Engineering 19 (2022) 240–251

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
AI Energizes Process Manufacturing—Article

One-Variable Attack on the Industrial Fault Classification System and Its Defense

Yue Zhuo [a], Yuri A.W. Shardt [b], Zhiqiang Ge [a,*]

[a] State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China
[b] Department of Automation Engineering, Technische Universität Ilmenau, Ilmenau D-98684, Germany

ARTICLE INFO

Article history:
Received 28 January 2021
Revised 19 June 2021
Accepted 13 July 2021
Available online 3 June 2022

Keywords:
Adversarial samples
Black-box attack
Industrial data security
Fault classification system

ABSTRACT

Recently developed fault classification methods for industrial processes are mainly data-driven. Notably, models based on deep neural networks have significantly improved fault classification accuracy owing to the inclusion of a large number of data patterns. However, these data-driven models are vulnerable to adversarial attacks; thus, small perturbations on the samples can cause the models to provide incorrect fault predictions. Several recent studies have demonstrated the vulnerability of machine learning methods and the existence of adversarial samples. This paper proposes a black-box attack method with an extreme constraint for a safe-critical industrial fault classification system: Only one variable can be perturbed to craft adversarial samples. Moreover, to hide the adversarial samples in the visualization space, a Jacobian matrix is used to guide the perturbed variable selection, making the adversarial samples in the dimensional reduction space invisible to the human eye. Using the one-variable attack (OVA) method, we explore the vulnerability of industrial variables and fault types, which can help understand the geometric characteristics of fault classification systems. Based on the attack method, a corresponding adversarial training defense method is also proposed, which efficiently defends against an OVA and improves the prediction accuracy of the classifiers. In experiments, the proposed method was tested on two datasets from the Tennessee–Eastman process (TEP) and steel plates (SP). We explore the vulnerability and correlation within variables and faults and verify the effectiveness of OVAs and defenses for various classifiers and datasets. For industrial fault classification systems, the attack success rate of our method is close to (on TEP) or even higher than (on SP) the current most effective first-order white-box attack method, which requires perturbation of all variables.

1. Introduction

In the field of fault classification, numerous data-driven machine learning methods have been proposed and have achieved good accuracy [1,2]. Some of them are derived from traditional learning machines, such as support vector machines (SVMs) for multi-fault classification [3], linear dynamic systems (LDSs) for handling the dynamic fault classification problem [4], transfer learning based on linear discriminant analysis [5], and ensemble methods, such as SVM-forest [6]. Deep learning has been extensively researched for fault classification. Owing to the rapid increase in the collected data size and computation capacity, model performance can be improved. Some examples of such improved methods include deep convolutional neural networks (CNNs) with time–frequency representation [7], bidirectional deep recurrent neural networks (RNNs) for sequential fault detection [8], and sparse stacked auto-encoders (AEs) considering dynamic information [9].

Despite the high classification accuracy of machine learning, recent research studies have shown an intriguing weakness that widely exists among these classifiers: Small imperceptible perturbations on input samples will cause the classifiers to output incorrect predictions with high confidence [10–12]. Many previous studies have investigated adversarial attacks of deep models for image classification. In 2013, Szegedy et al. [13] first discovered this property of deep networks. A meaningful work is the fast gradient sign method (FGSM) by Goodfellow et al. [14], which used the increasing gradient direction of samples to craft adversarial samples and trained robust deep neural networks (DNNs) with

* Corresponding author.
E-mail address: gezhiqiang@zju.edu.cn (Z. Ge).

https://doi.org/10.1016/j.eng.2021.07.033

them. Based on FGSM, some methods iterate more than once to calculate the perturbations on the adversarial samples and achieve better performances, such as project gradient descent (PGD) [15], free adversarial training (FreeAT) [16], and you only propagate once (YOPO) [17]. Unlike the abovementioned approaches that modify every pixel, Su et al. [18] claimed that they could successfully fool three different networks with the tested images where a single pixel per image was changed. Papernot et al. [19] also created an adversarial attack by restricting the number of pixels of the perturbations.

Likewise, some studies have studied the safety issues of conventional classifiers. Barreno et al. [20] examined the security of a Bayesian-based method for spam detection software. Biggio et al. [21] attacked SVM classifiers. Hu and Tan [22] used generative adversarial networks to synthesize adversarial samples and successfully attack malware detectors in the form of random forest, linear regression, and decision trees.

As for the defense against adversarial samples, the most common method is adversarial training that adds adversarial samples during the training process, making the model more robust and regularized [13,14,23]. Another method is to modify networks using contractive or denoising networks [24] or masking gradients to adversaries [25]. Alternatively, training an auxiliary model to detect adversaries is also an effective defense method [26].

Because the industrial fault classification system is highly safety-critical, where misclassification may lead to severe undesired outcomes [27,28], it is vital to study the issue of adversarial attacks and defense on industrial systems. Therefore, this study proposes a black-box attack method on an industrial fault classification system by perturbing only one variable, which is called a one-variable attack (OVA). The major motivations and distinguishing features of our attack method are as follows:

(1) **Attack scenarios:** In this study, we mainly consider the situation in which industrial fault diagnosis systems are under malicious attacks. One of the possible attack scenarios is that a malicious hacker gains access to the industrial system, either through physical approaches to real-world sensors or through electronic technologies connected to the smart diagnosis system. Perturbing an industrial manufacturing process, by changing the components in streams for instance, will cause serious damage to the entire system, but this is difficult to detect. Hence, this study investigates the most concealed attack method for intelligent fault diagnosis systems, where a hacker can set a tiny offset on a single sensor, such as temperature or visual measurement. According to our experimental results, these undetectable OVAs seriously threaten fault diagnosis systems.

(2) **One-variable:** Unlike process faults that may be related to multiple sensors in industrial systems [29], malicious attacks are more likely to occur on a single sensor measurement; thus, one variable, rather than multiple sensors, is perturbed simultaneously. Furthermore, from a theoretical perspective, OVA makes it easier to provide the geometric boundaries in the fault classifier input space by analyzing each fault variable separately, which provides a deeper insight into the fault classification system.

(3) **Black-box:** In adversarial learning, the black-box attack mode can only access the output of classification models and does not require any internal information of models, such as structures or parameters. The black-box attack property allows for very general results that can be applied to different circumstances, where the innermost fault classification models are not usually accessible for security reasons. With the black-box property, OVA is also capable of attacking indifferentiable classifiers without any gradient information.

Furthermore, this study adopts a selection order that can make the perturbations invisible in the dimensional reduction visualization space, which is a standard way to observe abstract industrial data. This is achieved by using the Jacobian matrix of the reduction mapping on variables, and the variables with smaller derivatives have a higher priority to be perturbed.

There is also an adversarial training method to defend against adversarial attacks. The proposed defense method trains classification models using adversarial samples that perturb the partial gradient directions. Our defense method is robust in detecting OVA, and at the same time it improves the classification accuracy, which is hardly achieved by some existing adversarial training methods.

In summary, the main contributions of this study are as follows:

- A black-box OVA method, which is more applicable to actual industrial fault classification scenarios, is proposed. An OVA perturbs only a single variable and attacks different types of fault classifiers without internal information, and the produced adversarial samples are undetectable.
- Insight into industrial variables and fault classification is provided by evaluating the vulnerability and geometry of variables and faults using an OVA.
- An adversarial training method to defend against OVAs, which provides a robust classification system with higher accuracy, is presented.

To the best of our knowledge, this is the first time that adversarial attack and defense methods have been proposed and analyzed for industrial fault classification systems. The remainder of this study is organized as follows, Section 2 presents the OVA method and the corresponding evaluation results. In Section 3, the vulnerability and robustness of industrial data are explored at the variable and fault levels, and an intuitive illustration of classification boundaries provides an insight into fault classification systems. In Section 4, the defense method against OVA is introduced. Finally, the conclusions are presented in Section 5.

2. One-variable attack

In this section, the methodology and evaluation of OVA are presented in detail. For the overall intuitive illustration, based on the Tennessee–Eastman process (TEP) dataset (see Section 2.2 for dataset details), Fig. 1, which is a plot of data points of some fault types, reports the attacked variables with corresponding classification results. The figure shows that the OVA method can threaten fault classifiers by perturbing only one variable of the samples with high confidence. Taking the DNN classifier for samples of fault 6 as an example (subfigures on the left), attacking variable 20 (resp.[†] 17) with 10% (resp. 20%) value offset misled the fault classifier to incorrectly predict them as fault 17 (resp. 8), with a nearly 100% confidence. Meanwhile, it can be seen that the drift of crafted adversarial samples is subtle and unnoticeable in the low-dimension visualized space, which is attributed to the perturbed variable searching order proposed by us.

2.1. Methodology

The OVA approach crafts adversarial examples by attacking original samples with perturbation vectors, which are constrained below certain values in specific norm measurements. First, classifiers for the industrial data are trained, and the classifier with the best generalization on the test set, which is denoted as f, is chosen to be attacked. The n-dimensional fault samples $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ correctly predicted by f are perturbed. It should be noted that the focus was on the samples that were correctly classified in the test set. The perturbation is defined as

[†] "resp." is the abbreviation of "respectively," which means the value inside brackets can respectively substitute the value outside, so as the resp. in the following.

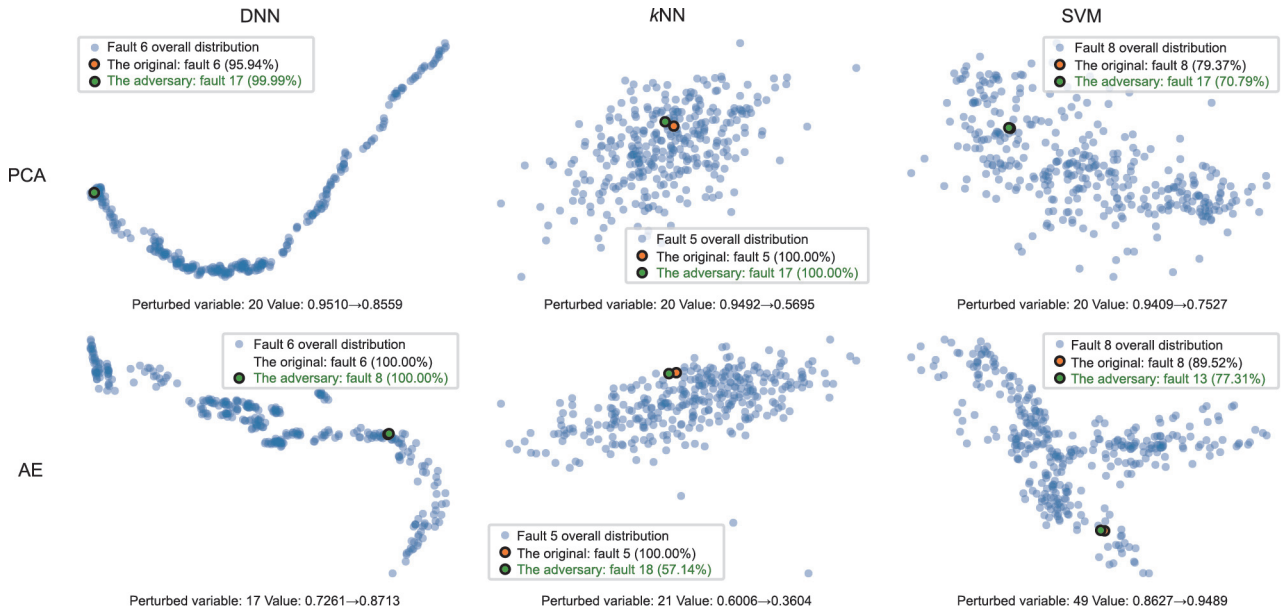

Fig. 1. OVA samples visualized. The light blue points present whole samples of a certain fault type, the orange points are for the original samples, and the greens are for the adversarial samples that perturb one variable on the original samples (some point pairs are so close that they overlap). The comments at the bottom of each figure give the attacked variable with the original and targeted value. The legends show the original and targeted predictive fault type and the confidence of the classifiers. The rows correspond to two-dimensional (2D) reduction skills (principal component analysis (PCA, top) and AE (bottom)); and columns correspond to different classifiers (DNN, k-nearest neighbor (kNN, k = 7), SVMs).

$\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_n)$ of the same size as the input samples. Thus, the OVA is

$$f(\boldsymbol{x}+\boldsymbol{\eta}) \neq f(\boldsymbol{x}) \tag{1}$$

$$\text{subject to } \|\boldsymbol{\eta}\|_0 = 1, \text{ and } \|\boldsymbol{\eta}\|_1 \leq \varepsilon$$

where ε is a hyperparameter and $\|\boldsymbol{\eta}\|_p$ ($p = 0$ or 1) is the l_p norm of $\boldsymbol{\eta}$. The l_0 norm counts the total number of nonzero elements of a vector, which is used to constrain the number of variables to be modified. The l_1 norm controls the distance between the adversarial and original samples below hyperparameter ε, which can be adjusted in the interval [0, 1]. To better evaluate the deviation of adversarial samples, the distortion in the perturbation ratio form is replaced with the absolute value ε in a part of later experiments. For the perturbed variable x_{pert}, the distortion is defined as the ratio of perturbation in the original variable value, which can be formulated as follows:

$$\text{Distortion} = \varepsilon / x_{\text{pert}} \tag{2}$$

To make the displacement of the adversarial samples in the visual space as small as possible, we calculate the column l_1 norm of the Jacobian matrix on a dimensional reduction mapping function to guide the priority during the variable search. The Jacobian matrix consists of the first partial derivatives between the input and output of a multivariate function. For the dimensional reduction visualization mapping function $\boldsymbol{z} = F(\boldsymbol{x})$, $F: \mathbb{R}^n \to \mathbb{R}^m$ ($m = 2$ or 3), the Jacobian matrix $\boldsymbol{J}_F$ and its column l_1 norm $\boldsymbol{v}$ can be written as

$$\boldsymbol{J}_F = \begin{bmatrix} \frac{\partial z_1}{\partial x_1} & \cdots & \frac{\partial z_1}{\partial x_n} \\ \frac{\partial z_2}{\partial x_1} & \cdots & \frac{\partial z_2}{\partial x_n} \\ \frac{\partial z_3}{\partial x_1} & \cdots & \frac{\partial z_3}{\partial x_n} \end{bmatrix} \tag{3}$$

$$\boldsymbol{v} = \|\boldsymbol{J}_F\|_{1-\text{col}} = \begin{bmatrix} \sum_{i=1}^{3} \frac{\partial z_i}{\partial x_1} & \cdots & \sum_{i=1}^{3} \frac{\partial z_i}{\partial x_n} \end{bmatrix} \tag{4}$$

Sorting the elements of $\boldsymbol{v}$ in ascending order yields a variable-order sequence for searching during the attack. The variable of the smaller gradient with respect to the low-dimensional space is first perturbed to attack the classifiers, which guarantees that the generated adversarial samples drift the least in the visualization space.

Algorithm 1 shows the procedure for OVA.

Algorithm 1. One-variable attack for fault classification.

d is the predefined distortion; $\boldsymbol{v}^s$ is the index vector indicating the variable position in ascending order by sorting $\boldsymbol{v}$ (e.g. $\boldsymbol{v}^s(0)$ denotes the position index of the minimal variable, while $\boldsymbol{v}^s(n-1)$ denotes that of the maximal one);

Input: $\boldsymbol{x}, f, d, \boldsymbol{v}^s$

Output: adversarial samples $\boldsymbol{x}^*$

initial $\boldsymbol{x}^* \leftarrow \boldsymbol{x}, i = 0$

repeat

$k = \boldsymbol{v}^s(i)$

$\varepsilon = d \times \boldsymbol{x}(k)$

$\boldsymbol{x}^*(k) = \boldsymbol{x}(k) \pm \varepsilon$

clip $\boldsymbol{x}^*(k)$ to [0, 1]

$i = i + 1$

until $f(\boldsymbol{x}^*) \neq f(\boldsymbol{x})$ or $i = n$

2.2. Evaluation

To comprehensively verify the effectiveness of an OVA on industrial fault classification systems, we chose two industrial datasets: the TEP [30] and steel plates (SP) [31]. TEP is a public benchmark dataset for the development, study, and evaluation of industrial processes. The TEP dataset consists of 52 variables, 28 fault types, and one normal working condition. A flowchart of the TEP is presented in Fig. 2. In the experiments, we chose the data of the first 21 fault types along with the normal working condition, each of which had approximately 500 samples. The SP dataset came from the research by Semeion [31], Research Center of Sciences of Communication, seeking to correctly classify the type of surface defects in stainless SP. There are a total of 1941 samples

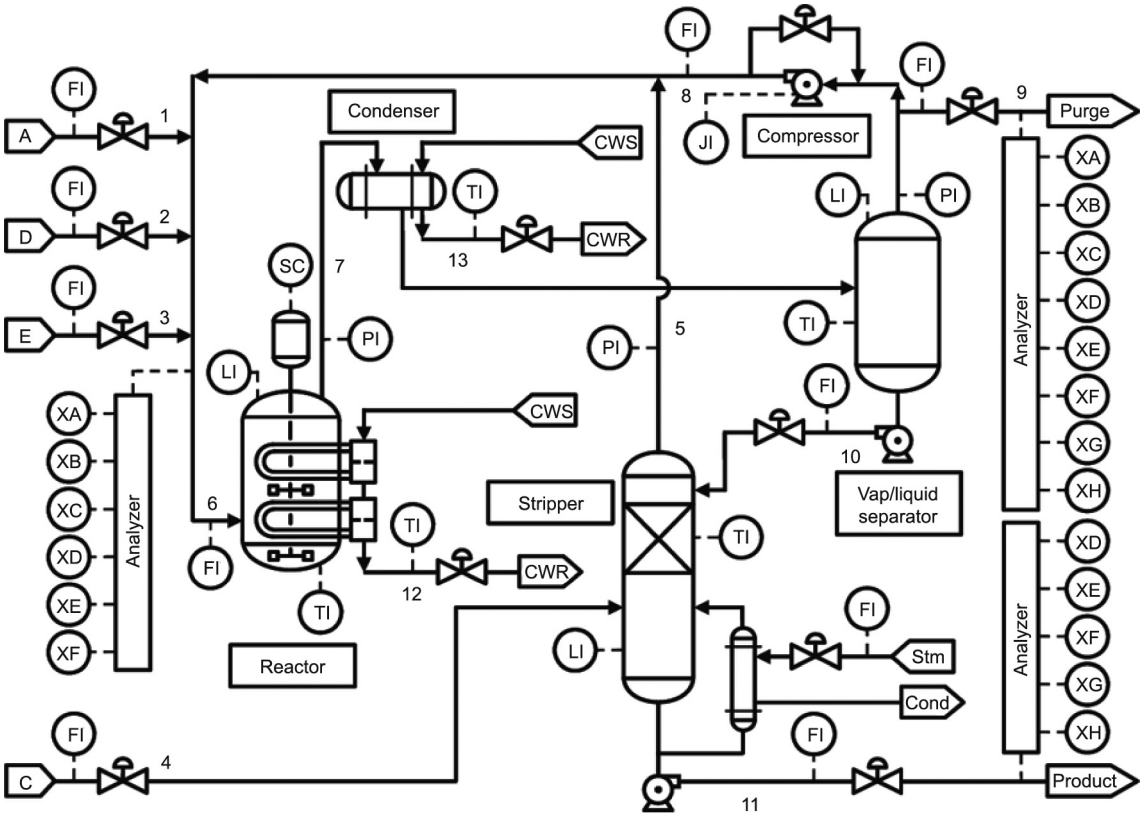

Fig. 2. The flow chart of TEP [30]. FI: flow indicator; Stm: steam; Cond: condenser; LI: level indicator; PI: pressure indicator; TI: temperature indicator; JI: power indicator of compressor; SC: synchrocyclotron; XA, XB, XC, XD, XE, and XF: analysis of Component A, B, C, D, E, and F, respectively; CWS: chilled water supply; CWR: chilled water return.

and seven fault types without the normal type in the SP dataset. Table 1 summarizes the attributes and fault types in the SP dataset. The 27 attributes of SP dataset can be categorized into two types according to their source. Using visual skills, the geometric shape of the steel defect and its outline were extracted. In addition, the intrinsic properties of the steel and conveyer are also described. Practically, visual sensors are easier to be attacked, and small perturbations may have a significant impact on the measurements. Hence, the SP dataset was chosen for the evaluation of OVA in fault classification systems.

In the experiments, we first normalized all the samples to range from 0 to 1 and split the dataset into a test set and a training set in a ratio of 3:7. Then, the training set was used to train the classifiers, and the correctly predicted samples in the test set were selected to be perturbed to attack the classifiers. We selected three types of classifiers: DNN representing the deep learning model, SVM, and k-nearest neighbor (kNN), for the conventional machine learning models. Table 2 reports the classification results, including the classification accuracy and confidence. Classification accuracy is the major metric to measure how many test samples are correctly predicted, and the confidence can indicate how confident the classifier f is on its predictions, which can be formulated as

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} \mathbb{I}\left\{f(\boldsymbol{x}_{(i)}) = y_{(i)}\right\}}{N} \tag{5}$$

$$\text{Confidence} = \frac{\sum_{i=1}^{c} \Pr_f(y_{(i)}|\boldsymbol{x}_{(i)})}{c} \tag{6}$$

where N is the sample number in the test set, $\mathbb{I}$ is the indicator function, and $\{\boldsymbol{x}_{(i)}, y_{(i)}\}$ is a pair of samples and their true labels. Confidence is calculated on the correctly predicted test set, the sample number of which is $c = N \times$ accuracy.

2.2.1. Effectiveness of the attacks

In this section, we study the extent to which an OVA can mislead fault classification. The results were obtained by experiments with two industrial fault data and three classifiers. Because the order of variable search is not the focus of this subsection (that will be discussed in the following subsection), all OVA experiments are performed with a constant random sequence of searching variables. The following metrics were used for the adversarial attack evaluation:

Table 1
Summary of the SP fault dataset [31].

Attributes type	Attributes details	Fault types
Visual location (X, Y)	min, max, perimeter	1. pastry; 2. Z-scratch; 3. K-scratch; 4. stains;
Visual luminosity	min, max, sum	5. dirtiness; 6. bumps; 7. others
Visual areas	pixels, sigmoid, log	
Visual index	edges, empty, square, outside, orientation	
Steel	type, thickness	
Conveyer	length	

Table 2
Classification results on the test set.

Dataset	Accuracy (%)			Confidence (%)		
	DNN	*k*NN	SVM	DNN	*k*NN	SVM
TEP	75.8	57.4	56.2	81.7	62.0	54.0
SP	78.4	68.6	69.1	85.6	70.9	69.5

- **Success rate:** In an adversarial attack, the success rate is the proportion of the number of successfully attacked samples among c attackable samples (defined in Eq. (6)), which can be formulated as

$$\text{Success rate} = \frac{\sum_{i=1}^{c} \mathbb{I}\{f(\boldsymbol{x}_{(i)}+\boldsymbol{\eta}) \neq y_{(i)}\}}{c} \tag{7}$$

- **Confidence:** Confidence in adversarial attacks is slightly different from that in classification. The confidence here is computed based on the incorrect predictive probability of successfully perturbed samples. A higher confidence means that the classifier predicts the adversarial input samples to be the incorrect fault types with a higher probability. The formulation is

$$\text{Confidence}_{\text{adv}} = \frac{\sum_{i=1}^{s} \Pr_f(f(\boldsymbol{x}_{(i)}+\boldsymbol{\eta}) \mid \boldsymbol{x}_{(i)}+\boldsymbol{\eta}}{s} \tag{8}$$

where $s = c \times$ success rate. In the following, all confidence metrics in the reported figures are defined in Eq. (8) for the adversarial attacks.

First, we verify that, for the classifiers of two different industrial datasets, only one variable of the samples needs to modified. In this case, the classifiers can successfully attack and predict the wrong fault types with relatively high confidence. Fig. 3 shows the results.

According to the success rate, DNN is the most vulnerable among the three classifiers mainly because of the depth characteristics of deep models, which lead to significant amplification of small perturbations. For TEP, even when the distortion is relatively small, the success rate is considerable (10% distortions (resp. 20%) yielded a success rate of 36.5% (resp. 62.6%)). On the other hand, the other two traditional classifiers are slightly more robust than DNN but still succumb to OVA, especially when distortion reaches a high level. The confidence metric indicates that the DNN is generally more vulnerable to threats. The probability output by *k*NN and SVM is nearly 50%, despite the increase in distortion and different datasets. On the contrary, the confidence of DNN overwhelms the other two in most cases. However, for SP, when the distortion is less than 20%, the exception is that the confidence of SVM is higher than that of DNN. This interesting observation indicates that although only a very small number of samples can be attacked, the successfully attacked samples gain high confidence in SVM. In the comparison of the two industrial datasets, TEP is much more vulnerable than SP for all three classifiers. The main reason is that the classifiers on SP are much more confident and accurate (as shown in Table 2), which corresponds to more robust models. Therefore, larger perturbations are required to successfully attack the classifiers for SP. This rule is confirmed by the inner data analysis in Section 3.

Next, we compare our methods with some competitive attack methods in the adversarial field, FGSM, and PGD. Both are white-box methods, and they calculate the adversarial samples based on the gradient. Because they perturb all variables, for fair

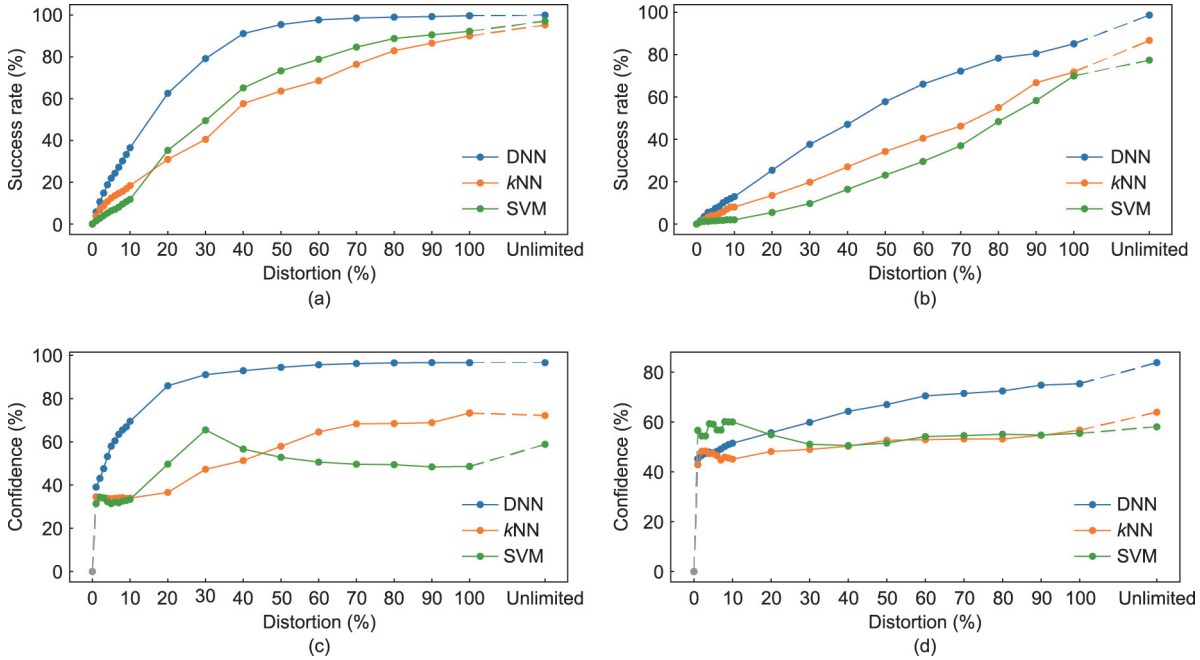

Fig. 3. OVA results. The figures show two metrics, (a, b) success rate and (c, d) confidence of two datasets, (a, c) TEP and (b, d) SP with regard to the distortion. The distortion varies from 0 to unlimited, where the unlimited stands for perturbing the variable to the maximal bounds [0, 1]. †

† For the variables less than 0.5, the maximal distortions are greater than 100%. For example, the maximal distortion on the variable of value 0.2 is 400% ((1 − 0.2)/0.2).

competition, the l_2 norm of the distance between the adversary and the original is used as the limitation of perturbations. The attack results are compared under perturbations with the same l_2 norm, and the DNN classifier is attacked. According to Figs. 4(a) and (b), for industrial fault classification, with the identical deviation distance limitation, perturbing only one variable can obtain a similar or even higher attack success rate than perturbing every variable. Compared with PGD, which is considered the strongest first-order attack method [15], OVA has a significant advantage in the success rate with the SP dataset. However, the confidence value of OVA was not better than that of the other two methods.

2.2.2. Effectiveness of variable searching order

This subsection demonstrates how the search order of variables during an attack can influence the distribution of adversaries in the visualization space. The experiments attack the DNN classifier without distortion limitation and apply two-dimensional (2D) reduction skills: principal component analysis (PCA) and AE. Random and Jacobian-based variable searching orders are compared, as shown in Fig. 5. Searching variables from the smallest gradient direction of reduction mapping can merge the adversarial samples into the original distribution, whereas random search makes the adversaries visually distinguishable.

In addition, the average l_2-distances between the adversaries and the original are calculated to compare two variable searching methods. As can be seen from Table 3, for the Jacobian-based variable search, the deviation distances in the visualization space significantly decrease.

3. Industrial data vulnerability analysis

The previous results on different classifiers and datasets show that industrial fault classification systems can be attacked with only one variable and that OVA is a general and competitive attack method. Furthermore, we explore the vulnerability of industrial data at the variable and fault levels. Our goal is to provide insight into how the variables and their fault types affect the vulnerability of the overall model.

All the vulnerability insights are analyzed (with all the reported figures in this section) under the scenario of OVA on the DNN classifier and the TEP dataset. DNN and TEP are the most mainstream classifiers and industrial datasets, and this combination is representative.

3.1. Fault variable study

Unlike in the previous section, we consider all variables in all correctly predicted fault samples to explore which variables and fault types are vulnerable. First, a set of the perturbation limit values $\varepsilon \in \{0.01, 0.02, ..., 0.19, 0.2, 0.3, ..., 1.0\}$ (the hyperparameter in Eq. (1)) is designed to test the minimum perturbation on a certain variable, which can successfully attack the classifier. Because the attack success rate rises rapidly under the smaller limit values of perturbation, we designed a smaller test ε value space (0.01) when less than 0.2.

The average minimum perturbation of each variable with regard to faults is shown in Fig. 6, where a small perturbation value indicates that a slight change in these variables can make the classifier predict incorrectly on this fault sample, and vice versa. Because some variables of fault samples cannot be successfully attacked, the attack success rate of fault variables is also reported in Fig. 7, which indicates that perturbing this variable leads to the vulnerability of the sample partitions. Two heat maps are negatively correlated, and they illustrate the vulnerability of the variables per fault type. Because the success rate with maximal perturbation tends to represent the variables that are difficult to attack, the success rate with a relatively small perturbation (0.1) limitation is also plotted in Fig. 8, which emphasizes the vulnerable variables. Moreover, the average minimal perturbations along the rows and columns in Fig. 6 are computed with respect to variables and faults, which are plotted in Fig. 9.

From the rows of the three heat maps from Figs. 6–8 and Fig. 9(a), fault 7 is the most robust in the classification of TEP, whereas faults 15, 16, and 21 are easier to craft adversarial samples. This is correlated with the confidence of the classifier in these faults. DNN has much more confidence for fault 7 and less confidence for

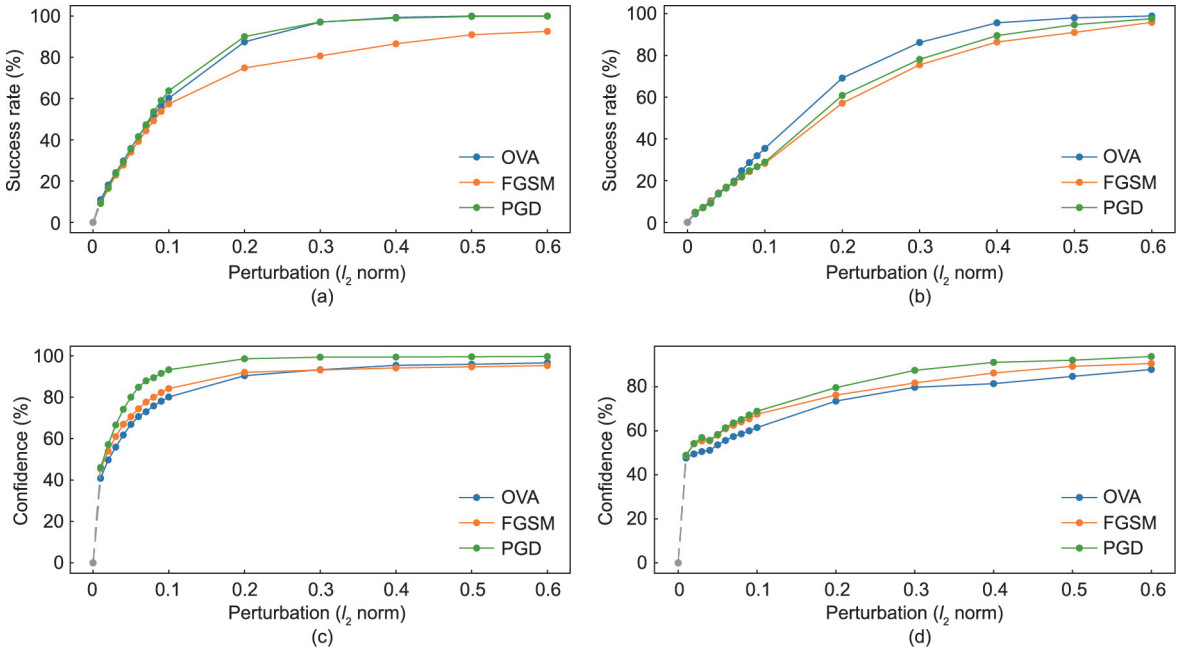

Fig. 4. Comparison of adversarial attack methods on the DNN classification model. The figures show two metrics, (a, b) success rate and (c, d) confidence of two datasets, (a, c) TEP and (b, d) SP with regard to the l_2 norm of the perturbations.

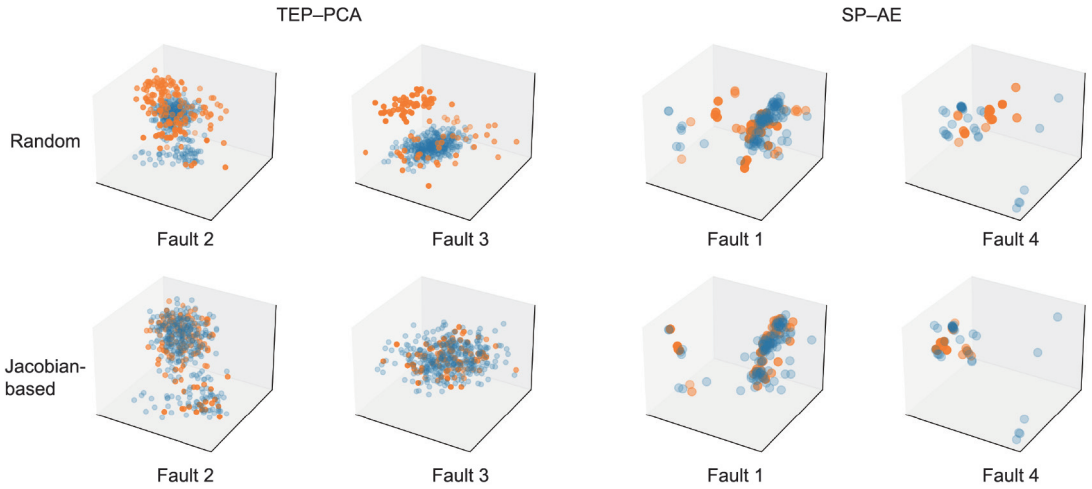

Fig. 5. Visualization of adversarial samples using the random and Jacobian-based variable searching order. Blue points represent the overall dataset distribution, and orange points represent the generated adversaries. The left half is the fault data from the TEP with PCA reduction, and the right half is the SP dataset with AE reduction. The upper row is adversaries using the random search, and the bottom is using the Jacobian-based search.

Table 3
Distances between adversarial and original samples in the reduced dimensional space.

Searching method	TEP–PCA	TEP–AE	SP–PCA	SP–AE
Random	0.1095	0.1725	0.1356	0.3521
Jacobian-based	0.0068	0.0702	0.0203	0.1342

vulnerable fault types. This conforms to the DNN's confusion matrix on the TEP test set (Fig. S1 in Appendix A), where vulnerable faults are also difficult to classify.

Based on the columns of the three heat maps and Fig. 9(b), variables 17 and 48 are much more vulnerable than others, whereas variables 3 and 26 are relatively more robust to perturbations. This corresponds to the loss gradient of the classifier for each variable. The heat map is shown in Appendix A Fig. S2. The gradients from the loss function of the DNN to variables 17 and 48 are much more significant than the others, which implies that perturbations on these variables will significantly impact the classifier.

3.2. Fault pair study

This subsection analyzes vulnerability at the fault level. Let the fault pair A–B be defined as: Fault A is the true fault type and fault B is the wrong type that an attacked classifier predicts. This means that, given the perturbed samples (of fault A) and an attack, the classifier gives the wrong diagnosis as fault B. As before, the experiments were performed with the TEP and DNN classifier.

The metrics are the minimum successful perturbation and the success rate, the results of which are calculated based on the original target fault pairs. We search all variables of all fault samples that can be attacked and record the smallest perturbation ε that can fool the classifier, the results of which are shown in Fig. 10(a). For the success rate of attacked samples, if fault A can be attacked to fault B on different variables, each sample of fault A only counts once for the number of successful attacks. Fig. 10(b) represents this in 0.1 perturbation limit to show the vulnerable fault pairs. For the success rate of attacked variables, it counts all the variables in one sample when the attack is successful, as shown in Fig. 10(c). As in Section 3.1, the average minimal perturbations are also computed for a rigorous analysis of the original and targeted fault pairs, as shown in Fig. 10(d).

For the original faults, the robustness and vulnerability are analyzed in Section 3.1, with respect to the variables. From the fault pair perspective, the result is very similar, and the only difference is that fault 4 becomes the hardest to be misclassified as most other faults.

For targeted faults, faults 10 and 11 are more vulnerable to being targeted, whereas faults 5 and 6 are less targeted. Combined

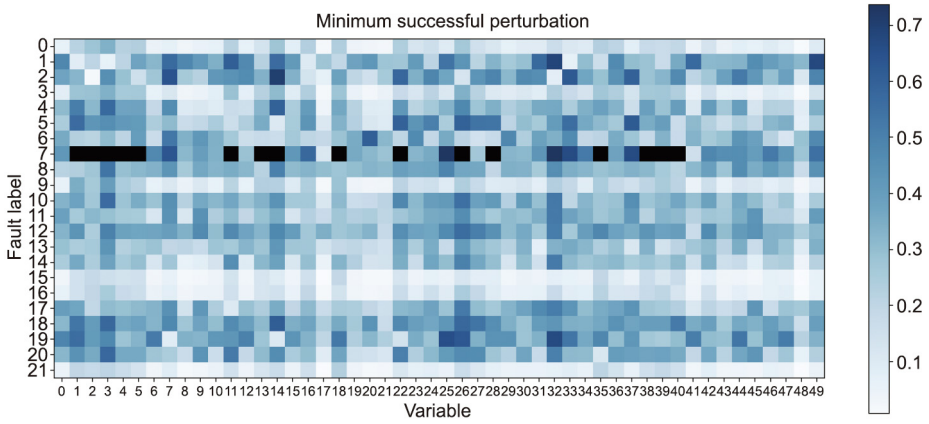

Fig. 6. Average minimum successful perturbation for the fault variables. The points with darker colors indicate the variables are harder to attack. The black points indicate the variables that cannot be attacked even once, at maximal perturbation.

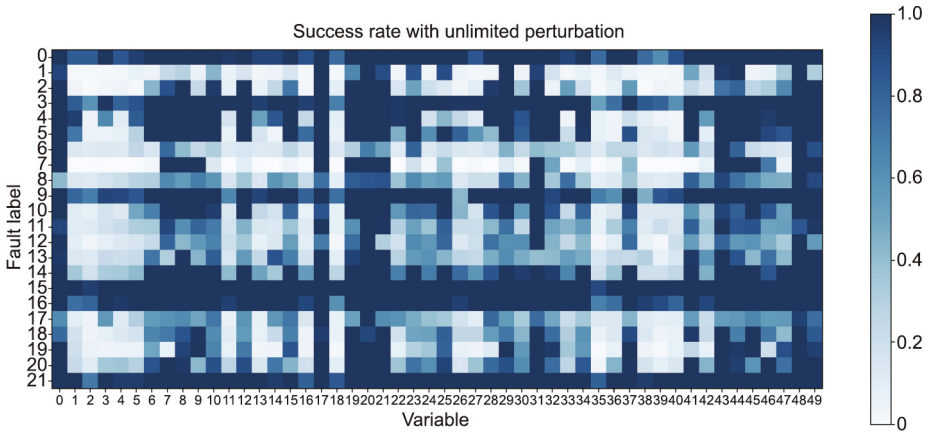

Fig. 7. Success rate with maximal perturbation for the fault variables. The points with lighter color indicate the variables are harder to attack.

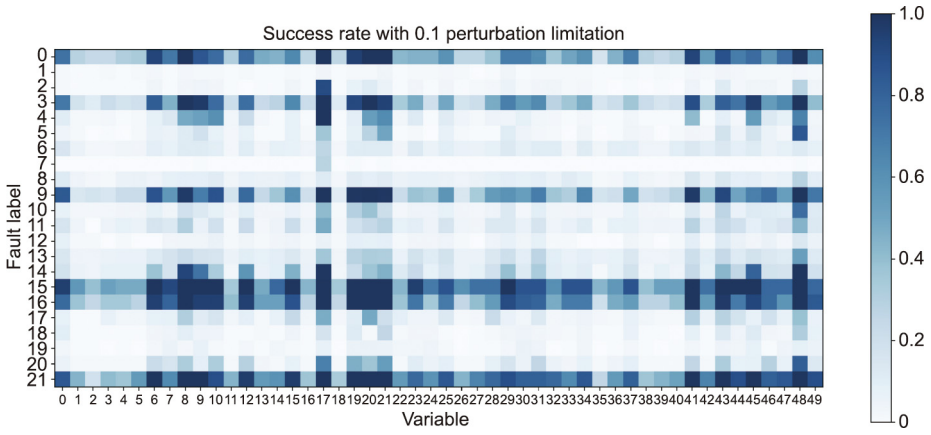

Fig. 8. Success rate with 0.1 perturbation for the fault variables. The points with lighter color indicate the variables are harder to attack.

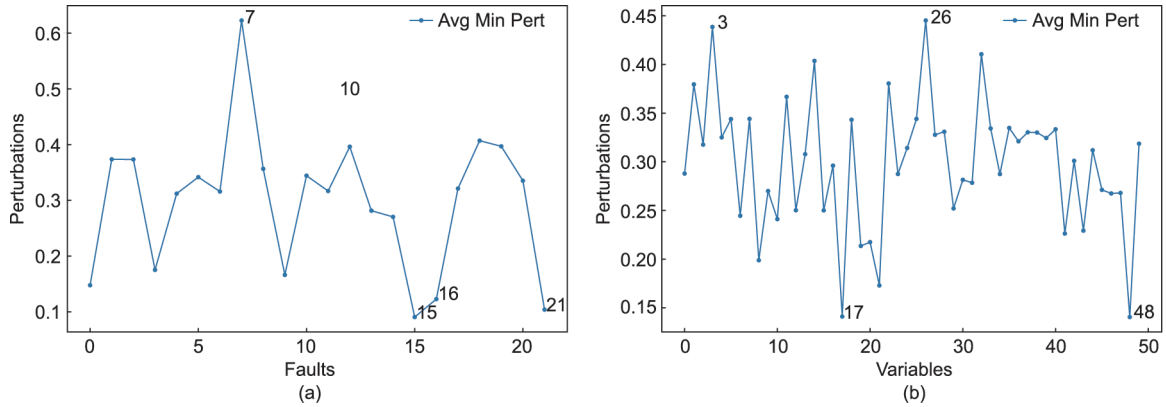

Fig. 9. Average minimal perturbations (Avg Min Pert) of (a) fault and (b) variable, which are the mean values along the rows and columns in Fig. 6.

with the classifier confusion matrix (Fig. S1), an interesting connection can be found: The robust and vulnerable faults are the faults with the highest and lowest classification accuracy in the test set, respectively. One possible geometric interpretation is that, for these vulnerable faults, the classification area is smaller, and they are closer to the boundaries, while the robust faults are far away from the boundaries, which makes them more difficult to attack.

Furthermore, some intriguing patterns were discovered:

(1) **Asymmetry:** Intuitively, if fault A is easily attacked as fault B, fault B should be perturbed as fault A as well, which means fault pairs A–B and B–A are equally vulnerable, and the heat maps should be symmetric about the diagonal. The symmetric pattern conforms to intuitive expectations because vulnerable pairs are commonly closer in the input space. However, in fact, only a small number of the fault pairs in the three heat maps are

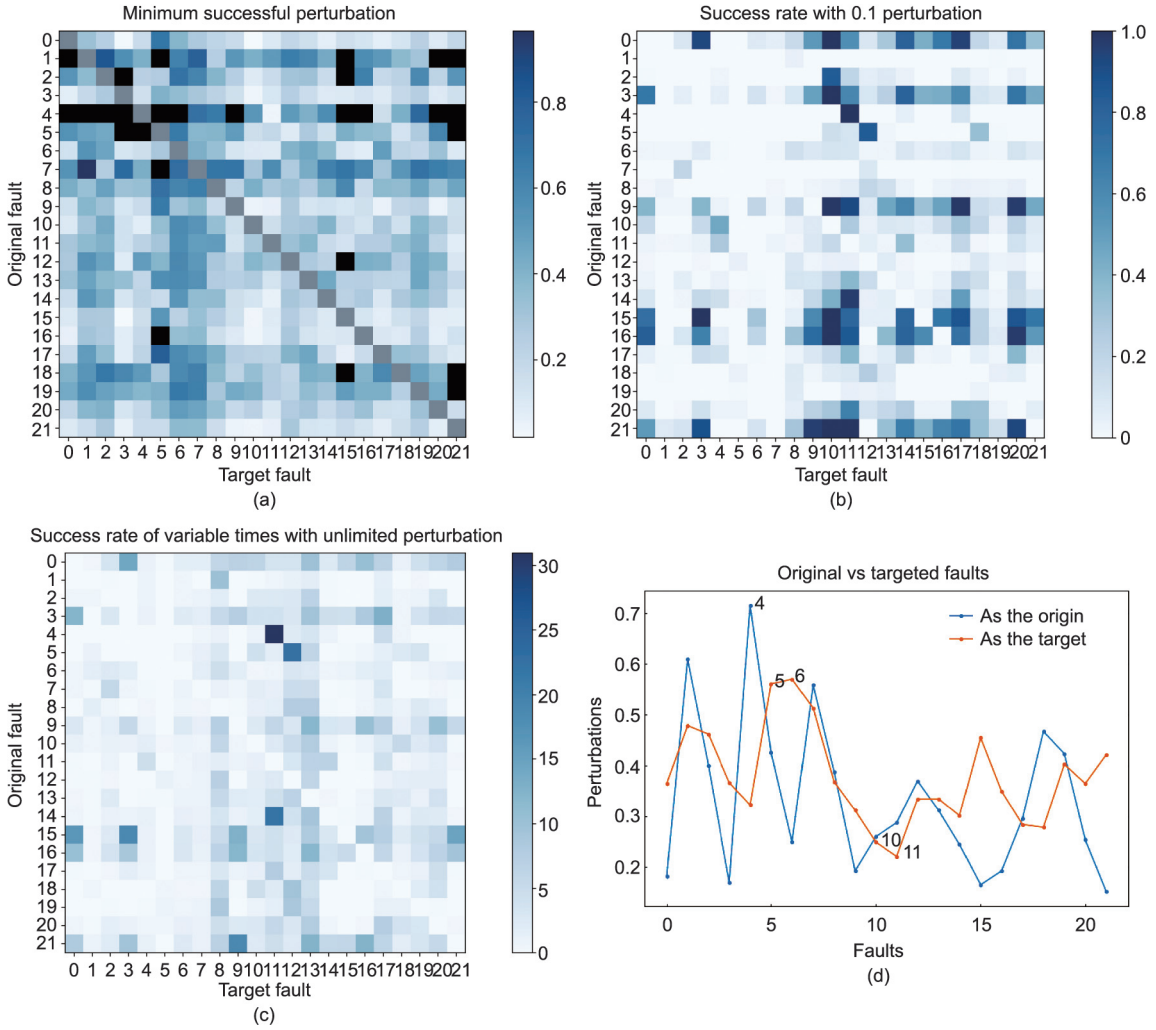

Fig. 10. (a) Average minimum successful perturbation of the fault pairs. It is the mean of the smallest ε between the original–target fault pairs. The points with darker colors indicate the fault pairs are harder to attack. The black points indicate the fault pairs that cannot be attacked even once at maximal perturbation. (b) Success rate with 0.1 perturbation of fault pairs. Success rate is the percentage of the successfully attacked samples between the original–target fault pairs to the total number of original fault samples. The points with lighter color indicate the fault pairs are harder to attack. (c) Success variable rate with unlimited perturbation. A successful perturbed variable counts once. The points with lighter color indicate the fault pairs are harder to attack. (d) Average minimal perturbations of original and targeted fault pairs, from the rows and columns in Fig. 10(a).

symmetrical, and a large number of asymmetrical patterns exist, such as original–target pairs 21–20, 21–17, 5–12, and so on. Such asymmetry is intriguing; it shows that some faults like fault 21 resemble many other faults, such as faults 20 and 17, but not vice versa.

(2) **Concentration:** An examination of the two heat maps shows that, like the original fault, fault 4 is robust overall, and the transition between it and most other faults is difficult. However, according to the previous section, the attack success rate of fault 4 is not very low, so the targets of fault 4 are concentrated on one fault, fault 11. This is more evident when calculating the success variable rate, that is, what percentage of variables in a fault can be attacked to other faults, as shown in Fig. 10(c). The value of fault pair 4–11 is 30.98, which means that each sample of fault 4 has nearly 31 variables (total 50) that can be perturbed to make the DNN classifier predict fault 11. It can be noted that for the original fault 4, the largest target fault 11 accounts for 68.55% of the total.

The most straightforward interpretation of these two patterns is given from a geometrical perspective of classification in the following subsection.

3.3. Geometrical interpretation of fault classification

This section explains, using a geometric approach, why OVA works and why correlations between faults and variables are as shown in the previous two subsections. Based on the geometric approach, a deeper insight into the industrial fault classification system is also provided.

The primary approach in this section is to draw classification boundaries to demonstrate the geometric characteristics of fault classification. Owing to the limited number of visual dimensions, the changes in the classification boundary are only drawn for two variables of a specific type of fault. Because the values of the other variables will affect the shape of the classification boundary, the values of the remaining variables are represented by the mean value of this fault.

First, for variables 0 and 46, the DNN decision boundaries of faults 20 and 21 are drawn, as shown in Fig. 11. Because the values for the other variables are approximated by means, there are some offsets on the positions of the fault point. In fact, the fault samples were correctly classified using the classifier. A comparison of these two results gives the following points:

(1) Why can a perturbation in a single variable successfully attack a fault classification system? From the Fig. 11, the output of the classifier is varied along one variable. In particular, for fault 21 (the left of Fig. 11), both horizontal and vertical perturbed samples can make the classifier incorrectly predict at least three different fault categories.

(2) Why do different faults have different vulnerabilities for the same variable? Compared to fault 21, fault 20 is easier to classify for the classifier, and its output confidence is also higher; hence, the classification area is broader, and the sample is farther from the classification boundary. On the other hand, the confidence of the classifier for fault 21 is lower, the classification area is narrower, and the samples are very close to the classification boundary. This means that only slight perturbations can make the classifier output a wrong prediction for the samples of fault 21, whereas the samples of fault 20 are more robust to perturbations. This is consistent with the heat maps in Section 3.1, and the confusion matrix (Fig. S1).

(3) Why are fault pairs asymmetric? The major reason is that the input space for classification is high dimensional, and the fault samples only occupy a small part of the whole space, whereas the classification area of some faults is vast, occupying a large area in the space. This means that, despite the distance between different faults being extremely far, in some areas of the input space, the classification boundaries of these two types of faults may be adjacent. In the two figures, for the samples of fault 21, they are close to the fault 20 decision area in the projection on these two variables. The classification boundaries of the two types of faults are adjacent in some parts of the space, but not the part of the space that is close to the samples of fault 20. Projecting a *z*-axis onto Fig. 11 reveals that the decision area of fault 20 expands on that *z*-axis and squeezes the areas of faults 21, 3, 18, and 14. The samples of the fault 20 cluster somewhere on the *z*-axis, so it is not adjacent to fault 21 in the projection onto variables 0 and 46. In practice, the *z*-axis represents the other variables.

Next, we consider Fig. 12, which seeks to explain why some fault pairs are highly vulnerable. From the sample of fault pairs 4–11, on the eight variables of the four subfigures, the decision area of fault 4 is closely surrounded by fault 11. Therefore, we can assume that in the higher-dimensional input space, a large part of the fault 4 decision area is also surrounded by fault 11. Therefore, the adversarial samples for fault 4 have a high probability of being mistakenly classified as fault 11.

Finally, the robust and vulnerable variables were studied from a gradient perspective. The sample distribution of fault 5 is drawn against variables 2 and 48, and the gradient of the classification loss as a function of these two variables is calculated at each point, as shown in Fig. 13. The loss value of the DNN is calculated by the cross-entropy between the output and ground truth, which measures the difference between two probability distributions. In terms of the gradient value, variable 48 is an order of magnitude higher than variable 2. In the direction starting from the position of the samples, the change in the gradient along the direction of variable 48 is significantly greater than that of variable 2, and there is almost no gradient along variable 2. This means that applying a perturbation to variable 48 can make the output of the classifier change more drastically, and it is easier to obtain the wrong output of the classification, which also verifies the results in Section 3.1. In addition, by combining the gradient map and the classification boundary, it can be found that the closer the fault is to the classification boundary, the higher the gradient becomes. This means that the samples near the decision boundary are more vulnerable and vice versa, which is consistent with the geometric interpretation in Section 3.2.

4. One-variable defense

To defend the adversaries generated by the OVA and improve the robustness of the DNN, we propose an adversarial training method to add adversarial samples during the training process of the DNN to train a classifier that is more robust to perturbations. Unlike most existing adversarial training methods, which perturb the training samples on every variable to obtain adversarial training samples, our method only adds perturbation to the variables with higher gradients. The adversarial training methods are only suitable for iterative training models, such as DNN, so SVM and *k*NN are not used for our defense method.

To achieve this, a gradient table of the average gradient for the variables of each fault is calculated at each epoch during the training process, which is a graph similar to Fig. S2. Only the fault variables ranked in the top *K* position in the table are selected for perturbation to generate the adversaries, where *K* is an adjustable hyperparameter to control the partition of perturbed variables during adversarial training. The sign function was used to determine the direction of the selected variables. The perturbation was applied along the direction of the ascending gradient.

A more robust classifier can be obtained by adding adversarial samples with the same label as the original samples during model training. Fig. 14 shows the success rate of OVA for the three classifiers, the original DNN, the DNN with FGSM training, and the DNN with the proposed method (OVA-training). In the experiments, *K* was set to 50%, which means that the fault variables with the highest half of the gradient are perturbed in the proposed method.

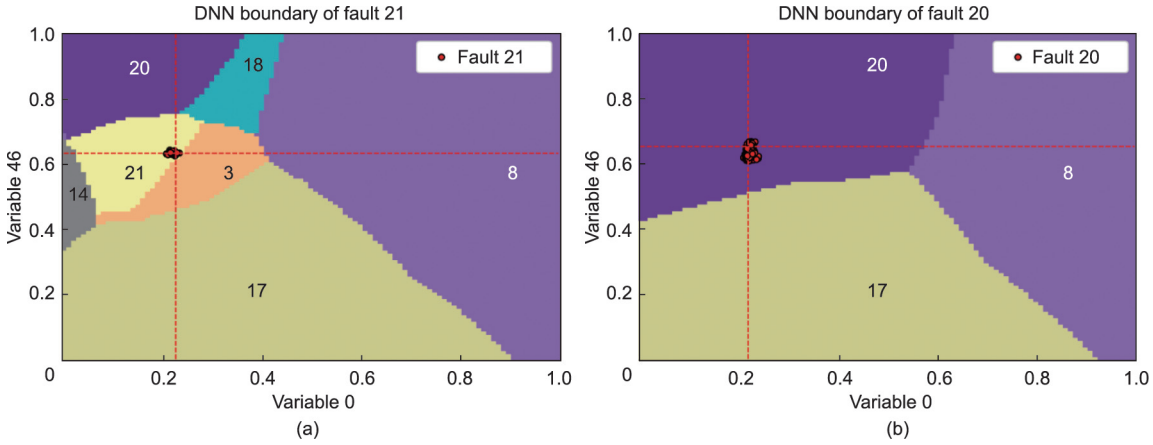

Fig. 11. DNN classification boundaries for faults (a) 21 and (b) 20 as a function of variables 0 and 46. Red points are samples of a certain fault type. Different colors represent the classification area of different faults, where the number shows its fault type.

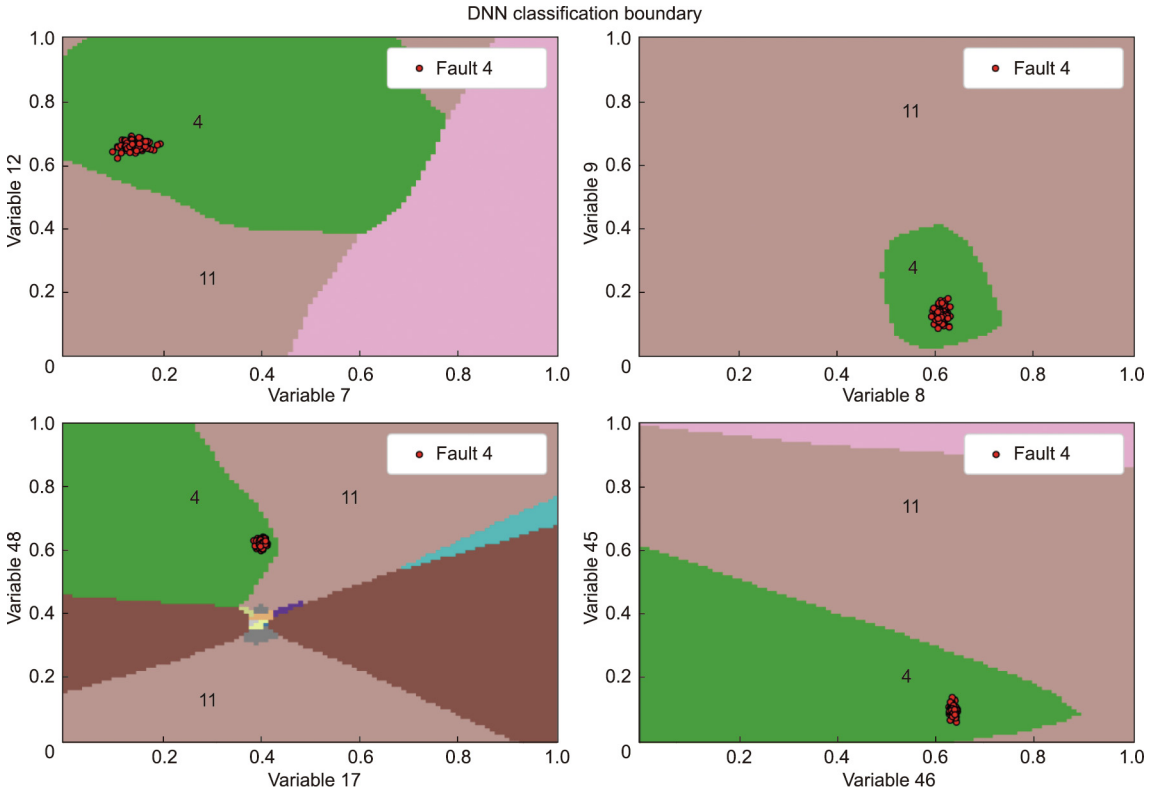

Fig. 12. DNN classification boundaries of fault 4 as a function of eight variables.

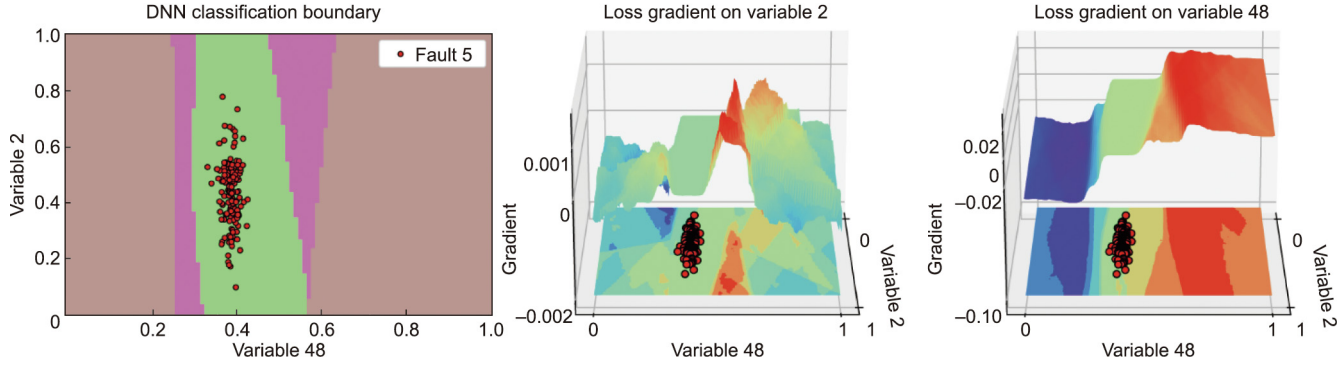

Fig. 13. DNN classification boundaries for fault 5 as a function of variables 2 and 48, and the corresponding gradient. The gradient is from a classifier's cross-entropy loss of fault 5 to variables 2 and 48, respectively, and three-dimensional (3D) and 2D projections of the gradient contour are plotted.

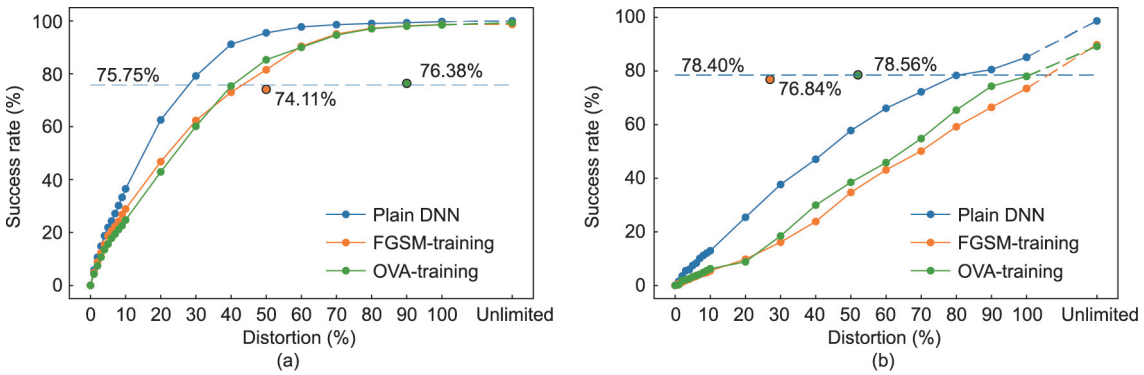

Fig. 14. Success rate and test accuracy for different training methods: (a) TEP and (b) SP.

The results on the two datasets show that the proposed method can effectively reduce the success rate of OVA. However, compared with FGSM, the OVA-training is not as robust when distortion increases, but we gain high accuracy on the test set, 0.63% (resp. 0.16% in the SP) higher than that of the plain DNN model, and 2.27% (resp. 1.76%) higher than that of the DNN with FGSM adversarial training. This is mainly because of the global variable selection across the different faults. Adding perturbation to each variable reduces the attack success rate, but as a trade-off, the test accuracy decreases due to some perturbations in the high-confidence variables. Our method only adds the perturbations on the variables that are difficult to classify, so as to help the model learn more explicit boundaries for those variables, which in turn improves the accuracy of the test set. Meanwhile, because the attack method is on only one variable, reducing the perturbed variables during adversarial training does not decrease robustness.

5. Conclusions

This study examined the security of industrial fault classification systems. An OVA was proposed to attack the fault classification models by perturbing only a single variable. The results showed that perturbing only one variable was sufficient to attack industrial fault classifiers. The attack success rate was high even when the perturbation was limited to a small value. In the TEP, 10% (resp. 20%) distortion on a single variable perturbed 36.5% (resp. 62.6%) of the samples to successfully attack the classification systems of DNNs.

Exploiting the OVA method, this study also explored the geometry of an industrial fault classification model, represented by DNNs. The classification boundaries and gradients were plotted to provide insight into the vulnerability and robustness of industrial fault classification systems. Finally, to minimize the impact of adversarial attacks, an adversarial training method using some of the variables was proposed. This resulted in a trade-off between a small decrease in robustness under large perturbations to give a much higher prediction accuracy (0.63% higher than the DNN without adversarial training and 2.27% higher than the DNN with all variable adversarial training).

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) (92167106, 62103362, and 61833014) and the Natural Science Foundation of Zhejiang Province (LR18F030001).

Compliance with ethics guidelines

Yue Zhuo, Yuri A.W. Shardt, and Zhiqiang Ge declare that they have no conflict of interest or financial conflicts to disclose.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.eng.2021.07.033.

References

[1] Ge Z. Semi-supervised data modeling and analytics in the process industry: current research status and challenges. IFAC J Syst Control 2021;16:100150.
[2] Ge Z, Song Z, Ding SX, Huang B. Data mining and analytics in the process industry: the role of machine learning. IEEE Access 2017;5:20590–616.
[3] Dash PK, Samantaray SR, Panda G. Fault classification and section identification of an advanced series-compensated transmission line using support vector machine. IEEE Trans Power Deliv 2007;22(1):67–73.
[4] Chen X, Ge Z. Switching LDS-based approach for process fault detection and classification. Chemom Intell Lab Syst 2015;146(C):169–78.
[5] Wang Y, Wu D, Yuan X. LDA-based deep transfer learning for fault diagnosis in industrial chemical processes. Comput Chem Eng 2020;140:106964.
[6] Chen G, Ge Z. SVM-tree and SVM-forest algorithms for imbalanced fault classification in industrial processes. IFAC J Syst Control 2019;8:100052.
[7] Zhao D, Wang T, Chu F. Deep convolutional neural network based planet bearing fault classification. Comput Ind 2019;107:59–66.
[8] Chadha GS, Panambilly A, Schwung A, Ding SX. Bidirectional deep recurrent neural networks for process fault classification. ISA Trans 2020;106:330–42.
[9] Jiang L, Ge Z, Song Z. Semi-supervised fault classification based on dynamic sparse stacked auto-encoders model. Chemom Intell Lab Syst 2017;168:72–83.
[10] Ren K, Zheng T, Qin Z, Liu X. Adversarial attacks and defenses in deep learning. Engineering 2020;6(3):346–60.
[11] Akhtar N, Mian A. Threat of adversarial attacks on deep learning in computer vision: a survey. IEEE Access 2018;6:14410–30.
[12] Xu H, Ma Y, Liu H, Deb D, Liu H, Tang J, et al. Adversarial attacks and defenses in images, graphs and text: a review. Int J Autom Comput 2020;17(2):151–78.
[13] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, et al. Intriguing properties of neural networks. In: Proceedings of the 2nd International Conference on Learning Representations; 2014 Apr 14–16; Banff, AB, Canada; 2014.
[14] Goodfellow I, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. In: Proceedings of the 3rd International Conference on Learning Representations; 2015 May 7–9; San Diego, CA, USA; 2015.
[15] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: Proceedings of the 6th International Conference on Learning Representations; 2018 Apr 30–May 3; Vancouver, BC, Canada; 2018.
[16] Shafahi A, Najibi M, Ghiasi MA, Xu Z, Dickerson J, Studer C, et al. Adversarial training for free! In: Proceedings of Advances in Neural Information Processing Systems 32; 2019 Dec 8–14; Vancouver, BC, Canada; 2019.
[17] Zhang D, Zhang T, Lu Y, Zhu Z, Dong B. You only propagate once: accelerating adversarial training via maximal principle. In: Proceedings of Advances in Neural Information Processing Systems 32; 2019 Dec 8–14; Vancouver, BC, Canada; 2019.
[18] Su J, Vargas DV, Sakurai K. One pixel attack for fooling deep neural networks. IEEE Trans Evol Comput 2019;23(5):828–41.
[19] Papernot N, McDaniel PD, Jha S, Fredrikson M, Celik ZB, Swami A. The limitations of deep learning in adversarial settings. In: Proceedings of the 1st IEEE European Symposium on Security and Privacy; 2016 Mar 21–24; Saarbrücken, Germany; 2016.
[20] Barreno M, Nelson B, Joseph AD, Tygar JD. The security of machine learning. Mach Learn 2010;81(2):121–48.
[21] Biggio B, Corona I, Maiorca D, Nelson B, Srndic N, Laskov P, et al. Evasion attacks against machine learning at test time. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases; 2013 Sep 23–27; Prague, Czech Republic. Heidelberg: Springer; 2013. p. 387–402.
[22] Hu W, Tan Y. Generating adversarial malware examples for black-box attacks based on GAN. 2017. arXiv:1702.05983.
[23] Sankaranarayanan S, Jain A, Chellappa R, Lim SN. Regularizing deep networks using efficient layerwise adversarial training. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence; 2018 Feb 2–7; New Orleans, LA, USA; 2018.
[24] Gu S, Rigazio L. Towards deep neural network architectures robust to adversarial examples. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015); 2015 May 7–9; San Diego, CA, USA; 2015.
[25] Papernot N, Mcdaniel P, Goodfellow I, Jha S, Celik ZB, Swami A. Practical black-box attacks against machine learning. In: Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security; 2017 Apr 2–6; Abu Dhabi, United Arab Emirates. New York City: Association for Computing Machinery; 2017. p. 506–519.
[26] Akhtar N, Liu J, Mian A. Defense against universal adversarial perturbations. In: Proceedings of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018 Jun 18–23; Salt Lake City, UT, USA; 2018.
[27] Shang C, You F. Data analytics and machine learning for smart process manufacturing: recent advances and perspectives in the big data era. Engineering 2019;5(6):1010–6.
[28] Chen Y. Integrated and intelligent manufacturing: perspectives and enablers. Engineering 2017;3(5):588–95.
[29] Yi TH, Huang HB, Li HN. Development of sensor validation methodologies for structural health monitoring: a comprehensive review. Measurement 2017;109:200–14.
[30] Downs JJ, Vogel EF. A plant-wide industrial process control problem. Comput Chem Eng 1993;17(3):245–55.
[31] Research center of sciences of communication [Internet]. Rome: Semeion Communication Science Research Centre; 2022 Apr 19 [cited 2022 Apr 30]. Available from: https://www.semeion.it.