Power Data Watermarking: A New Methodology to Protect Power System Data Assets

Zhenghao Zhou, Yiyan Li*, Runlong Liu
College of Smart Energy
Shanghai Jiao Tong University
Shanghai, China
{zhenghao.zhou, yiyan.li*,
runlong_liu}@sjtu.edu.cn

Zheng Yan
School of Electronic Information and
Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China
yanz@sjtu.edu.cn

Mo-Yuen Chow
University of Michigan - Shanghai Jiao
Tong University Joint Institute
Shanghai Jiao Tong University
Shanghai, China
moyuen.chow@sjtu.edu.cn

***Abstract*—Data is being regarded as a kind of valuable assets with the fast development of the data-driven Artificial Intelligence technologies. In this paper, we introduce the concept of data watermarking to protect the power system data assets to secure the data authenticity and prevent from being misused. A deep-learning based watermarking model with specially designed loss functions and network structure is proposed to embed watermarks into the original dataset. A comprehensive evaluation framework is designed to evaluate the watermarking performance in the aspects of invisibility, restorability, robustness, secrecy and false-positive detection. Case study based on real-world load time series dataset demonstrate the effectiveness of the proposed method.**

Keywords—Watermarking, Data asset protection, Deep neural networks, Time series data.

I. INTRODUCTION

In recent years, data is being regarded as a kind of valuable asset with the fast development of the data-driven Artificial Intelligence (AI) technologies. In power system studies, data is particularly valuable and hard to acquire because it is considered related with the user privacy and energy security. As a result, it is increasingly necessary to claim the ownership of the power data assets and protect the data copyright from being tampered, misused or counterfeited.

Watermarking is a commonly used method to protect the digital assets, such as image, audio, text, etc. By embedding either visible or invisible information into the original files, watermarking technology can deter unauthorized use and ensure authenticity of the digital assets [1]. In the field of image watermarking, numerous model-based methods have been proposed, both in the spatial domain and frequency domain. Meanwhile, with the development of deep learning techniques, it has been noticed that neural networks can learn to embed tiny perturbations into the digital assets as invisible watermarks based on the encoder-decoder framework [2]. The encoder takes both the image and the watermark as inputs and generates a watermarked image, while the decoder attempts to extract the watermark from it [3]. Compared to traditional watermarking techniques, deep-learning based methods offer significant advantages, including high invisibility, strong robustness, large encoding capacity and excellent real-time decoding capabilities. Beyond image watermarking, the watermarking method described in [4] enables embedding invisible bitstrings into the parameters of the trained deep neural network models, serving as a way to protect the intellectual property of the models.

In power systems, the study of using watermarking technology to protect the data assets is still at the infant stage with few literature published [5]-[8]. In 2024, there were a few related studies mainly focusing on the topic of power system cyber security using analytical or signal processing techniques as the watermarking methods. For instance, [6] proposes a dynamic watermarking method that detects cyberattacks by embedding random signals into the control inputs of photovoltaic systems, while [8] introduces a timestamp-based digital text watermarking technique to detect data integrity and replay attacks in power systems. However, as these researches mainly focus on preventing from cyberattacks during the system operation process, the theory, methodology and evaluation criterions of implementing watermarking to protect static data assets still need to be further explored.

In this paper, we propose a deep-learning-based watermarking method to protect the time series data assets in power systems. This watermarking method can embed invisible watermarks into the original data based on specially-designed network structure and loss function. The embedded watermark has negligible influence to the data quality while still being able to be decoded even under malicious data tampering.

Our contributions are summarized as follows:

1. We propose a deep-learning watermarking model for power system data asset protection purposes. The model is formulated as an encoder-decoder framework with specially-designed loss functions. Particularly, a noise layer is introduced into the deep learning model that significantly enhances the watermarking quality.

2. We develop a comprehensive evaluation framework to evaluate the performance of the proposed watermarking method in data assets protection, including invisibility, restorability, robustness, secrecy and false-positive detection ratio. Such an evaluation framework can provide references for follow-up studies.

This work was supported in part by National Natural Science Foundation of China under Grant 52307121, and in part by Shanghai Sailing Program under Grant 23YF1419000. (Corresponding author: Yiyan Li.)

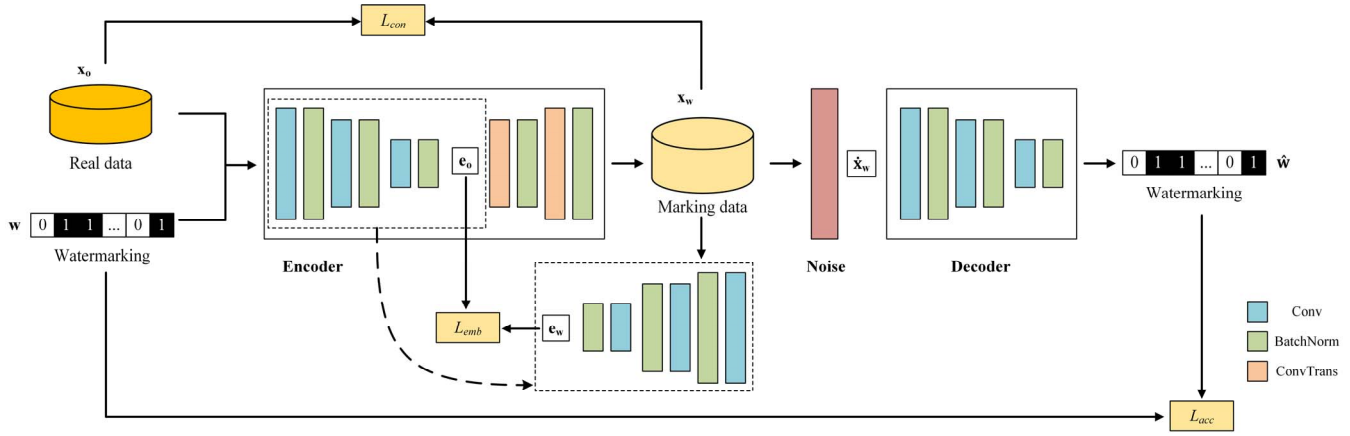

Fig. 1. Watermarking network framework.

II. Methodology

A. Watermarking Concept

Watermarking is a form of steganography to hide encoded information within the original data. The core idea of watermarking is to embed additional information into the data in such a way that it remains invisible to users but can be detected or extracted when necessary. In the context of digital assets like images, documents, or time series data, watermarking serves to protect intellectual property and ensure data integrity. A successful watermark must have the following key characteristics [9] [10]:

Invisibility: The presence of the watermark should not alter the characteristics of the original data. It should have minimal or no impact on the functionality or quality of the data. Any visible degradation could reveal the existence of the watermark and reduce its effectiveness.

Restorability: The watermarking system must ensure that the embedded watermark can be accurately restored by the decoder. The extracted watermark should be highly similar to the embedded one, with minimal distortion or error.

Robustness: The watermark should be resistant to various forms of post-processing or malicious tampering, such as removal or alteration. Even the data undergoes with compression, transformation, or noise injection, the watermark must remain detectable and intact. This ensures the watermark can endure real-world usage scenarios without degradation.

Low false-positive detection: If a watermarking method is credible, the probability of false positive detection should be exceedingly low, meaning that the data without a watermark should not be erroneously identified as containing one.

Secrecy: The watermark must be sufficiently secret so that it is difficult for unauthorized parties to detect or extract. If third parties can easily identify the watermark, they may attempt to manipulate or remove it, undermining the protection effect.

B. Deep-Learning Based Watermarking Method

1) Watermarking network framework

The watermarking network follows an encoder-decoder architecture, as shown in Fig. 1. We denote the encoder and decoder networks as E and D, respectively. The input to the encoder consists of two parts: the original electrical data $\mathbf{x_o}$ and the watermark $\mathbf{w}$, where $\mathbf{w} \in \{0, 1\}^n$ represents the source identity space, and n is its dimension. The structure of the encoder is similar to that of an autoencoder. The input data $\mathbf{x_o}$ is first processed through convolutional layers to reduce dimensionality, resulting in an intermediate embedding $\mathbf{e}$. Then, through transposed convolution, the watermarked data $\mathbf{x_w}$ is produced. Note that a noise layer is introduced between the encoder and the decoder to simulate the real-world data disturbances to enhance the watermarking robustness. After noise is added to $\mathbf{x_w}$, the decoder is designed as a deep convolutional network to recover the watermark $\mathbf{w}$ from noisy data $\dot{\mathbf{x}}_\mathbf{w}$, with the output being the reconstructed $\hat{\mathbf{w}}$.

2) Training loss

The training process of the encoder and decoder is synchronized with specially designed loss functions. To ensure the invisibility of the watermark, the encoder's objective is to generate the watermarked data $\mathbf{x_w}$ that is as close as possible to the original data $\mathbf{x_o}$. On the other hand, the decoder's goal is to accurately reconstruct the watermark $\mathbf{w}$, enhancing the model's overall accuracy. Therefore, the loss functions during the training phase are defined as follows:

$$\min_{E,D} \mathrm{E}_{\mathbf{x_o}\sim\mathrm{X},\mathbf{w}\sim\{0,1\}^n} L_{Acc}(\mathbf{w},\hat{\mathbf{w}};E,D) + \lambda(L_{con}(\mathbf{x_o},\mathbf{x_w};E)+L_{emb}(\mathbf{e_o},\mathbf{e_w};E)) \tag{1}$$

$$L_{Acc}(\mathbf{w},\hat{\mathbf{w}};E,D) = \frac{1}{n}\sum_{k=1}^{n} \mathbf{w_k}\log\hat{\mathbf{w}}_\mathbf{k} + (1-\mathbf{w_k})\log(1-\hat{\mathbf{w}}_\mathbf{k}) \tag{2}$$

$$L_{con}(\mathbf{x_o},\mathbf{x_w};E) = \left\|\mathbf{x_w}-\mathbf{x_o}\right\|_2^2 \tag{3}$$

$$L_{emb}(\mathbf{e_o},\mathbf{e_w};E) = \left\|\mathbf{e_w}-\mathbf{e_o}\right\|_2^2 \tag{4}$$

$$\mathbf{x_w} = E(\mathbf{x_o},\mathbf{w}) \tag{5}$$

$$\dot{\mathbf{x}}_\mathbf{w} = \mathbf{x_w} + noise \tag{6}$$

$$\hat{\mathbf{w}} = D(\dot{\mathbf{x}}_\mathbf{w}) \tag{7}$$

where $\mathbf{e_o}$ is the original data embedding, $\mathbf{e_w}$ is the watermarked data embedding, $\mathbf{w_k}$ and $\hat{\mathbf{w}}_\mathbf{k}$ are the k^{th} bit of the input watermark and detected watermark separately.

The loss function of the watermarking network includes 3 terms: the accuracy loss (L_{acc}), the content loss (L_{con}) and the embedding-matching loss (L_{emb}), as shown in(1)–(4). λ is the weight to balance. L_{acc} effectively represents the binary

cross-entropy, which serves to guide the decoder in the demarcation of the watermark embedded by the encoder. L_{con} employs mean squared error(MSE) to minimize the point-to-point discrepancies. Similarly, leveraging MSE, L_{emb} measures the divergence between high-level feature embeddings extracted from the intermediate layers of the encoder. Given that the high-level features of authentic power temporal data are integrated into the intermediate layer outputs, the embedding matching loss steers the encoder to generate more invisible results by aligning the high-level features present in both the real and watermarked data.

Since improving the watermark decoding accuracy can increase the distortion of the watermarked data to some extent, this paper adopts a dynamic weighting training strategy. The hyper-parameter λ in (1) is initially set to 0 at the beginning of training, ensuring that the model focuses first on the accuracy of watermark reconstruction. Once the accuracy reaches a predefined threshold, λ gradually increases with further training iterations and eventually stabilizes. This allows the model to continue improving accuracy while also considering the invisibility of the watermark. This training strategy ensures an orderly process and effectively avoids conflicts between different loss terms.

3) Noise layer

In real-world scenarios, time series data may suffer from various distortions, leading to alterations in data values and subsequently impairing data quality, making it difficult to recognize. Common issues include noise interference, missing data, and others. To enhance the robustness of the watermarking model, we add a noise layer before the decoder. This layer simulates real-world data degradation by introducing Gaussian noise with a mean of 0 and a standard deviation of 0.1 to the data. The decoder is then responsible for decoding the watermarked data after the noise has been applied.

III. CASE STUDY

To evaluate the effectiveness of our watermarking method, we utilize the dataset of 1-minute interval smart meter, which was in Austin, TX, in 2015 by the PECAN Street association [11]. We select 720 residential daily load profiles to set up the test case and reshape them from the (1440,1) configuration to the (36,40) format to facilitate convolutional processing. To test robustness, we build two new datasets: 1) noisy dataset by adding Gaussian noise with different proportions and different standard deviations(std) to the original data. 2) missing dataset by zeroing out different proportions of the original data, because typical residential load data almost never records a value of zero.

A. Invisibility

We calculate Fréchet Inception Distance (FID) and Kullback–Leibler divergence(KL) between the original data and the watermarked data to evaluate the invisibility. Lower distance indexes indicate higher similarity between the original data and the watermarked data (i.e. better invisibility). As a benchmark, we add Gaussian noise with different standard deviations to the real data to assess the invisibility of the watermark. Results are shown in Table I.

TABLE I. MODEL METRICS OF TWO METHODS

Datasets	Metrics	
	FID	KL
Noise(std=0.001)	1.985×10^{-11}	0.0085
Noise(std=0.002)	1.486×10^{-8}	0.0766
Noise(std=0.01)	4.470×10^{-8}	3.1666
Noise(std=0.05)	1.567×10^{-5}	4.9715
Noise(std=0.1)	2.159×10^{-4}	5.7164
Watermark	$\mathbf{2.296}\times10^{-5}$	**0.0149**

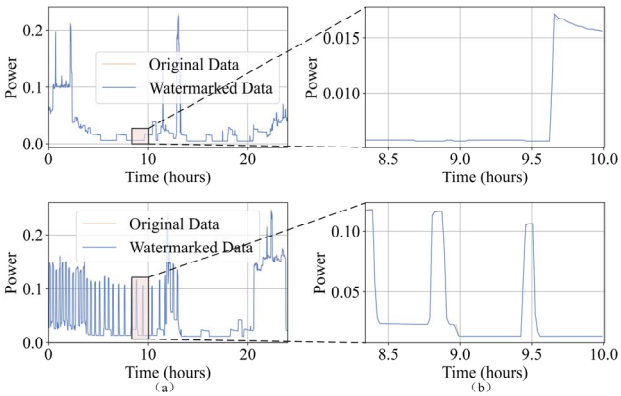

Fig. 2. Comparison between samples of the original data and the watermarked data. (a) Results overview on the samples, (b) regional zoom-in of the results.

From Table I we can see that the watermark model achieves a low FID score, comparable to the influence of the noise injection with std=0.05. The KL score is even lower and is comparable to the noise influence with std=0.001. Both metrics indicate that the watermark model performs well, introducing minimal distortion. The FID, which reflects perceptual data quality, suggests that the visual impact of the watermark is akin to the effects of low-level noise (std=0.05). Meanwhile, the KL divergence for the watermark model is even lower, suggesting a closer alignment to the noise model with std=0.001 in terms of statistical distribution. This indicates that while the watermark affects the high-level visual features (as reflected in FID), its influence on the fine-grained statistical properties of the data is minimal, similar to the very subtle noise of std=0.001. In essence, the watermark introduces a slight visual distortion, but maintains an even better performance when it comes to preserving the underlying data distribution, leading to a better KL score.

As shown in Fig. 2, the watermark embedding process has only caused a minimal impact on the data. The fluctuation and variation trends of the data have also been preserved. Such tiny impact is negligible for research purposes, such as data analytics, parameter identification and machine learning model training.

B. Restorability and Robustness

In the proposed watermarking model, the watermark is represented as a binary vector $\mathbf{w} \in \{0, 1\}^n$, where n denotes the length of the vector, set to 100 in this paper. We employ bitwise accuracy to assess the restorability of the watermark recovery process.

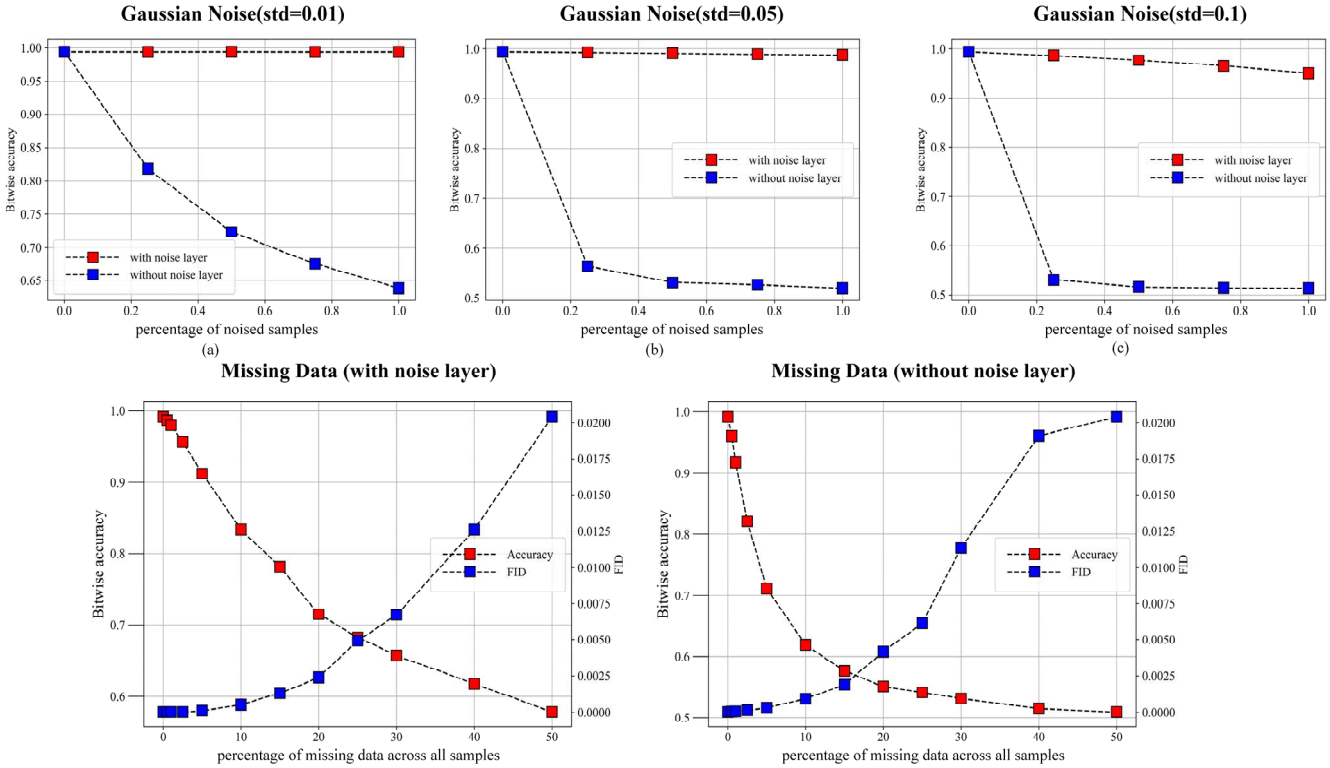

Fig. 3. Reconstruction performance of the decoder under different noise injection levels and missing data ratios. (a) Noise std = 0.01, (b) noise std = 0.05, (c) noise std = 0.1, (d) with noise layer under different missing data ratio, (5) without noise layer under different missing data ratio.

Our watermark can be successfully detected and extracted from the watermark-embedded time-series data, with the reconstruction accuracy of 99.41%. For data that has not been watermarked, the accuracy of the reconstructed watermark is 50%, given that the watermark is a binary vector, represents a level of accuracy equivalent to random guessing.

The robustness of a model is primarily reflected in the resilience of its decoder component. Consequently, we can perform post-processing operations on the watermarked data prior to its input into the decoder to emulate real-world scenarios. This includes the introduction of Gaussian noise to simulate the inherent noise in actual environments, and the random masking on real data to mimic the missing data scenarios that may arise from communication or device failures. We don't retrain the model with the post-processed data. Instead, we directly feed the post-processing data into the pre-trained decoder to assess its inherent robustness. We employ a 0.75 accuracy rate as the threshold for determining whether the watermark can still be recognized.

From Fig. 3 we have the following observations:

1. The introduced noise layer significantly improves the watermarking robustness. As shown in Fig. 3(a) - (c), as the percentage of noised samples increases, the accuracy of the watermarking model without noise layer drops significantly. On the contrary, the model with noise layer remains above 95% even when all training samples are polluted with strong Gaussian noises with standard deviation 0.1. Similar observation can be seen in Fig. 3(d)-(e). The accuracy of the watermarking model with noise layer decreases slower with the increasing of missing data ratio. Such observations demonstrate the value of the noise layer in enhancing the model robustness to resist the influence of data perturbation.

2. The data quality measured by FID decreases faster than the watermarking accuracy. As shown in Fig. 3(d)-(e), when the missing data rate reached 15%, the decoder still achieves over 75% accuracy, even though the data quality had significantly degraded (FID increased by 5200%). At a 50% missing data rate, the decoder is still able to capture some information without degrading to random guessing (accuracy of 50%), compared with the severe data quality degradation (FID increased by 77,000%). Such observations demonstrate the watermarking model with noise layer is robust and can still functioning under severe data conditions.

C. Secrecy

The presence of a watermark should not be easily detectable by third parties, as it could be susceptible to manipulation. The confidentiality of the watermark is typically measured using the ATS (Artificial Training Sets) method [12]. This is framed as a binary classification problem, distinguishing between fingerprinted and non-fingerprinted test images. But in practice, it operates in an unsupervised manner. Specifically, the method works as follows: ATS repeatedly applies the watermark algorithm to generate a set **A** containing non-watermarked and single-watermarked data, a set **B** containing single- and double-watermarked data, and a set **C** containing double- and triple-watermarked data. **C** is used as the positive dataset, while **A** serves as the negative dataset to train a supervised SVM classifier. Finally, the classifier's performance is evaluated on **B**.

TABLE II. CLASSIFICATION ACCURACY UNDER ATTACKERS

Attacker	Access to model	Access to watermark	Classification Accuracy
Weak	No	No	0.4978
Moderate	No	Yes	0.5099
Strong	Yes	Yes	0.9212

During testing, we use the previously described dataset (720 residential load profiles) as the original, non-watermarked data. We evaluate three levels of threat severity:

1) *Weak attacker* knows that a watermarked model is being used and is aware of its exact architecture, hyperparameters, and dataset. However, they do not have access to the trained model itself or the true watermark. To simulate this scenario, we train five watermarked models with different random seeds to construct an artificial training set. We then test the classification results on a sixth model, trained from scratch with random initialization.

2) *Moderate attacker* does not have access to the trained model but knows the true watermark. The evaluation follows the same procedure as in the weak attacker scenario.

3) *Strong attacker* has full access to the watermarked model and knows the true watermark. Specifically, we train a watermarked model and directly use it to evaluate detection accuracy.

As shown in Table II, even when the attacker knows the exact composition of the model, the security of the watermark remains assured (with detection results close to 50%). This indicates that each model embeds the watermark into the original data in a unique way, offering an advantage over traditional algorithms. Only when the attacker has direct access to our model can they achieve relatively high detection accuracy (with results exceeding 90%). However, in practice, we do not disclose the watermark model and only reveal the watermarked data.

D. False-positive Detection

When the watermark does not exist but the model still detects watermark samples within the data, such samples are referred to as false positives. The presence of these false positives can lead to misjudgments regarding data ownership. A well-designed watermarking model must ensure a low false positive rate. Our framework inherently exhibits a low false positive detection rate. We set both the watermark $\mathbf{w}$ for the encoder input and the watermark $\hat{\mathbf{w}}$ for the decoder output as 100-dimensional binary vectors, implying the existence of 2^{100} different possible outcomes. Meanwhile, the decoded watermark for non-watermarked data is random. The sufficiently large watermark space makes the probability of a random watermark colliding with a genuine watermark very low. Specifically, we adhere to the previously mentioned criterion: a watermark matching rate of $\geqslant 75\%$ indicates the presence of a watermark in the data.

We conduct false positive detection experiments on the training dataset. We use the positive detection rate as the evaluation metric, which refers to the ratio of detected watermarks. In the training set, watermarked data achieved a positive detection rate of 100%, whereas original data had a detection rate of 0%. Our model demonstrates outstanding performance with no false positive occurrences.

IV. CONCLUSION

In this paper, a deep-learning-based watermarking method is introduced to protect the power system time series dataset. The proposed model employs an encoder-decoder structure with a comprehensive loss function composed of accuracy loss, content loss and embedding-matching loss. Case study demonstrate that the proposed watermarking method has negligible influence to the original dataset and is considered invisible. Meanwhile, the watermark is robust to malicious data tampering, restorable by the owner but remains secrete to the unauthorized third parties. Therefore, the proposed data watermarking method is considered efficient and could be a promising way to protect power system data assets.

Future work may focus on extending the application of the watermarking method to protect other types of power system data, such as system topology, equipment models, etc.

REFERENCES

[1] M. Begum and M. S. Uddin, "Digital Image Watermarking Techniques: A Review," Information, vol. 11, no. 2, p. 110, Feb. 2020, doi: 10.3390/info11020110.

[2] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "HiDDeN: Hiding Data With Deep Networks," Jul. 25, 2018, arXiv: arXiv:1807.09937. doi: 10.48550/arXiv.1807.09937.

[3] M. Tancik, B. Mildenhall, and R. Ng, "StegaStamp: Invisible Hyperlinks in Physical Photographs," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA: IEEE, Jun. 2020, pp. 2114–2123. doi: 10.1109/CVPR42600.2020.00219.

[4] N. Yu, V. Skripniuk, S. Abdelnabi, and M. Fritz, "Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data," in 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada: IEEE, Oct. 2021, pp. 14428–14437. doi: 10.1109/ICCV48922.2021.01418.

[5] I. Balahewa, L. Bjorndal, C. Mi, and T. Huang, "A Natural Watermarking Approach to Cyber Attack Detection for Power Electronics- Interfaced Renewables," in 2024 56th North American Power Symposium (NAPS), El Paso, TX, USA: IEEE, Oct. 2024, pp. 1–6. doi: 10.1109/NAPS61145.2024.10741694.

[6] H. A. J. Ibrahim, J. Kim, J. A. Ramos-Ruiz, W. H. Ko, T. Huang, and P. N. Enjeti, "Detection of Cyber Attacks in Grid-Tied PV Systems Using Dynamic Watermarking," IEEE Trans. Ind. Appl., vol. 60, no. 1, 2024.

[7] F. Kabir, T. K. Araghi, and D. Megías, "Privacy-preserving protocol for high-frequency smart meters using reversible watermarking and Paillier encryption," Comput. Electr. Eng., vol. 119, p. 109497, Oct. 2024, doi: 10.1016/j.compeleceng.2024.109497.

[8] S. Deb Roy, A. Sharma, S. Chakrabarti, and S. Debbarma, "Securing Power System Data in Motion by Timestamped Digital Text Watermarking," IEEE Trans. Smart Grid, vol. 15, no. 5, pp. 4974–4985, Sep. 2024, doi: 10.1109/TSG.2024.3370892.

[9] D. Lin, B. Tondi, B. Li, and M. Barni, "A CycleGAN Watermarking Method for Ownership Verification," IEEE Trans. Dependable Secure Comput., pp. 1–15, 2024, doi: 10.1109/TDSC.2024.3424900.

[10] N. Agarwal, A. K. Singh, and P. K. Singh, "Survey of robust and imperceptible watermarking," Multimed. Tools Appl., vol. 78, no. 7, pp. 8603–8633, Apr. 2019, doi: 10.1007/s11042-018-7128-5.

[11] " Pecan Street Dataport." [Online]. Available: https://www.pecanstreet.org/dataport/

[12] Lerch-Hostalot, Daniel, and David Megías. "Unsupervised steganalysis based on artificial training sets." Engineering Applications of Artificial Intelligence 50 (2016): 45-59.