Unlearnable Examples For Time Series

Yujing Jiang[1], Xingjun Ma[2], Sarah Monazam Erfani[1], James Bailey[1]

[1] Faculty of Engineering and Information Technology, The University of Melbourne
{yujingj@student., sarah.erfani@, baileyj@}unimelb.edu.au
[2] School of Computer Science, Fudan University
xingjunma@fudan.edu.cn

Abstract. Unlearnable examples (UEs) refer to training samples modified to be unlearnable to Deep Neural Networks (DNNs). These examples are usually generated by adding error-minimizing noises that can fool a DNN model into believing that there is nothing (no error) to learn from the data. The concept of UE has been proposed as a countermeasure against unauthorized data exploitation on personal data. While UE has been extensively studied on images, it is unclear how to craft effective UEs for time series data. In this work, we introduce the first UE generation method to protect time series data from unauthorized training by deep learning models. To this end, we propose a new form of error-minimizing noise that can be *selectively* applied to specific segments of time series, rendering them unlearnable to DNN models while remaining imperceptible to human observers. Through extensive experiments on a wide range of time series datasets, we demonstrate that the proposed UE generation method is effective in both classification and generation tasks. It can protect time series data against unauthorized exploitation, while preserving their utility for legitimate usage, thereby contributing to the development of secure and trustworthy machine learning systems.

Keywords: Time Series Analysis · Unlearnable Example.

1 Introduction

The rapid advancement of deep learning and large models is largely driven by the vast amounts of data "freely" available on the Internet. While there has been significant research aimed at training deep learning models with privacy preservation [30,1,22,23,31], these approaches still neglect the necessity to obtain users' consent to use their data. Recent works have proposed useful tools such as Fawkes [29] to address this gap by promoting consent-based data utilization and protection. Yet, the issue remains unresolved. Rising public concerns stem from several instances where personal data, harvested from the Internet without consent, has been utilized to train commercial machine learning models [10]. Concerns now encompass not only images but also time series and multi-modal data. This broadening scope underscores the need for thorough data protection strategies, particularly in the relatively underexplored field of time series data.

In response to privacy concerns, a number of data protection techniques have been developed including secure release and protective data poisoning. Among those works, protective data poisoning techniques have become increasingly attractive as they allow users to actively add poisoning or adversarial noise into their data (like selfies) before posting them on online social media platforms to protect data exploits. Recently, more advanced data protection techniques such as Unlearnable Examples (UEs) [11,25,8,39] have been proposed which can make (image) data unlearnable to machine learning models. Contrasting this with conventional data protection techniques that simply obscure identifiable data, UEs ensure that a DNN trained on such examples performs no better than random guessing on standard test examples.

Existing research on either data poisoning-based data protection or UEs has primarily focused on image-based applications, overlooking the significance of time series data which is vital in applications such as financial forecasting [2], health monitoring [20], energy prediction [7], and transportation [18]. Given its distinct characteristics and broad applications, there is an urgent need for time series data protection methods. Image-oriented data protection methods might not translate well to time series data due to their dynamic and sequential nature [13]. Although the concept of UEs has predominantly been confined to computer vision, our research will demonstrate that this concept can also be effectively extended to time series applications.

Existing methods developed for image-based UEs often apply unlearnable noise across the whole image. However, this approach is less suitable for time series data, which is inherently sequential and often requires interventions in certain segments instead of the entire dataset. Given that a short segment in time series data can hold critical information about a particular process or entity, the direct application of image-based UE techniques to time series data encounters significant challenges. Recognizing these limitations, we propose to make only a fraction, i.e., the most sensitive or crucial part, of the time series data unlearnable. This allows for the protection of specific data segments while maintaining the usability of the remainder, balancing security with data integrity.

In this work, we extend the concept of UEs to time series data and propose a novel and effective UE generation method. Our contributions are as follows:

- We introduce a new form of error-minimizing noise that can be applied *selectively* to segments of time series. This noise is imperceptible to humans, preserving the overall utility of the data while ensuring its primary purpose of rendering the data unlearnable to DNNs.
- We propose a novel unlearnable noise generator that can mitigate the potential risk of the underlying time series data being recognized or trained by either classification or generative models. By applying this noise, we effectively create a layer of protection around the data, making it ineffective for exploitation by AI technologies, while preserving its value for legitimate use.
- We conduct empirical studies to demonstrate the effectiveness of our method in generating unlearnable examples. Our evaluation covers a broad range of time series datasets, showcasing its versatility and robustness.

2 Related Work

In this section, we briefly review the most relevant data protection methods including data poisoning, adversarial attacks, and unlearnable examples.

2.1 Data Poisoning

Data poisoning attacks aim to weaken a model's performance by altering training data. Such attacks on Support Vector Machines (SVM) were shown by [3]. Koh et al. [14] expanded this, targeting influential training samples in DNNs with adversarial noise. This was later adopted into an end-to-end framework [21]. The work "Poison Frogs!" presents a clean-label poisoning technique that retains correct labels, making the attack more insidious [26]. Backdoor attacks, another variant of data poisoning techniques, involve embedding a hidden trigger pattern into the training dataset. Despite this manipulation, these attacks will not have a detrimental impact on the model's performance when evaluated on benign (clean) data [17,40,13]. Our work diverges from these approaches by generating unlearnable examples using imperceptible noise to effectively "bypass" the training process of DNNs, rendering them incapable of learning from the altered data, thereby offering a more robust strategy for data protection.

2.2 Adversarial Attack

Adversarial attacks are techniques designed to deceive machine learning models, especially DNNs, by injecting minor, often imperceptible noise that can lead models to make different predictions. The aim is to identify the minimal input modification causing misclassification or heightened prediction error. Extensive research has established adversarial examples that can deceive DNNs during the testing phase [32,9,15,4,19,5,27]. In these attacks, the adversary identifies a form of error-maximizing noise that significantly increases the model's prediction error. In response to the vulnerabilities exposed by adversarial attacks, adversarial training has emerged as the most robust countermeasure [19,38,33,36,34,28]. This training strategy is formulated as a *min-max* optimization problem, where the objective is to minimize the model's vulnerability to error-maximizing noise while maximizing its performance on clean data.

2.3 Unlearnable Examples

In contrast to adversarial examples, which focus on error-maximizing noise, unlearnable examples (UEs) pursue the opposite direction by identifying minimal noise that reduces the model's error through a *min-min* optimization process. In this regard, Huang et al. [11] proposed the concept of UE, aimed at making training data ineffective for DNNs. Similarly, Yuan et al. [37] introduced Neural Tangent Generalization Attacks (NTGAs), a method that proficiently conducts generalization attacks on DNNs without requiring explicit knowledge about the learning model. Fu et al. [8] identified privacy limitations using error-minimizing

noise and introduced robust error-minimizing noise via a *min-min-max* optimization. This limits adversarial learners from gleaning dataset information. Ren et al. [25] introduced transferable UEs that can improve their data-wise transferability. Based on this, Zhang et al. [39] proposed Unlearnable Clusters (UCs), offering a versatile approach to create UEs adaptable to various label exploitations. On the other hand, several countermeasures have been proposed against unlearnable examples, such as UEraser [24] that uses error-maximizing data augmentation, and Jiang et al. [12] propose a method to revert unlearnable samples to learnable ones. Our work expands the UE from the image domain to the time series domain across classification and generation tasks. Our approach can target the specified segments of data and make them unlearnable, thereby safeguarding the sensitive time series data against misuse and exploitation.

3 Error-minimizing Noise For Time Series

In this section, we introduce our proposed method for generating error-minimizing noise *selectively* on segments of time series data.

3.1 Objective

In this paper, we primarily focus on applications related to time series classification and generation tasks. The models of interest in this domain are Recurrent Neural Networks (RNNs). The goal of our research is to protect time series samples that contain sensitive information in the public domain from being exploited by RNNs to ensure sensitive details are not inadvertently learned by machine learning models. Consequently, the defender's objectives are twofold. First, given the open accessibility of the data, it is imperative to inhibit deep learning models (RNNs) from processing or learning from this sensitive information. Second, these protective measures should not adversely affect the model's ability to generalize or perform its intended functions using non-sensitive information.

3.2 Threat Model

The defenders (data subjects) are aware of the general characteristics of the dataset into which their data will be collected and incorporated. This knowledge may include aspects such as the type of data, its source, and its intended application. While the defenders lack the authority to directly access or modify the dataset, they have the ability to access or alter their own individual data within it. Additionally, defenders are aware of the architecture of the DNNs being employed, but they lack information on more granular details such as the exact training procedure, optimization methods, or hyperparameters. This setting simulates the real-world scenario where defenders are often equipped with only partial information and lack full access or a complete understanding of the system. The defenders seek to safeguard their sensitive information from unauthorized exploitation by introducing error-minimizing noise into the time series data. This addition of noise is designed to render only the sensitive portions of

the data unlearnable for machine learning models. Given the defenders' limited knowledge of the exact training model, the noise introduced should be adaptable across various machine learning models. Defenders, therefore, create noise based on a model they estimate to be close to the real one. This estimated model aids in crafting noise that remains effective across different architectures.

3.3 Challenges

Building upon the established concept of unlearnable examples in image data [11,25,8,39], our research extends this technique to time series data. We aim to create specific unlearnable (error-minimizing) noise that can be added to time series samples, hindering DNNs from effectively learning from these modified samples. While most existing methods focus on images and CNNs, our approach focuses on time series and RNNs. RNNs operate by processing sequential elements and retaining information from prior elements, and this iterative, memory-like nature of RNNs poses unique challenges in generating unlearnable noise. Unlike image data, where noises added to different locations are more independent, noises in time series data can be highly interconnected across the sequence, interfering with each other. Thus, a change at one single time point can cascade effects throughout the sequence. Given the memory mechanism of RNNs, even slight perturbations can amplify in later stages, greatly affecting the final output. Hence, creating error-minimizing noise for RNNs requires a novel approach that ensures the targeted segments of data are unlearnable while preserving the integrity and semantic value of the remaining parts.

3.4 Problem Formulation

Consider a time series sequence x_i, indexed by time t, which can be formally represented as $x = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{t-1}, \mathbf{x}_t\}$. This sequence is processed through an RNN model for classification task that yields $y_i = f_\theta(x_i)$, where y_i serves as a class probability vector in the context of time series classification, or as a generated sequence for sequence generation. θ represents the model's learnable parameters, which govern the transformation f. Training the RNN model is to minimize its empirical error on the training samples, which can be achieved via empirical risk minimization (ERM). The optimization problem can be formulated as follows:

$$\min_{\theta} \mathbb{E}_{(x_i, y_i) \in D} \ell(f_\theta(x_i), y_i). \tag{1}$$

where D represents the training data and ℓ is the loss function that quantifies the dissimilarity between the model's output and the true target. To ensure minimal or negligible updates to the model parameters for a given time series sample, we introduce an error-minimizing noise denoted as δ. The primary objective of incorporating this noise is to significantly reduce the training loss of a sample when noise has been added to it. This noise term is designed to have the same dimensional structure as the input sample x, resulting in a sequence of noise

values $\delta = \{\delta_0, \delta_1, ..., \delta_{t-1}, \delta_t\}$. Consequently, when considering the time series sample x_i in conjunction with its sample-specific error-minimizing noise δ_i, the combined effect can be mathematically expressed as follows:

$$\ell(f_\theta(x_i + \delta_i), y_i) \to 0. \tag{2}$$

The objective of this perturbation is to drive the loss $\ell(\cdot)$ towards zero. By doing so, the noise serves to minimize the discrepancy between the RNN's output and the actual target y_i. Consequently, the model is tricked into learning nothing from these perturbed samples.

3.5 A Straightforward Baseline Approach

A baseline method can be established for the generation of unlearnable examples in time series data by leveraging the concept of unlearnable examples as described in [11]. During the training phase of a basic noise generator, denoted as f'_θ, the system aims to solve the optimization problem as stated in Equation 3.

$$\min_\theta \frac{1}{n} \sum_{i=1}^{n} \min_{\|\delta_i\| \leq \rho_u} \ell(f'_\theta(x_i + \delta_i), y_i). \tag{3}$$

The generation of an unlearnable example, represented as (x', y), is accomplished using the trained noise generator f'_θ. This transformed data point is formally defined in Equation (4).

$$x' = x + \arg \min_{\|\delta\| \leq \rho_u} \ell(f'_\theta(x + \delta), y). \tag{4}$$

Given the sequential nature of RNNs, Backpropagation Through Time (BPTT) will be used where the network will be unrolled to match the length of the time series data. The calculation of the loss with respect to this unrolled RNN model takes into account these hidden states, allowing for a more detailed understanding of how each temporal data point in the sequence influences the overall loss.

3.6 Controllable Noise on Partial Time Series Samples

A significant limitation of directly translating image-based methods to time series data is the inability to localize and control the region of noise application. In this case, noise tends to be distributed uniformly across the entire sequence. In the context of fixed-sized inputs, such as images, this uniform distribution is generally acceptable because the noise can be easily processed and interpreted within a consistent framework. However, given that RNN models process time series data in sequential order across the time regions, the effectiveness of this noise is not uniform across different temporal segments. Consequently, some portions of the time series will be more affected than others, leading to inconsistent training and prediction outcomes.

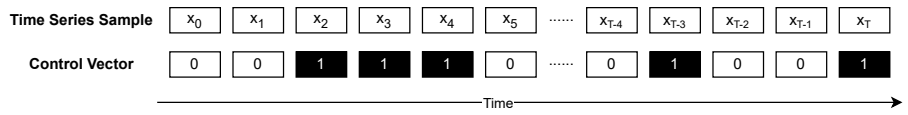

Fig. 1. Illustration of the control vector applied on a time series sample of length T. Data protection is indicated when the control vector highlights particular time stamps with a value of 1 (marked in black).

We propose a novel control vector, denoted as v, that highlights regions within the samples that should be "protected" from data exploitation. This concept is depicted in Figure 1. As an example, consider a dialogue that comprises speech data from multiple individuals. If there is a need to protect the speech of a specific individual, their corresponding temporal segments can be distinctly marked using the control vector. To achieve this, we selectively add error-minimizing noise to the targeted segments of the time series data. Our primary objective is to reduce the training loss associated with these specified regions of the time series samples by solving the following optimization problem:

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \min_{\|\delta_i\| \leq \rho_u} |\ell(f'_\theta(x_i + \delta_i \odot v_i), y_i) - \alpha \cdot \ell(f'_\theta(x_i \odot (1 - v_i)), y_i)|, \tag{5}$$

where $\odot$ represents element-wise multiplication on two vectors, and $|x|$ represents the absolute value of x. Specifically, our objective is to ensure that the training loss incurred by the sample with noise added to the targeted region ($x_i + \delta_i \odot v_i$) is equivalent to the training loss when the target region is completely omitted from the time series sample ($x_i \odot (1 - v_i)$). This is achieved by minimizing the absolute difference between these two loss terms. By aligning the loss from noise addition to that of complete removal on the partial time series sample, we ensure that the model does not derive any insights from the target regions of the sample, while preserving the consistent learning patterns from the other segments. In summary, our method endeavors to provide a new solution that bridges the gap between conventional error-minimizing noise generation methods and the unique requirements of time series data.

4 Experiments

In this section, we evaluate our proposed controllable error-minimizing noise in both time series classification and sequence generation tasks.

4.1 Experiment Setup

For our experiments on time series classification tasks, we use a simple RNN architecture as the backbone model. This architecture consists of an input layer,

Table 1. Performance degradation results of various noise types introduced into the training data. Datasets D_1 through D_6 are univariate, sourced from the UCR Archive; whereas D_7 to D_{10} are multivariate from the MTS Archive. The 2^{nd} column, labeled as **Clean**, depicts the accuracy of models trained on benign data.

Dataset	Clean	Masking	Universal	Ours$_{(20\%)}$	Ours$_{(50\%)}$	Ours$_{(100\%)}$
(D_1) BirdChicken	96.0%	80.9% (-15.1%)	39.1% (-56.9%)	19.3% (-76.7%)	12.1% (-83.9%)	8.8% (-87.2%)
(D_2) ECG5000	94.6%	78.4% (-16.2%)	41.6% (-53.0%)	16.9% (-77.7%)	11.5% (-83.1%)	7.4% (-87.2%)
(D_3) Earthquakes	72.5%	65.1% (-7.4%)	27.7% (-44.8%)	10.7% (-61.8%)	6.4% (-66.1%)	3.1% (-69.4%)
(D_4) ElectricDevices	72.3%	63.3% (-9.0%)	22.7% (-49.6%)	10.5% (-61.8%)	7.6% (-64.7%)	5.0% (-67.3%)
(D_5) Haptics	50.2%	29.0% (-21.2%)	17.5% (-32.7%)	13.4% (-36.8%)	8.3% (-41.9%)	6.3% (-43.9%)
(D_6) PowerCons	88.2%	68.4% (-19.8%)	37.0% (-51.2%)	15.6% (-72.6%)	9.3% (-78.9%)	4.9% (-83.3%)
(D_7) ArabicDigits	99.4%	83.0% (-16.4%)	26.1% (-73.3%)	9.4% (-90.0%)	4.7% (-94.7%)	2.1% (-97.3%)
(D_8) ECG	87.4%	74.2% (-13.2%)	20.2% (-67.2%)	8.9% (-78.5%)	5.6% (-81.8%)	2.6% (-84.8%)
(D_9) NetFlow	89.4%	78.5% (-10.9%)	16.7% (-72.7%)	6.7% (-82.7%)	3.2% (-86.2%)	1.9% (-87.5%)
(D_{10}) UWave	93.4%	80.8% (-12.6%)	26.4% (-67.0%)	10.2% (-83.2%)	7.9% (-85.5%)	4.0% (-89.4%)

the dimensions of which are determined by the feature set of the dataset. The model includes three recurrent hidden layers, each having 64 hidden units, and one output layer. For training, we adopted a batch size of 256 and used the Adam optimizer with a starting learning rate of 0.01. Specific parameters for RNN training included a 0.01 learning rate for noise generation (γ), a maximum noise magnitude set to $0.05 \times max_{\text{magnitude}}$ per sample (ρ_u), a trade-off parameter of 1 (α), a warm-start duration of 5 epochs ($T_{\text{warm\_start}}$), and a total training epoch of 50 (T_{training}). We use the model checkpoint at the 55^{th} epoch as the final error-minimizing noise generator. Subsequently, we applied three different noise configurations with the control vector v, covering 20%, 50%, and 100% of the sample with 10% non-overlapping consecutive segments. The positioning of these segments is selected randomly for every sample.

We use ten unique time series datasets, including six univariate datasets from the UCR Archive and four multivariate datasets from the MTS Archive. We also employ two baseline methods including masking and universal adversarial perturbation (UAP) [16]. In our approach, we use masking to hide specific segments within time series samples. We randomly choose segments covering 50% of each sample, dividing them into five non-overlapping regions, each spanning 10% of the sample. This masking serves as a baseline to gauge the model's performance with significant data absence. To ensure equitable comparison, the adversarial perturbation was capped at $0.05 \times max_{\text{magnitude}}$ and integrated into 50% of every sample, specifically at the same regions chosen for masking.

4.2 Against Classification Models

Our experimental results for the controllable unlearnable noise generator, featuring three configurations and two baseline methods, are presented in Table 1. Using the masking baseline, we noticed a 14.18% average drop in accuracy

compared to the clean model. The unmasked segments, retaining key features, possibly account for the limited decline. With the time series UAP, the accuracy decrease averaged 56.84%. Remarkably, our proposed noise method, targeting only 20% of samples, achieved a more significant average accuracy drop of 72.18%, emphasizing its efficacy over the UAP. With 50% targeting, as in the baselines, accuracy fell to 7.66%, marking a 76.68% reduction from the clean model. Additionally, our error-minimizing noise demonstrates more significant impacts on multivariate datasets, which capture the interactions and relationships across multiple variables. This multi-dimensionality allows the unlearnable noise to envelop both primary and subtle features. As a result, it can reduce the training loss more effectively, obscuring the genuine data patterns.

Taking a closer look at the accuracy drops, we found that increasing the amount of unlearnable noise does not linearly decrease classification accuracy. This suggests a diminishing return on increasing noise levels, indicating that beyond a certain threshold, the addition of more unlearnable noise might not yield significantly enhanced privacy protections. This observation implies that introducing noise to only a segment of the time series might be the most beneficial strategy. By targeting only a small part of the samples, one can achieve the desired reduction in training effectiveness without compromising the entire dataset. The results highlight the efficacy of our method, showcasing its adaptability in safeguarding time series data privacy, especially potent against data misuse in classification tasks.

4.3 Against Generative Models

We extend the evaluation to assessing the application of our proposed unlearnable noise in the context of time series generation tasks. We employ 8 multivariate time series datasets for this study, encompassing a range of classes and sample sizes. For the task of data generation, we apply two time series generative models: the Recurrent GAN (RGAN) [6] and Quant GAN (QGAN) [35]. These models are then used to generate synthetic data for the first class (class 0) of each dataset. We follow the training procedure stated in the original papers. The noise is configured to perturb 50% of the samples in the target class, and every selected sample is entirely perturbed by the noise.

We apply the *Train on Synthetic, Test on Real* (TSTR) [6] approach to test the effectiveness of our proposed unlearnable noise. Specifically, we first train a GAN model with data perturbed by unlearnable noise, then train a classifier model using data generated by the GAN and subsequently test it on a separate set of genuine samples. In this experiment, we subset all samples from the first class (class 0) of each dataset and then feed them for GAN training. The objective is to minimize the generator's reconstruction loss on the entire sample. Then, we train the time series classifiers using the generated synthetic samples, using Long Short-Term Memory (LSTM) and Fully Convolutional Network (FCN).

The experimental results shown in Table 2 demonstrate a significant drop in performance when adding unlearnable noise to 50% of the training samples. The average classification accuracy drops below 10%, marking an average reduction

Table 2. Classification accuracy of real or synthetic time series samples using the "Train on Synthetic, Test on Real" (TSTR) approach. The 2^{nd} column, labeled as **Real**, depicts the accuracy of classification models trained and tested on benign data. The columns presented as $\mathbf{Model}_c$ use clean data to train the generative model. The columns presented as $\mathbf{Model}_n$ use unlearnable data to train the generative model.

Dataset	Network	Real	$\mathbf{RGAN}_c$	$\mathbf{RGAN}_n$	$\mathbf{QGAN}_c$	$\mathbf{QGAN}_n$
(D_7)	FCN	99.6%	75.2%	6.2%	78.6%	8.4%
	LSTM	98.4%	83.4%	4.2%	81.4%	2.6%
(D_8)	FCN	91.2%	77.6%	7.4%	76.0%	7.8%
	LSTM	89.4%	72.0%	3.0%	73.6%	5.8%
(D_9)	FCN	94.6%	75.4%	11.6%	77.0%	10.6%
	LSTM	90.1%	74.0%	6.8%	75.6%	3.4%
(D_{10})	FCN	95.0%	82.0%	10.5%	84.0%	13.6%
	LSTM	93.0%	86.0%	5.2%	88.0%	6.2%
(D_{11})	FCN	94.2%	80.4%	8.4%	83.2%	9.6%
	LSTM	95.0%	76.4%	3.8%	82.4%	5.8%
(D_{12})	FCN	78.9%	54.2%	6.2%	58.0%	8.4%
	LSTM	76.0%	57.8%	2.8%	60.8%	4.0%
(D_{13})	FCN	86.4%	68.6%	11.6%	73.6%	10.8%
	LSTM	75.0%	64.2%	6.0%	72.4%	5.2%
(D_{14})	FCN	71.0%	54.6%	9.5%	61.2%	10.2%
	LSTM	64.0%	49.0%	5.4%	56.0%	4.8%

of over 60% when compared to the results obtained for clean data training. Note that, while the noise is introduced into only 50% of the samples within a specific class, it has the capability to render the entire class unlearnable (non-generative) against the sequence generation model. This implies that our proposed noise has great potential to be applied to protect sensitive samples from being learned during the training of a generative model, preventing the model from recreating or understanding the sensitive or private aspects of the original data.

5 Conclusion

In this work, we have studied the problem of protecting time series data against unauthorized exploitations. We extended the concept of Unlearnable Examples (UEs) from the image domain to the time series domain and proposed a novel method specifically designed for generating unlearnable noise for time series. The proposed method leverages a novel min-min bilevel optimization framework alongside a control vector, enabling the creation of unlearnable noise targeted at the most sensitive parts of a time series. This approach can be selectively used on specific segments of the time series data. Through extensive experiments on both time series classification and generation tasks, we demonstrated the effectiveness of our method across different datasets. Our work could help individuals and organizations protect their time series data from being exploited (without permission) in the development of commercial models.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: SIGSAC (2016)
2. Barra, S., Carta, S.M., Corriga, A., Podda, A.S., Recupero, D.R.: Deep learning and time series-to-image encoding for financial forecasting. IEEE/CAA Journal of Automatica Sinica **7**(3), 683–692 (2020)
3. Biggio, B., Nelson, B., Laskov, P.: Poisoning attacks against support vector machines. arXiv preprint arXiv:1206.6389 (2012)
4. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: SP (2017)
5. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. arXiv preprint arXiv:2003.01690 (2020)
6. Esteban, C., Hyland, S.L., Rätsch, G.: Real-valued (medical) time series generation with recurrent conditional gans. arXiv e-prints pp. arXiv–1706 (2017)
7. Feng, Y., Duan, Q., Chen, X., Yakkali, S.S., Wang, J.: Space cooling energy usage prediction based on utility data for residential buildings using machine learning methods. Applied energy **291**, 116814 (2021)
8. Fu, S., He, F., Liu, Y., Shen, L., Tao, D.: Robust unlearnable examples: Protecting data against adversarial learning. arXiv preprint arXiv:2203.14533 (2022)
9. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. ICLR (2015)
10. Hill, K.: The secretive company that might end privacy as we know it (2020)
11. Huang, H., Ma, X., Erfani, S.M., Bailey, J., Wang, Y.: Unlearnable examples: Making personal data unexploitable. In: ICLR (2020)
12. Jiang, W., Diao, Y., Wang, H., Sun, J., Wang, M., Hong, R.: Unlearnable examples give a false sense of security: Piercing through unexploitable data with learnable examples. arXiv preprint arXiv:2305.09241 (2023)
13. Jiang, Y., Ma, X., Erfani, S.M., Bailey, J.: Backdoor attacks on time series: A generative approach. In: SaTML (2023)
14. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: ICML (2017)
15. Kurakin, A., Goodfellow, I., Bengio, S.: Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236 (2016)
16. Li, J., Zhang, X., Jia, C., Xu, J., Zhang, L., Wang, Y., Ma, S., Gao, W.: Universal adversarial perturbations generative network for speaker recognition. In: ICME (2020)
17. Liu, Y., Ma, X., Bailey, J., Lu, F.: Reflection backdoor: A natural backdoor attack on deep neural networks. In: ECCV (2020)
18. Ma, T., Antoniou, C., Toledo, T.: Hybrid machine learning algorithm and statistical time series model for network-wide traffic forecast. Transportation Research Part C: Emerging Technologies **111**, 352–372 (2020)
19. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: ICLR (2018)
20. Maweu, B.M., Shamsuddin, R., Dakshit, S., Prabhakaran, B.: Generating healthcare time series data for improving diagnostic accuracy of deep neural networks. IEEE Transactions on Instrumentation and Measurement **70**, 1–15 (2021)
21. Muñoz-González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E.C., Roli, F.: Towards poisoning of deep learning algorithms with back-gradient optimization. In: AISec (2017)

22. Phan, N., Wang, Y., Wu, X., Dou, D.: Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In: AAAI (2016)
23. Phan, N., Wu, X., Hu, H., Dou, D.: Adaptive laplace mechanism: Differential privacy preservation in deep learning. In: ICDM (2017)
24. Qin, T., Gao, X., Zhao, J., Ye, K., Xu, C.Z.: Learning the unlearnable: Adversarial augmentations suppress unlearnable example attacks. arXiv preprint arXiv:2303.15127 (2023)
25. Ren, J., Xu, H., Wan, Y., Ma, X., Sun, L., Tang, J.: Transferable unlearnable examples. In: ICLR (2022)
26. Shafahi, A., Huang, W.R., Najibi, M., Suciu, O., Studer, C., Dumitras, T., Goldstein, T.: Poison frogs! targeted clean-label poisoning attacks on neural networks. NeurIPS (2018)
27. Shafahi, A., Najibi, M., Xu, Z., Dickerson, J., Davis, L.S., Goldstein, T.: Universal adversarial training. In: AAAI (2020)
28. Shan, S., Ding, W., Wenger, E., Zheng, H., Zhao, B.Y.: Post-breach recovery: Protection against white-box adversarial examples for leaked dnn models. In: ACM SIGSAC Conference on Computer and Communications Security (2022)
29. Shan, S., Wenger, E., Zhang, J., Li, H., Zheng, H., Zhao, B.Y.: Fawkes: Protecting personal privacy against unauthorized deep learning models. In: USENIX-Security (2020)
30. Shokri, R., Shmatikov, V.: Privacy-preserving deep learning. In: SIGSAC (2015)
31. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: SP (2017)
32. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. ICLR (2014)
33. Wang, Y., Ma, X., Bailey, J., Yi, J., Zhou, B., Gu, Q.: On the convergence and robustness of adversarial training. In: ICML. pp. 6586–6595 (2019)
34. Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., Gu, Q.: Improving adversarial robustness requires revisiting misclassified examples. In: ICLR (2020)
35. Wiese, M., Knobloch, R., Korn, R., Kretschmer, P.: Quant gans: deep generation of financial time series. Quantitative Finance **20**(9), 1419–1440 (2020)
36. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. Advances in Neural Information Processing Systems **33** (2020)
37. Yuan, C.H., Wu, S.H.: Neural tangent generalization attacks. In: International Conference on Machine Learning. pp. 12230–12240. PMLR (2021)
38. Zhang, H., Yu, Y., Jiao, J., Xing, E.P., Ghaoui, L.E., Jordan, M.I.: Theoretically principled trade-off between robustness and accuracy. ICML (2019)
39. Zhang, J., Ma, X., Yi, Q., Sang, J., Jiang, Y.G., Wang, Y., Xu, C.: Unlearnable clusters: Towards label-agnostic unlearnable examples. In: CVPR (2023)
40. Zhao, S., Ma, X., Zheng, X., Bailey, J., Chen, J., Jiang, Y.G.: Clean-label backdoor attacks on video recognition models. In: CVPR (2020)