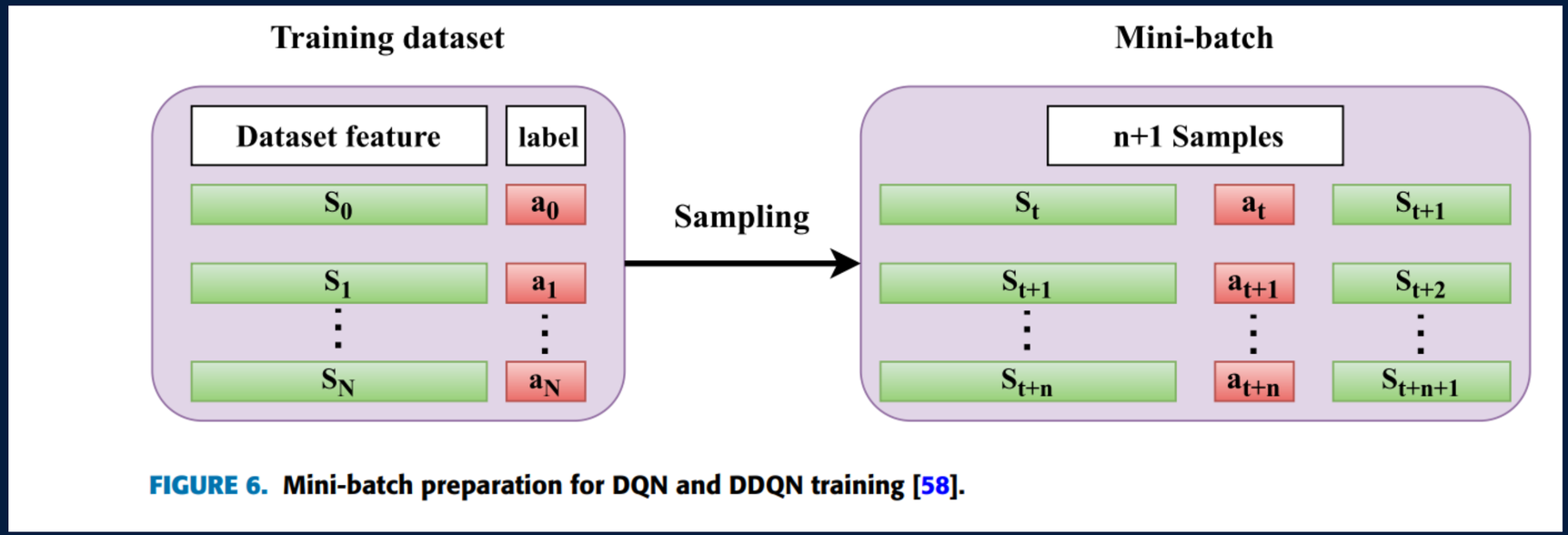


FIGURE 6. Mini-batch preparation for DQN and DDQN training [58].

A. DEEP Q NETWORK (DQN)

1. DQN model is employed to compute the value of the Q function.
2. FFNN, CNN, GRU, and hybrid CNN+GRU architectures, training with mean square error loss. Rewards are determined through comparisons between predicted and actual labels, with the model leveraging multiple available labels for each state prediction.

B. DOUBLE DEEP Q NETWORK

1. DDQN model with two neural networks - current and target - for Q function predictions, featuring delayed parameter updates for the target network.
2. Current network is used to predict the Q function of the current state,Qt.
3. Target network is used to predict the Q function of the next state,Qt+1.

ALGORITHM

DDQN Training Algorithm

Input: Training epochs T, batch size B, exploration rate ϵ, discount factor γ, and learning rate α.
Output: The optimal action a^*.

Initialize the action value function $Q(s, a)$ arbitrarily.
Initialize the state s by sampling the training dataset randomly.
for $i = 0, 1, 2, \ldots, T$ **do**
  for each state s in i. **do**
    Input the state s_t and the actions set A in the current network in order to predict $Q(s, A)$ for all actions.
    Use the ϵ-greedy policy to select the action $\hat{a}_t$.
    Given s_t and $\hat{a}_t$, obtain $Q(s_t, \hat{a}_t)$.
    Calculate the reward r_t.
    Input the next state s_{t+1} and the actions set A in the target network in order to predict $Q(s_{t+1}, A)$ for all actions.
    Use $\arg\max_a Q(s_{t+1}, A)$ policy to select $\hat{a}_{t+1}$.
    Given s_{t+1} and $\hat{a}_{t+1}$, obtain $\hat{Q}_{t+1}(s_{t+1}, \hat{a}_{t+1})$.
    Using $\hat{Q}_{t+1}$, r_t, and γ, Obtain Q_{ref}.
    Calculate the loss function.
    Update the Q-value $Q(s_t, a_t)$.
    Repeat until s_{t+1} is terminal.
  end for
  Repeat until getting to epoch T.
end for
Compute the optimal policy π^* and optimal action a^*.
Execute the optimal action a_t^* at current time slot t.

HOW TO EVALUATE PERFORMANCE ?

1. Four scenarios were conducted to evaluate the performance of the proposed detection approaches: (a) a global RL model utilizing DQN and DDQN with various neural network architectures, (b) a customized model based on DDQN for new customers, (c) updates to the customized model to adapt to changes in consumption patterns of current customers, and (d) handling the challenge of learning new cyber-attacks using the global model.
2. The parameters of the proposed DRL detection schemes are adjusted and given in TABLE 3

TABLE 3. Parameters of DRL schemes.

Parameter	Value
No. of training epochs (T)	10
Batch size (B)	128
Exploration rate (ϵ)	0.6
Discount factor (γ)	0.001
Learning rate (α)	0.00001

A. PERFORMANCE EVALUATION METRICES

The proposed detection approaches are evaluated using metrics like accuracy, precision, recall, false alarm, false negative rate, highest difference, and F-1 score, based on the confusion matrix.

The core elements of the confusion matrix and are defined as follows:

- TP
- TN
- FP
- FN

1.ACCURACY

$$ACC(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100.$$

2. PRECISION

$$Precision(\%) = \frac{TP}{TP + FP} \times 100.$$

3. RECALL

$$Recall(\%) = \frac{TP}{TP + FN} \times 100.$$

4. FALSE ALARM (FA)

$$FA(\%) = \frac{FP}{FP + TN} \times 100.$$

5. FALSE NEGATIVE RATE (FNR)

$$FNR(\%) = \frac{FN}{FN + TP} \times 100.$$

6.HIGHEST DIFFERENCE (HD)

$$HD(\%) = Recall(\%) - FA(\%).$$

7. F-1 SCORE

$$F1(\%) = \frac{2 * Precision * Recall}{Precision + Recall} \times 100.$$

B. EXPERIMENTAL RESULTS OF SCENARIO 1

1. Performance evaluation of global detectors includes RL-based, DQN-RL-based, and DDQN-RL-based models, with CNN and hybrid (CNN+GRU) architectures showing better training performance than FFNN.
2. The hybrid architecture (CNN+GRU) excels by combining CNN's feature extraction and GRU's input correlation capture, outperforming other architectures.
3. DDQN-RL-based hybrid (CNN+GRU) detectors stand out as the top performers, making them the preferred choice for subsequent scenarios due to their continuous target network updates and avoidance of the moving target issue.

TABLE 4. The parameters of the FFNN detection model.

Architecture	Parameters		
	Layer	Number of units	AF
FFNN	Input	48	Linear
	Dense	512	Relu
	Dense	700	Relu
	Dense	850	Relu
	Dense	1024	Relu
	Dense	512	Relu
	Dense	256	Relu
	Dense	200	Relu
	Dense	50	Relu
	Output	2	Softmax

TABLE 5. The parameters of the CNN detection model.

Architecture	Parameters		
	Layer	Number of units	AF
CNN	Input	48	Linear
	Conv1D	32	Relu
	Conv1D	64	Relu
	Conv1D	128	Relu
	Dense	64	Relu
	Dense	128	Relu
	Dense	256	Relu
	Dense	256	Relu
	Dense	512	Relu
	Dense	2	Softmax

TABLE 7. The parameters of the hybrid (CNN+GRU) detection model.

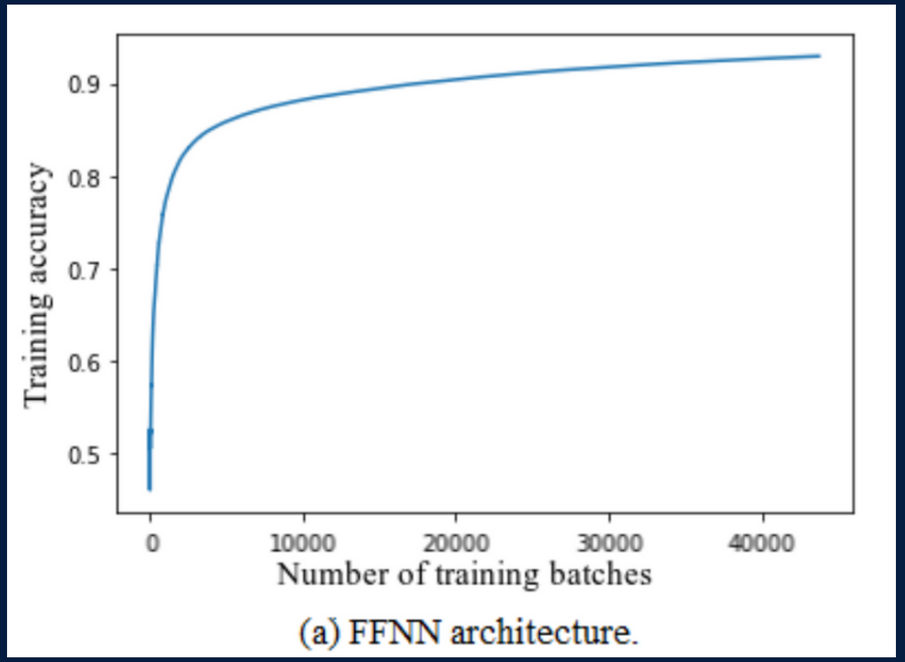
Architecture	Parameters		
	Layer	Number of units	AF
CNN+GRU	Input	48	Linear
	Conv1D	64	Relu
	Conv1D	64	Relu
	Conv1D	128	Relu
	GRU	64	Tanh
	GRU	64	Tanh
	GRU	64	Tanh
	Dense	64	Relu
	Dense	128	Relu
	Dense	2	Softmax

TABLE 6. The parameters of the GRU detection model.

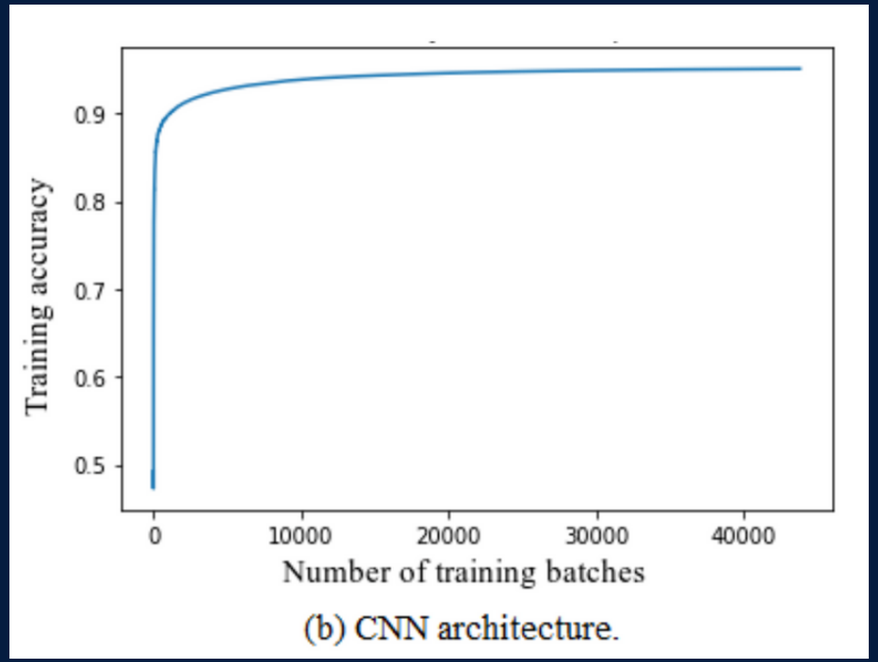
Architecture	Parameters		
	Layer	Number of units	AF
GRU	Input	48	Linear
	GRU	64	Sigmoid
	GRU	64	Tanh
	Dense	64	Relu
	Dense	128	Relu
	Dense	2	Softmax

TABLE 8. Comparison between the performance of DL, DQN-RL, and DDQN-RL global models.

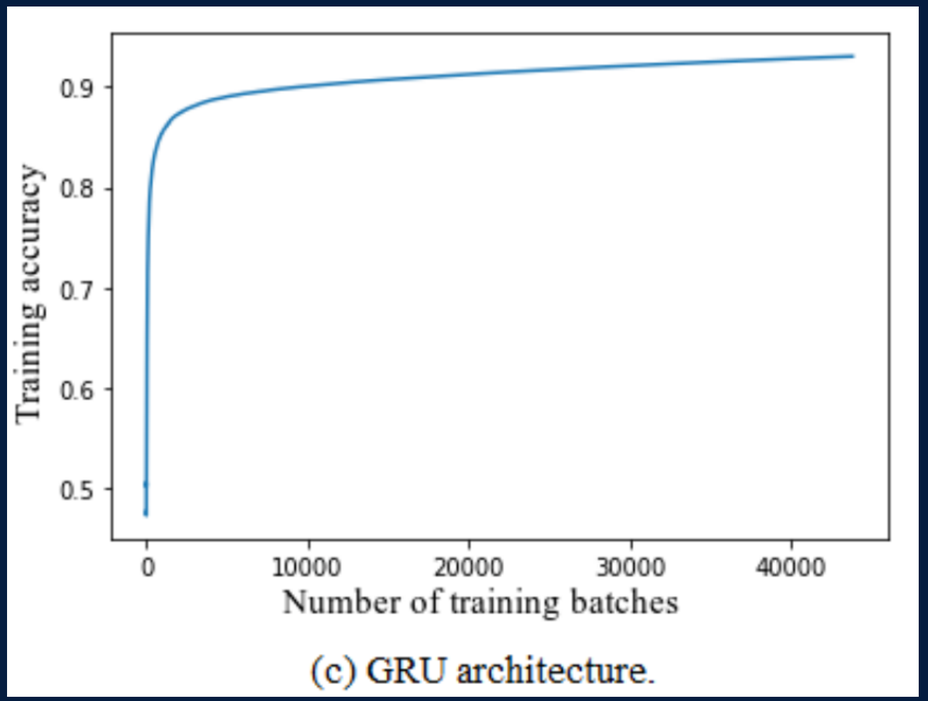
Metrics	DL				DQN-RL				DDQN-RL			
	FFNN	CNN	GRU	CNN+GRU	FFNN	CNN	GRU	CNN+GRU	FFNN	CNN	GRU	CNN+GRU
ACC (%)	92.42	93.14	91.10	94.71	95.02	95.84	95.84	96.84	94.63	95.22	95.82	97.33
Precision (%)	92.41	92.78	91.67	93.68	95.10	95.93	95.90	96.89	94.86	95.50	95.94	97.38
Recall (%)	92.40	93.52	90.38	95.84	95.02	95.84	95.84	96.84	94.63	95.22	95.82	97.33
FA (%)	7.56	7.23	8.17	6.42	4.47	2.99	3.75	2.68	2.43	1.37	2.48	2.06
FNR (%)	7.59	6.47	9.62	4.15	5.48	5.32	4.57	3.64	8.30	8.16	5.86	3.27
HD (%)	84.85	86.30	82.21	89.43	90.55	92.86	92.09	94.16	92.20	93.86	93.35	95.27
F1 (%)	92.40	93.15	91.02	94.75	95.06	95.89	95.87	96.86	94.74	95.36	95.88	97.35



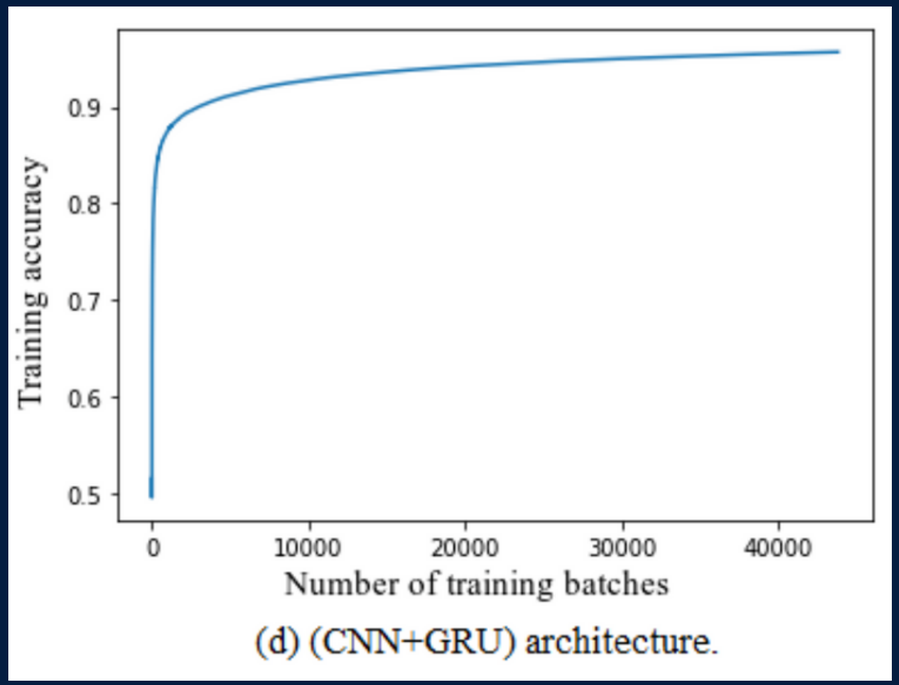
(a) FFNN architecture.



(b) CNN architecture.



(c) GRU architecture.



(d) (CNN+GRU) architecture.

C. EXPERIMENTAL RESULTS OF SCENARIO 2

1. New customers create customized DDQN-based detection models using their electricity data.
2. Customized models outperform the global model with higher accuracy and lower false alarms.
3. Training accuracy improves with more batches, as shown in Figure 10.

TABLE 9. Comparison between the performance of the global and customized models for different customers.

Metrics	Global model	Customized models		
		Customer 14	Customer 30	Customer 48
ACC (%)	97.332	99.439	98.102	99.502
Precision (%)	97.388	99.441	98.119	99.507
Recall (%)	97.332	99.439	98.102	99.502
FA (%)	2.06	0.25	0.93	0.14
FNR (%)	3.27	0.86	2.88	1.00
HD (%)	95.27	99.19	97.17	99.36
F1 (%)	97.355	99.440	98.110	99.504

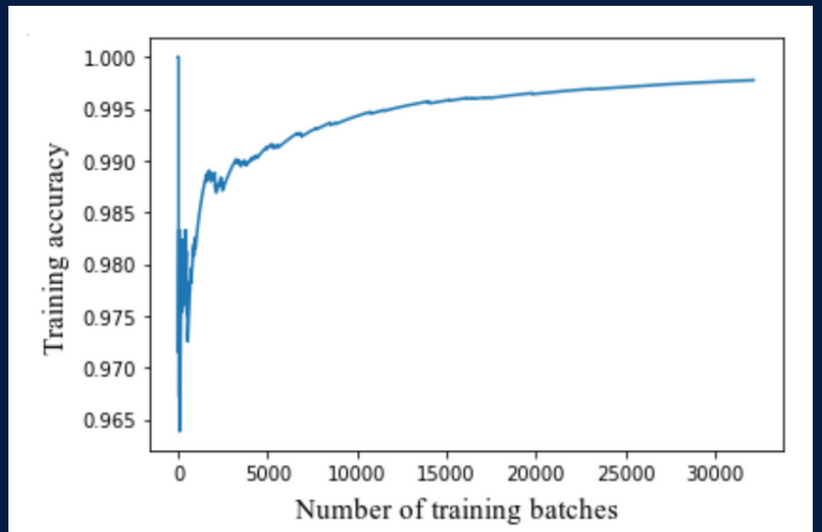


FIGURE 10. Training accuracy of DDQN-based hybrid(CNN+GRU) customized model for customer 14.

D. EXPERIMENTAL RESULTS OF SCENARIO 3

1. When a customer's electricity consumption pattern changes, the DDQN-based customized detection model is retrained with new data, updating it sample by sample.
2. The updated customized detection models outperform the old models for various customers, with improvements of up to 20% in ACC and up to 55% in HD, while FA and FNR decrease by up to 33% and 26%, respectively.
3. Comparing the updated models to the original models reveals closer performance alignment across different customers, showcasing the effectiveness of the model updates in adapting to changing consumption patterns.

TABLE 10. The performance of old and updated customized models for different customers due to changing their consumption behavior.

Metrics	Old customized model			Updated customized models		
	Customer 14	Customer 30	Customer 48	Customer 14	Customer 30	Customer 48
ACC (%)	78.095	79.595	79.9	99.222	99.626	99.248
Precision (%)	79.865	79.717	80.519	99.233	99.234	99.230
Recall (%)	78.095	79.595	79.9	99.222	99.626	99.248
FA (%)	33.56	19.14	12.58	0.11	0.37	0.12
FNR (%)	10.0	15.06	27.84	1.57	0.38	1.45
HD (%)	44.54	60.455	67.32	99.11	99.26	99.10
F1 (%)	78.970	79.655	80.208	99.228	99.429	99.238

TABLE 11. Comparison between the performance of the global and customized models for newly launched attacks for different customers.

Metrics	Global model				Newly launched attacks customized models			
	Customer 5	Customer 20	Customer 25	Customer 35	Customer 5	Customer 20	Customer 25	Customer 35
ACC (%)	77.753	81.704	77.069	79.060	99.782	98.444	98.630	99.751
Precision (%)	80.271	83.648	81.416	85.170	99.387	98.481	98.648	99.752
Recall (%)	77.753	81.704	77.069	79.060	99.782	98.444	98.630	99.751
FA (%)	8.04	6.20	4.45	3.01	0.65	0.19	0.57	0.42
FNR (%)	36.21	30.48	41.14	42.68	0.44	2.90	2.16	0.49
HD (%)	69.72	75.5	72.62	82.16	99.13	98.26	98.07	99.33
F1 (%)	78.992	82.665	79.182	82.001	99.584	98.462	98.639	99.751

E. EXPERIMENTAL RESULTS OF SCENARIO 4

1. A global detection model is initially trained on known cyber-attacks (1st, 2nd, and 4th attacks from TABLE 2).
2. Customers launch new attacks (3rd, 5th, and 6th) using novel methods, which are used to retrain the global model into a new DDQN-based customized detection model (hybrid CNN+GRU).
3. The results show significant improvements in ACC (up to 21%) and HD (up to 30%), along with reductions in FA (up to 7.4%) and FNR (up to 42%), demonstrating the model's ability to effectively learn and detect new cyber-attacks.

TABLE 2. Cyber attack functions.

Attack No	Attack function
1^{st}	$f_1(x_i(t)) = \beta x_i(t)$
2^{nd}	$f_2(x_i(t)) = \beta_t x_i(t)$
3^{rd}	$f_3(x_i(t)) = mean(x_i)$
4^{th}	$f_4(x_i(t)) = \beta_t mean(x_i(t))$
5^{th}	$f_5(x_i(t)) = \begin{cases} 0 & t \in [t_s, t_e] \\ x_i(t) & t \notin [t_s, t_e] \end{cases}$
6^{th}	$f_6(x_i(t)) = x_i(R - t)$

CONCLUSION

This paper explores using Reinforcement Learning (RL) to detect cyber-attacks on electricity grids. They create malicious readings from real power consumption data and propose deep RL detectors for this purpose.

1. Scenario 1: RL-based detectors (DQN and DDQN) outperform DL-based detectors, with lower False Alarms (FA) and highest Difference(HD). The hybrid architecture (CNN+GRU) works best.
2. Scenario 2: Customized DDQN-based detectors for new customers perform better than global detectors.
3. Scenario 3: When consumption patterns change for existing customers, the updated detection model still performs well.
4. Scenario 4: The model can learn and detect new cyber-attacks with high accuracy, recall, and HD while maintaining low FA.

REFERENCES

[1] M. I. Ibrahem, M. Nabil, M. M. Fouda, M. M. E. A. Mahmoud, W. Alasmary, and F. Alsolami.
"Efficient privacy-preserving electricity theft detection with dynamic billing and load monitoring for AMI networks,"
IEEE Internet Things J., vol. 8, no. 2, pp. 1243–1258, Jan. 2021.

[2] A. Takiddin, M. Ismail, M. Nabil, M. M. E. A. Mahmoud, and E. Serpedin,
"Detecting electricity theft cyber-attacks in AMI networks using deep vector embeddings,"
IEEE Syst. J., vol. 15, no. 3, pp. 4189–4198, Sep. 2021.

[3] I. Ibrahem, M. M. Badr, M. Mahmoud, M. M. Fouda, and W. Alasmary,
"Countering presence privacy attack in efficient AMI networks using interactive deep-learning,"
in Proc. Int. Symp. Netw., Comput. Commun. (ISNCC), Oct. 2021, pp. 1–7.

[4] M. M. Badr, M. I. Ibrahem, M. Mahmoud, M. M. Fouda, F. Alsolami, and W. Alasmary,
"Detection of false-reading attacks in smart grid netmetering system,"
IEEE Internet Things J., vol. 9, no. 2, pp. 1386–1401, Jan. 2022.

[5] M. M. Badr, M. I. Ibrahem, M. Mahmoud, W. Alasmary, M. M. Fouda, K. H. Almotairi, and Z. M. Fadlullah,
"Privacy-preserving federatedlearning-based net-energy forecasting,"
in Proc. SoutheastCon, Mar. 2022, pp. 133–139.

Thank You