DOI: 10.58895/ksp/1000162754-9 erschienen in:
Proceedings. 33. Workshop Computational Intelligence - Berlin, 23. – 24. November 2023
DOI: 10.5445/KSP/1000162754 | https://www.ksp.kit.edu/site/books/e/10.5445/KSP/1000162754/

Robust Training with Adversarial Examples on Industrial Data

Julian Knaup, Christoph-Alexander Holst, Volker Lohweg

inIT – Institute Industrial IT
Technische Hochschule Ostwestfalen-Lippe
Campusallee 6, 32657 Lemgo
E-Mail: {julian.knaup, christoph-alexander.holst, volker.lohweg}@th-owl.de

Abstract

In an era where deep learning models are increasingly deployed in safety-critical domains, ensuring their reliability is paramount. The emergence of adversarial examples, which can lead to severe model misbehavior, underscores this need for robustness. Adversarial training, a technique aimed at fortifying models against such threats, is of particular interest. This paper presents an approach tailored to adversarial training on tabular data within industrial environments.

The approach encompasses various components, including data preprocessing, techniques for stabilizing the training process, and an exploration of diverse adversarial training variants, such as Fast Gradient Sign Method (FGSM), Jacobian-based Saliency Map Attack (JSMA), DeepFool, Carlini & Wagner (C&W), and Projected Gradient Descent (PGD). Additionally, the paper delves into an extensive review and comparison of methods for generating adversarial examples, highlighting their impact on tabular data in adversarial settings.

Furthermore, the paper identifies open research questions and hints at future developments, particularly in the realm of semantic adversarials. This work contributes to the ongoing effort to enhance the robustness of deep learning models, with a focus on their deployment in safety-critical industrial contexts.

1 Introduction

In recent years, artificial intelligence (AI) has witnessed tremendous advancements, revolutionizing various domains and becoming an integral part of our daily lives. From computer vision systems [1, 2] to natural language processing [19, 4] and object detection [5] for autonomous vehicles, deep learning models have showcased remarkable capabilities, surpassing human performance in many complex tasks. In particular, AI experienced extreme media interest due to the capabilities of ChatGPT [4]. However, as AI systems become increasingly integrated into critical applications, ensuring their reliability and robustness becomes imperative.

One of the key challenges in the deployment of deep learning models is their vulnerability to adversarial examples (AEs). AEs are carefully crafted perturbations applied to input data, often imperceptible to humans, that can cause deep learning models to misbehave or produce incorrect predictions [6]. The existence of AEs has raised significant concerns about the reliability and security of AI systems, particularly in safety-critical domains such as healthcare, autonomous driving, and industrial automation.

Nowadays, industrial production plants are intelligent technical systems. These cyber-physical production systems can be severely affected by AEs, causing major financial or personnel damage. An attacker can either stop systems without an anomaly being present or allow them to continue operating even though a fault has occurred [7, 8]. Notably, in the industrial context, data exhibits high heterogeneity, diverging significantly from the limited value ranges typically encountered in image data, the origin of AEs. Additionally, industrial data can often be unstructured and accompanied by sparse labels. To effectively employ common AE generation algorithms, preprocessing of industrial data becomes a necessary step.

This paper delves into the practical application of AEs within the industrial landscape. Specifically, this paper encompasses the following key elements:

- an exploration of prevalent AE generation algorithms,
- practical insights into adversarial training techniques,

- preprocessing methodologies tailored for tabular data,
- a comparative analysis of diverse adversarial attacks, evaluating their suitability for adversarial training with tabular data drawn from the industrial context,
- identification of ongoing challenges and a prospective outlook on future research avenues.

2 Related Work and Preliminaries

This section provides background information and relevant methods for generating adversarial examples (AEs) and countermeasures to enhance robustness.

2.1 Adversarial Examples

The concept of AEs was initially introduced by *Szegedy et al*. [6] and *Biggio and Roli* [9]. In general they can be defined as:
Let $x \in \mathbb{R}^d$ be an input with true label y_0 and y_t is a (target) label different from y_0. An AE x' results from a mapping $\mathscr{A} : \mathbb{R}^d \to \mathbb{R}^d$ such that the modified input $x' = \mathscr{A}(x)$ is misclassified as y_t without changing its true class.

However, mapping $\mathscr{A}(\cdot)$ is often limited to a linear operation [10], so that an additive perturbation δ is introduced

$$x' = x + \delta.$$

To avoid changing the original class membership, δ must be small w. r. t. a distance metric. On image data, δ is commonly minimized in the literature [10] w. r. t. the L_p norm

$$||x' - x||_p = ||\delta||_p = \left(\sum_{i=1}^{n} |\delta_i|^p \right)^{\frac{1}{p}}$$

to create AEs that are visually indistinguishable to human observers. In particular, the L_0, L_2 and L_∞ norm are employed [11]. The L_0 norm represents the number of changed features or pixel, the Euclidean distance is measured with the L_2 norm and the L_∞ norm indicates the maximum change of a feature or pixel.

2.2 Adversarial Attacks

To generate AEs, a variety of approaches have been proposed, again with various modifications [10]. In the following, the most influential basic methods are presented, which will be compared later. Only white-box methods were considered, i. e. those that have complete knowledge of all parameters, as they allow for the strongest attacks [10].

Fast Gradient Sign Method

Szegedy et al. describe the generation of AEs as a constrained optimization problem [6]. They leverage the box-constrained limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm to obtain solutions. However, to reduce the computational cost, *Goodfellow et al*. introduce the Fast Gradient Sign Method (FGSM) [12]. Here, gradients ∇_x are calculated once for all input features. Each input feature is then modified in gradient ascent direction by a fixed step size ε to maximize the loss function $\mathscr{L}$

$$\delta = \varepsilon \cdot sign(\nabla_x \mathscr{L}(x, y_0)). \tag{1}$$

Since the stepsize ε is equal for all input features and they are all modified at once, the FGSM is optimized for the L_∞ norm. Furthermore, the FGSM is fast to compute but not an optimal solution. *Kurakin et al*. [13, 14] provide an iterative version of this attack, which leads to more sophisticated AEs.

Jacobian-based Saliency Map Attack

Papernot et al. introduced the Jacobian-based Saliency Map Attack (JSMA) [16]. They compute the Jacobian matrix for a specific target class w. r. t. its input features. Based on these partial derivatives a saliency map is constructed indicating the influence of each input feature. Subsequently, the most influential input is modified accordingly and checked whether an AE is present. This process is repeated until a predefined number of features has been altered or an AE has been found. Due to the successive nature of feature changes, the JSMA is optimized for the L_0 norm.

DeepFool

The basic idea of the DeepFool algorithm [15] is to view the model as an affine transformation, i. e. the authors linearize the models decision boundary around an input x. In binary classification the decision boundary becomes a hyperplane and in the multinomial case the decision boundaries around the input x are approximated with a polyhedron formed by each of the decision hyperplanes. They project the input orthogonal, i. e. with minimum distance, to the nearest hyperplane and push it slightly beyond it to craft an AE. Since the linearization is an approximation they just take a step in the direction of this projection and iterate this process until an AE is reached. In the original version the algorithm is optimized for the L_2 norm.

Carlini & Wagner

Carlini and Wagner [11] offer a variety of attacks with L_0, L_2, and L_∞ distance metrics. However, they claim their L_2 attack (C&W) to be the strongest one and in fact, the L_0 version leverages the L_2 attack. They iteratively optimize an objective function consisting of a misclassification term and a distance measure of the perturbation. Furthermore, they exploit a scaled and shifted *tanh* function with a variable exchange

$$\delta = \frac{1}{2}(tanh(w)+1) - x \tag{2}$$

to let the perturbation map natively into the interval $[0, 1]$. By eliminating the necessity of clip functions in this way, they are able to employ momentum-based optimizers such as Adam [17]. The C&W attack is one of the strongest attacks in terms of finding minimal perturbation and fooling the machine learning model [11]. Additionally, they overcame numerous defensive strategies [19], such as Defense Distillation [18], that existed at the time of release.

Projected Gradient Descent

As Gradient Descent is a standard way to solve an unconstrained optimization problem, Projected Gradient Descent (PGD) in general provides a way to solve constrained optimization problems. The PGD attack [20] leverages this approach to generate AEs. One starts from a random perturbation in an L_p ball around an input sample, takes a step in the gradient direction of the loss function w. r. t. its input data and, if necessary, projects the result back into the L_p ball. This procedure is repeated until convergence or exceeding the maximum number of iterations. Therefore, *Madry et al*. [20] reference the iterative FGSM as an L_∞ bounded PGD attack, where the projection is realized by the clipping function. The authors claim that the PGD method is probably the strongest first-order attack. They argue that AEs generated with it are more suitable for adversarial training, since models are also robust against weaker methods after training with these AEs.

2.3 Adversarial Defensives

The sequence of developments in countermeasures for adversarial examples is similar to the history of cryptography. After methods for defense are proposed, there are new attack strategies, which in turn overcome them [19]. Defensive approaches that do not require the secrecy of specific aspects, such as gradients, are therefore to be preferred here as well [21].

Adversarial training is a primary strategy for enhancing the adversarial robustness of neural networks. By introducing AEs during training [6, 12], models can be designed to be more robust to small perturbations. *Madry et al*. consider

adversarial training as a saddle point or min-max problem [20]. On the one hand, the goal is to generate AEs that maximize the loss function and, on the other hand, to find model parameters that minimize this loss. Moreover, *Tsipras et al*. demonstrate that adversarial training can lead to more robust features, which, however, are obtained at the expense of accuracy [22]. A more detailed overview of adversarial training can be found in [23].

3 Approach

In this section, an approach is developed that facilitates cross-comparison of AE generation methods. To ensure comparability among the presented methods, appropriate metrics must be selected. However, there is no uniform definition of quantifiable adversarial robustness in the literature. Additionally, adversarial attacks are optimized w. r. t. different L_p norms, further complicating the assessment of AE quality. To address these challenges, we first empirically test whether adversarial training enhances model robustness against attacks using the same method as in training. To achieve this, we employ both the accuracy on the original data and the accuracy on the AEs as metrics. Subsequently, models trained using one method are evaluated against the remaining attacks.

Another critical consideration is the nature of the data. Humans have less intuition for tabular, numerical data compared to speech or images [24]. In the image domain, the L_p norm serves as an approximation for human perception. Visual inspection helps assess whether visible artifacts are present in the AEs. This allows to establish a budget for the adversarial attacks such that these artifacts are minimized while ensuring that the original class membership of the sample is maintained. However, this is not feasible for tabular data, so alternative constraints on the adversarial attacks are required. The specific limitation of the attack methods is detailed in the next section.

Moreover, various approaches exist for conducting adversarial training. In [20], the iterative training is exclusively performed on the AEs to reduce computational costs, arguing that AEs already offer greater diversity than the original data points. Conversely, *Specht et al.* compute AEs only once and not

iteratively within the training, augmenting their original training dataset once with an equivalent amount of AEs [7, 8]. However, our initial tests failed to reproduce sufficient robustness when training solely on once-generated AEs. Given the trade-off between adversarial robustness and accuracy [22], we adopt a mixed approach. We include the original data in training to prioritize accuracy, but AEs are recalculated in each minibatch with the current model parameters. For each input, an AE is computed without applying a weighting parameter, ensuring that AEs and original data have equal influence on the loss.

Additionally, we introduce a one-epoch warm-up phase to stabilize the training process. During this phase, only the original data is utilized. Starting from epoch two, a mixture of AEs and original data is incorporated. The warm-up phase is essential as AEs, considered as worst-case inputs, are typically more challenging to learn than the distribution of the original data, which can potentially interfere with finding the appropriate parameters at the beginning. Eliminating the warm-up phase in later implementations resulted in some classes not receiving any predictions at all.

Furthermore, the heterogeneous nature of industry data requires adjustments to restrict the range of feature values. This prevents the emergence of unrealistic values and eliminates the need to adapt algorithms, as they naturally operate in the constrained range $x + \delta \in [0,1]^n$, originating from the image processing domain. Achieving this is straightforward through a min-max scaler. However, when scaling, it is important to consider the variance within the dataset. Special attention must be paid to extreme outliers that would distribute the majority of data into a significantly smaller interval.

4 Evaluation

4.1 Experimental Setup

Dataset

The experimantal results are obtained on the Sensorless Drive Diagnoses (SDD) dataset [25]. This dataset is derived from two-phase currents measured in a 425W permanent magnet synchronous motor, which is part of a modular demonstrator, as detailed in [26, 27]. The demonstrator itself is comprised of several components, including the test motor, measuring shaft, bearing module, flywheel, and load motor. To simulate various fault conditions, synthetic hardware corruptions can be introduced. The raw data from the demonstrator were preprocessed as described in [28] when creating the SDD dataset. Thereby, empirical mode decomposition was applied to determine three intrinsic mode functions and their residuals per phase. Subsequently, the mean, standard deviation, skewness, and kurtosis were calculated for each, resulting in a total set of 48 features.

The SDD dataset encompasses 58,509 samples and includes 11 distinct classes, maintaining a balanced distribution. In this context, class 1 signifies the fault-free state of the engine, while the remaining ten classes represent various fault cases stemming from issues such as shaft misalignment, axis inclination, or bearing failure. A summary of the classes and their respective fault cases is provided in Table 1. To facilitate model evaluation, an 80/20 split between the training and test sets was employed, resulting in 46,806 samples in the training set and 11,703 samples in the test set.

The SDD dataset is particularly suitable as it offers multiple defect classes with diverse characteristics in the area of industrial data, which facilitates AE generation. At the same time, it is not too high dimensional, which keeps the computation times within reasonable limits.

Table 1: Error indicators of the individual classes of the SDD dataset. Class 1 is error-free, and the remaining ten are error cases. Equal classes, such as class 4 and 5, are not identical; they differ in the level of error, for example, the angle of the axis inclination.

Class	1	2	3	4	5	6	7	8	9	10	11
Bearing Failure	0	0	0	0	0	1	1	1	1	1	1
Axis Inclination	0	0	1	1	1	0	1	1	0	1	1
Shaft Misalignment	0	1	0	1	1	1	0	1	1	0	1

Model Implementation

A deep neural network (DNN) with four hidden layers is deployed for evaluation. The input layer comprises 48 neurons, followed by hidden layers with 590, 1180, 2360, and 590 neurons, respectively, and an output layer with 11 neurons. The architecture is based on [7]. Although smaller DNNs can classify the SDD dataset [29], the selection should prevent a bottleneck of capacity as discussed in [20] and exclude this influence. After each hidden layer, Batch Normalization (BatchNorm) [31] is applied followed by the Rectified Linear Unit (ReLU) activation function. Dropout [30] with a dropout rate of 20% is utilized after each ReLU layer to prevent overfitting. The output layer employs a linear layer with Softmax for classification. Cross-entropy loss is used with the Adam optimizer [17], configured with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$, and a learning rate of 10^{-4}. The implementation is carried out using the PyTorch framework [32]. A min-max scaler is used for preprocessing to scale feature values to the range $[0,1]$.

The generation of AEs is performed using the adversarial-robustness-toolbox [33] and advertorch [34]. For the L_∞ attacks PGD and FGSM, the maximum perturbation is controlled by the explicit parameter ε, which is determined as described in the following section. The control parameter Γ for JSMA, regulating the proportion of features that can be altered, is set at 14.5% following [16]. As DeepFool and C&W do not have explicit attack budget parameters, they are limited to 10 iterations, equivalent to the number of iterations in the

Table 2: The perturbation budget influences adversarial robustness. A perturbation per feature of 1% of its range leads to adversarial robustness close to the accuracy of clean data. An increase in the attack budget significantly reduces adversarial robustness, suggesting a potential change in the true class.

Attack	Perturbation in %	Clean Accuracy	Adversarial Accuracy
PGD	1	0.99	0.92
PGD	2	0.96	0.74
PGD	3	0.93	0.60
PGD	4	0.99	0.22

PGD algorithm with the chosen ε. Further details, code, and parameter settings can be accessed here[1], allowing for result reproduction and further research.

4.2 Results

Attack Budget for L_∞ Norm Attacks

To establish an attack budget for L_∞ norm attacks, specifically FGSM and PGD, we selected the stronger of the two variants, i.e., PGD, and tested it with various parameter values. Table 2 presents the results of these tests. All the attacks listed in Table 2 were capable of reducing the accuracy of models without adversarial training by at least 60%. It was observed how long adversarial training remained effective. A significant drop in adversarial robustness can be interpreted as an indication that the true class membership has changed, rendering the model incapable of learning the data distribution. Based on the results in Table 2, a maximum perturbation of 1% of the feature range was set as the attack budget for the L_∞ norm attacks.

[1] https://ds-juist.init.th-owl.de/j.knaup/ciworkshop

Table 3: Accuracy results of the cross-comparison of different AE generation methods. Clean indicates the usage of only original training and test data, respectively. The remaining training methods utilize a mix of the data in the training phase, and the remaining attack methods are evaluated on the manipulated data exclusively.

Adversarial Training Method	Adversarial Attack Method					
	FGSM	JSMA	DeepFool	C&W	PGD	Clean
FGSM	0.92	0.21	0.08	0.74	0.91	0.99
JSMA	0.40	0.33	0.26	0.27	0.39	0.43
DeepFool	0.65	0.12	0.24	0.20	0.57	0.99
C&W	0.74	0.09	0.09	0.12	0.67	0.97
PGD	0.93	0.21	0.09	0.74	0.92	0.99
Clean	0.41	0.07	0.01	0.35	0.32	0.99

Cross-comparison of AE Generating Methods

Table 3 shows that training with JSMA generated AEs, significantly affects the accuracy on the original data. PGD and FGSM achieve almost identical values and are robust to themselves and each other. JSMA and DeepFool reduce the accuracies of the other models the most and C&W achieves an increased robustness against FGSM and PGD. A detailed discussion is provided in the next section.

5 Discussion

The results presented in Table 3 align with findings in the literature, where FGSM and PGD are commonly used for adversarial training on image data [23]. The fact that PGD differs only slightly from FGSM may be attributed to factors such as the number of iterations or the limited attack budget. JSMA, originally designed for gray-scale images like the MNIST dataset [35], poses certain challenges when applied to tabular data. By searching individual pixels and increasing or decreasing their values depending on the sign of the adjustment parameter, JSMA sets individual features here to 0 or 1, respectively. This leads to unrealistic inputs, which on the one hand are difficult to classify for

other models, but on the other hand it is not reasonable to enrich the training set with them. DeepFool is more suitable in this respect, since the respective decision boundary is only slightly exceeded. The C&W attack, known for its high success rate in finding minimal AEs [11], may benefit from further hyperparameter tuning but at the cost of increased computation time.

However, this study's approach has yielded the expected results. The preprocessing enabled the application of various algorithms and the employment of the original data and the manipulated data prioritized the clean as well as adversarial accuracy. The warm-up phase added stability to the training process. A similar approach to this is curriculum-based learning [36], where attack strength adapts and increases as the training progresses.

Nevertheless, challenges persist in distinguishing between adversarial examples and points at which the ground truth has fundamentally changed. While approaches like [37] improve PGD by considering the proximity of each input to the decision boundary when applying perturbations, they still do not identify the true tipping point. Additionally, selecting an appropriate distance measure for this assessment remains an open question. Even though L_∞ norm attacks show promise, perturbations with the same L_p norm can have vastly different effects. Comparing methods that employ different distance metrics poses particular challenges. Furthermore, adversarial training defined as min-max problem inherently lacks robustness guarantees due to the non-convex nature of deep neural networks, which makes it intractable to find a global optimum [23].

Moreover, the adversarial mapping $\mathscr{A}(\cdot)$ in this paper has been limited to additive perturbations δ. Future research directions may involve exploring the use of generative adversarial networks (GANs) [38, 39] to create adversarial examples. This approach could generate AEs with semantic information, leading to more natural and meaningful adversarial examples, commonly referred to as semantic adversarials in the literature [41, 40].

6 Conclusion and Outlook

In this paper, we presented an approach that extends the application of adversarial attacks to tabular data for adversarial training. We began by providing an overview of various adversarial example generation methods, followed by the introduction of a straightforward preprocessing technique and training stabilization mechanisms. Subsequently, we conducted a comprehensive cross-comparison of popular attack methods, including FGSM, JSMA, DeepFool, C&W, and PGD, on an industrial dataset.

The results of our study validate existing findings in the literature, demonstrating the effectiveness of FGSM and PGD for adversarial training. However, our investigation also highlights the unique challenges posed by tabular data when employing methods like JSMA, which generate unrealistic inputs. The quest for a suitable distance metric remains a pivotal aspect of future research, as it not only determines the presence of adversarial examples but also serves as the foundation for method comparisons.

Looking ahead, the exploration of non-additive perturbations presents a promising avenue for the development of new adversarial example generation methods. Incorporating semantic contextual information into the generation process may yield more natural and meaningful adversarial examples, albeit with potentially higher L_p norm values. This shift toward semantically enriched adversarial examples could lead to advancements in the robustness of machine learning models, particularly in applications involving tabular data.

Acknowledgment

The authors acknowledge financial support by the project “SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems” (Grant ID NW21-059C), which is funded by the program “Netzwerke 2021” of the Ministry of Culture and Science of the State of Northrhine Westphalia, Germany.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc. 2012.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In: *9th International Conference on Learning Representations (ICLR)*. 2021.

[3] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. In: *Neural computation*, 9(8):1735–1780. 1997.

[4] OpenAI. GPT-4 technical report. *ArXiv*, 2023.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. Los Alamitos, CA, USA. June 2016.

[6] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In: *International Conference on Learning Representations (ICLR)*. 2014.

[7] Felix Specht, Jens Otto, Oliver Niggemann, and Barbara Hammer. Generation of adversarial examples to prevent misclassification of deep neural network based condition monitoring systems for cyber-physical production systems. In: *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pages 760–765. 2018.

[8] Felix Specht and Jens Otto. Hardening deep neural networks in condition monitoring systems against adversarial example attacks.

In: *Machine Learning for Cyber Physical Systems*, pages 103–111. Springer, Berlin, Heidelberg. 2021.

[9] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2154–2156. New York, NY, USA. 2018.

[10] Naveed Akhtar, Ajmal S. Mian, Navid Kardan, and Mubarak Shah. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access*, 9. 2021.

[11] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In: *IEEE Symposium on Security and Privacy*, pages 39–57. 2017.

[12] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In: *International Conference on Learning Representations (ICLR)*. 2015.

[13] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In: *5th International Conference on Learning Representations (ICLR), Workshop Track Proceedings*. 2017.

[14] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In: *5th International Conference on Learning Representations (ICLR)*. 2017.

[15] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582. 2016.

[16] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In: *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. 2016.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In: *3rd International Conference on Learning Representation (ICLR)*. San Diego, CA, USA. May, 2015.

[18] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In: *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597. San Jose, CA, USA. 2016.

[19] Nicolas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In: *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3-–14. Dallas, Texas, USA. 2017.

[20] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In: *6th International Conference on Learning Representations (ICLR),Conference Track Proceedings*. Vancouver, BC, Canada. April 30 - May 3, 2018.

[21] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In: *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 274–283. Stockholm, Sweden. 2018.

[22] Dimitris Tsipras, Shibani Santurkar, Logan G. Engstrom, Alexander M. Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In: *7th International Conference on Learning Representations (ICLR)*. New Orleans, Louisiana, USA. May, 2019.

[23] Tao Bai, Jinqi Luo, Jun Zhao Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In: *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4312–4321. 2021.

[24] Vincent Ballet, Xavier, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, and Marcin Detyniecki. Imperceptible adversarial attacks on

tabular data. In: *NeurIPS 2019 Workshop on Robust AI in Financial Services: Data, Fairness, Explainability, Trustworthiness and Privacy*. Vancouver, Canada. 2021.

[25] Martyna Bator. Dataset for sensorless drive diagnosis. In: *UCI Machine Learning Repository*. https://doi.org/10.24432/C5VP5F . 2015.

[26] Detmar Zimmer, Christian Lessmeier, Kay Hameyer, Christelle Piantsop Mbo'o, and Isabel Coenen. Untersuchung von Bauteilschäden elektrischer Antriebsstränge im Belastungsprüfstand mittels Statorstromanalyse. In: *ant Journal - Anwendungsnahe Forschung für Antriebstechnik im Maschinenbau*, pages 8–13. 2012.

[27] Christian Lessmeier, Olaf Enge-Rosenblatt, Christian Bayer, and Detmar Zimmer. Data acquisition and signal analysis from Mmeasured motor currents for defect detection in electromechanical drive systems. In: *PHM Society European Conference*. 2014.

[28] Martyna Bator, Alexander Dicks, Uwe Mönks, and Volker Lohweg. Feature extraction and reduction applied to sensorless drive diagnosis. In: *22nd Workshop Computational Intelligence (VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik (GMA)*, pages 163–177. 2012.

[29] Anton Pfeifer and Volker Lohweg. Classification of faults in cyber-physical systems with complex-valued neural networks. In: *26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–7. Vasteras, Sweden. 2021.

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. In: *Journal of Machine Learning Research 15*, pages 1929-1958. 2014.

[31] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on Machine Learning*, pages 448–456. Lille, France. 2015.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance Deep Learning library. In: *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. 2019.

[33] Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molly, and Ben Edwards. Adversarial robustness toolbox v1.2.0. In: *Computing Research Repository (CoRR)*. 2018.

[34] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on PyTorch. *arXiv preprint arXiv:1902.07623*. 2019.

[35] Yann LeCun, Corinna Cortes, and Christopher Burges. MNIST handwritten digit database. Available: http://yann.lecun.com/exdb/mnist. 2010.

[36] Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint:1805.04807*. 2018.

[37] Yogesh Balaji, Tom Goldstein, and Judy Hoffman. Instance adaptive adversarial training: Improved accuracy tradeoffs in neural nets. *arXiv preprint:1910.08051*. 2019.

[38] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In: *Advances in Neural Information Processing Systems*, volume 27, pages 2672–2680. Curran Associates, Inc., 2014.

[39] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial

networks. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3905–3911. Stockholm, Sweden. 2018.

[40] Tommaso Dreossi, Somesh Jha, and Sanjit A. Seshia. “Semantic adversarial deep learning”. In: *Computer Aided Verification: 30th International Conference (CAV) Proceedings, Part I 30*, pages 3–26, Springer. Oxford, UK, July 14-17. 2018.

[41] Hossein Hosseini and Radha Poovendran. “Semantic adversarial examples”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2018.