A Two-layer Mechanism Identification Method for VIN Adversarial Examples

Yingdi Wang, Jiqiang Liu, Tong Chen, Yingxiao Xiang, Endong Tong\textsuperscript{a} and Wenjia Niu\textsuperscript{b}

Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China

E-mail: edtong@bjtu.edu.cn\textsuperscript{a}; niuwj@bjtu.edu.cn\textsuperscript{b}

Abstract. With the rapid development of machine learning algorithms, the security problems are gradually emerging. Most existing machine learning algorithms may be attacked by adversarial examples. Especially in the domain of path planning, the adversarial maps may result in multiple harmful consequences on the paths predicted by Deep Reinforcement Learning (DRL) algorithms. However, there is no suitable approach to automatically identify them. To our knowledge, all previous work used manual observation method to identify the adversarial maps, which is time-consuming. Therefore, this paper explores a two-layer mechanism method to automatically identify the adversarial examples in Value Iteration Networks (VIN). We define the four categories of attack results and identify them by combining path feature comparison and path image classification. Experiments show that this method can achieve an effective identification on VIN adversarial examples.

1. Introduction

Due to the advances in machine learning and deep learning algorithms, AI has developed rapidly and helped humans solve many complicated problems, such as image classification, face recognition, robot path planning and so on[1–3]. However, the vulnerability of the machine learning algorithms is gradually emerging. A well-designed small perturbation at the input layer will result in a totally wrong classification, which is so-called adversarial examples[4]. Adversarial examples have already brought forth potential security threats to many AI applications, such as image classification [4–8], speech recognition [9] and Atari games [10].

However, few studies have focused on adversarial examples in the domain of robot path planning. With the rise of deep learning models, DRL algorithms have achieved a state-of-the-art performance in the domain of robot path planning, such as VIN, DQN and A3C [3, 11, 12]. Some prior work has found there are adversarial examples existing in some DRL algorithms [13–15]. By reviewing these literatures, we can find some commonalities. For the grid-world path planning task, the map is a synthetic grid-world domain with a particular obstacle configuration. An adversarial example in path planning consists of obstacle-level perturbation added to the target map (see Figure 1), so that a human observer would recognize it as a normal map, but a machine learning algorithm will plan a very different path for it.

In this paper, we specially focus on the adversarial examples of VIN path planning, which can achieve a better generalization compared with other DRL algorithms [12]. In the paper [15], they proposed a method that can generate adversarial maps to attack VIN path planning by adding
obstacles to the grid-world domain. However, there are some deficiencies in their identification method on VIN adversarial examples. They evaluated the attack results through a manual observation method, which is time-consuming and ineffective. Especially when generating a large amount of adversarial maps, it is not possible to quickly get feedback about the effect of attacks through manual observation. Therefore, it is necessary to explore a method to identify the VIN adversarial examples automatically.

Aiming at the existing problem of the manual observation method, this paper explores a fast approach to automatically identify VIN adversarial examples. We analyze the possible scenarios of the adversarial maps and define the categories of the predicted path pairs. In order to realize a large-scale detection, we extract the two paths into one path image, and transform the path comparison to path image classification according to the defined categories. But the classifier may be confused by the various trajectories, so we also retain those easily distinguishable path features to help the classifier to achieve a high accuracy. In this way, we implement two-layer mechanism identification method for VIN adversarial examples by combining the path features and path images.

The rest of the paper is structured as follows: In Section 2, we discuss the related works in this field. This is followed in Section 3 by details about the category definition of adversarial maps. Section 4 describes the proposed method to identify the VIN adversarial examples automatically. Section 5 presents our experiments with the adversarial map dataset. Finally, Section 6 concludes the work of this paper.

2. Related Work
In this section, we investigate the existing research work in the field of adversarial examples. Most of the prior work paid attention to the adversarial examples in the field of image classification, which are generated by adding fine-grained per-pixel modifications to images and will lead to a misclassification [4–7]. In the field of speech recognition, attackers use synthesized obfuscated speech to cause a wrong voice command to be executed by the device [9]. In Atari games, adversarial examples can make the Atari game misfunction by adding noise to the background of the game [10]. By reviewing these literatures, we can find that the adversarial examples cause different attack results in different application fields. For some applications, the attack results can be clearly identified as either success or failure. However, in the field of robot path planning, the adversarial maps cause various impact on the path. Therefore, we need to explore a method to compare the difference between the two paths to automatically identify the attack results.

3. Preliminaries
We refer to an original map and an adversarial map as a pair of maps. In this way, the VIN-predicted trajectories for a pair of maps can be recognized as a pair of paths, the original path
and the adversarial path.

3.1. Definition of Path Pairs

The pair of paths can be described in a Cartesian coordinate system, as shown in Figure 2. The path pairs can be expressed as follows:

\[ P_{\text{original}} = (r_1, r_2, r_3, \ldots, r_n), \quad P_{\text{adversarial}} = (v_1, v_2, v_3, \ldots, v_m) \]  

where \( P_{\text{original}} \) represents the VIN-predicted path of original map, and \( r_i \) is the \( i \)th step of the path, which can be expressed as a point \((x_i, y_i)\). And \( P_{\text{adversarial}} \) is expressed in the same way.

3.2. Category Definition of VIN Adversarial Examples

By visualizing a pair of paths on a path image, we transform the attack results into path images. The attack results can be divided into four limited categories as shown in Figure 3. Based on the difference between \( P_{\text{original}} \) and \( P_{\text{adversarial}} \), we define the four categories of VIN
adversarial maps, namely UrP, FP, DP, UcP. It should be additionally noted that the critical value 4 is not an constant, but is determined by the size of maps.

The Unreached Path (UrP) $P_{\text{adversarial}}$ does not reach to the goal position.

The Fork Path (FP) There is a significant difference between $P_{\text{original}}$ and $P_{\text{adversarial}}$. Where $P_{\text{original}}$ and $P_{\text{adversarial}}$ do not coincide, the maximum vertical distance and the maximum horizontal distance between them are greater than 4.

The Detour Path (DP) There is a slight difference between $P_{\text{original}}$ and $P_{\text{adversarial}}$. Where $P_{\text{original}}$ and $P_{\text{adversarial}}$ do not coincide, the maximum vertical distance and the maximum horizontal distance between them are less than or equal to 4.

The Unchanged Path (UcP) $P_{\text{adversarial}}$ is exactly identical with $P_{\text{original}}$.

This paper supposes that an adversarial map which can make a significant change in the original path is a successful attack. Therefore, the adversarial maps that cause the UrP and FP results are the VIN adversarial examples that attack successfully. Correspondingly, the adversarial maps that cause the DP and UcP results fail to attack the VIN path planning.

4. Two-layer Mechanism Identification for VIN Adversarial Examples

In this section, we will introduce the two-layer mechanism identification method in detail.

4.1. Framework

Figure 4. The framework of VIN adversarial example identification approach.

The framework of the method is mainly divided into three modules, as shown in Figure 4.

VIN Path Planning Module A generated map pair consists of an original map and an adversarial map. After VIN path planning module, there will be a path pair corresponding to a map pair predicted by VIN. By studying the path pairs, we can determine whether the adversarial maps have attacked VIN successfully.
PathPairs Classification Module} Classifying the path pairs is the most important module of our method. It is mainly divided into two parts, the comparison of path features and the classification of path images, which will be introduced detailedly in section 4.2.

Identification Module In identification module, the label of the path pairs are matched with the corresponding map pairs to identify the VIN adversarial examples. Therefore, the map pairs are finally identified as Success-FP, Success-UrP, Unsuccess-DP and Unsuccess-UcP.

4.2. Two-layer Mechanism Classification for Path Pairs
As described in Section 3, it is easy to find that UrP and UcP can be well distinguished by path features. Whereas, DP and FP can be well recognized by path images, for the path trajectories in path images of DP and FP are very different. Therefore we combine the path features and the path images to do the classification for path pairs.

Path Feature Comparison Firstly, we compare whether the stop points in the two paths are equal. If they are not equal, which means the adversarial path fails to reach the goal, the path pair belongs to UrP category. Then we compare whether all the points in the two paths are completely equal. If they are equal, which means the path has not been changed, the path pair belongs to UcP category. In this way, through comparing the obvious features of the paths, the UrP and UcP categories can be quickly distinguished.

Path Image Classification At first, each pair of paths is visualized in one path image after path image extraction process, as shown in Figure 3. Each path pair is visualized with different colours on a blank map to prevent obstacles affecting the classification. Then, we need to train a path image classifier with the path image dataset. Unlike usual images, the pixel information of the path image is a sparse matrix and the training set is not large enough. Therefore, we chose Support Vector Machine (SVM) to do path image classification [16]. We built a labeled path image dataset to train SVM to classify the remaining path images of DP and FP categories.

4.3. VIN Adversarial Example Identification Method

Algorithm 1 VIN Adversarial Examples Identification Method

```
Input: D_{map} = \{MapPair_1, MapPair_2, ..., MapPair_i, ..., MapPair_n\}
Output: D_{adversarial} = \{(MapPair, PathPair, Label)_1, ..., (MapPair, PathPair, Label)_m\}

1: D_{map} \rightarrow D_{path} = \{(MapPair, PathPair, Label)\}_1^n
2: for each PathPair_i in D_{label} do
3:     if P_{original}(r_n) \neq P_{adversarial}(v_m) then
4:         Label_i = UrP
5:     else if P_{original} = P_{adversarial} then
6:         Label_i = UcP
7:     end if
8: end for
9: PathPair_i \rightarrow D_{path_image} D_{label_image}
10: for each Label_i in D_{label} do
11:     if Label_i = UrP OR FP then
12:         D_{adversarial} = D_{adversarial} + (MapPair_i, PathPair_i, Label_i)
13: end if
14: end for
15: return D_{adversarial}
```
Algorithm 1 describes the pseudo code for the VIN adversarial example identification method. In the algorithm, the input is the original map pairs $D_{map}$, and the output is the adversarial examples with label information $D_{adversarial}$.

5. Experiment and Evaluation
In our experiments, we generated map pairs using the method proposed in [15] as the experimental dataset. We evaluated the feasibility and dependability of the method.

5.1. Experiment Setup
We prepared the experiment from two aspects, data preparation of mappairs and pathpairs, and dataset building of path images.

5.1.1. Data Preparation
We generated 5,000 pairs of maps using the source code provided by [12]. The original map is a grid-world domain of size $28 \times 28$, where the obstacle configurations and start and goal positions are all random. The adversarial map is generated by adding one additional obstacle. Then we used VIN algorithm to obtain pathpairs (see Figure 5).

![Figure 5. The examples of map pairs.](image)

5.1.2. Dataset Building
In order to train the path image classifier, we need to build the path image dataset of DP and FP. The generated 5,000 path pairs are all transformed into path images through path image extraction. We labeled all the path images and picked out the DP and FP categories to build the path image dataset, as shown in Figure 6.

![The FP class](image)

![The DP class](image)

![Figure 6. The path image dataset of FP and DP categories.](image)
5.2. Evaluation and Analysis
We evaluated the dependability and feasibility of the method mainly from two aspects, classifier accuracy and processing time. The distribution of the 5,000 path pairs dataset is shown in Table 1. It can be seen that most of them are of UcP category and fail to attack VIN, and the data of FP category is the smallest.

| Categories       | Unreached Path | Fork Path | Detour Path | Unchanged Path |
|------------------|----------------|-----------|-------------|----------------|
| Quantity         | 1,632          | 204       | 1,076       | 2,088          |

5.2.1. Dependability Evaluation In this paper, the dependability of the method has a close relationship with the accuracy performance of the path image classifier. Therefore, we evaluated the dependability of our method by analyzing the classification performance of SVM. In order to avoid the impact of data imbalance on the classifiers, we randomly generated images for FP category to the same number as DP category. Then we randomly split the path image dataset into the training set and the testing set with a ratio of 7:3. We used the balanced training set to train the classifier, and used the testing set in real situation to test the classifiers. We evaluated the dependability of SVM classifier by ROC and PR curves. As we can see from Figure 7, the AUC is 0.976, which illustrates SVM can achieve a high accuracy for a small size training set, and the AP is close to 1, which illustrates that the test accuracy of SVM classifier is in high dependability. Therefore, it is confirmed that the SVM classifier can achieve a reliable classification for path images.

![PR curve](a) PR curve  ![ROC curve](b) ROC curve

Figure 7. The PR and ROC curve of SVM.

5.2.2. Feasibility Evaluation We evaluated the feasibility of the identification method by comparing the two-layer mechanism method and the manual observation method to process the same number of path pairs. We selected 100 path pairs from the dataset, including 40 path pairs from FP and DP categories, and 60 path pairs randomly selected from UrP and UcP categories. The processing-time comparison of the two methods are shown in Table 2. Obviously, the processing speed of the two-layer mechanism method has a significant improvement, which confirmed that the method proposed in this paper can realize a fast and feasible identification of VIN adversarial examples.
Table 2. The processing-time comparison of two methods.

| Methods                  | Processing Time (100 path pairs) |
|--------------------------|----------------------------------|
| Two-layer Mechanism Method | 0.22s                            |
| Manual Observation Method | 300s (approximately)             |

In summary, the two-layer mechanism identification approach can achieve an automatical identification for VIN adversarial examples. The combination of path feature comparison and path image classification can ensure the high-accuracy of the classification and the high-speed of the identification approach.

6. Conclusion

This paper mainly focuses on identifying the VIN adversarial examples. We define the four categories of adversarial maps by analysing the possible impacts that adversarial maps may cause on VIN path planning. Based on the categories definition, we implement a two-layer mechanism identification approach by combining path feature comparison and path image classification. The experiments prove that the method can effectively identify VIN adversarial examples automatically. In future work, we will further study whether our identification approach is applicable to the adversarial examples of other DRL algorithms in the domain of path planning.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (61972025, 61802389, 61672092, U1811264, 61966009), the Fundamental Research Funds for the Central Universities of China(2018JBZ103, No.2019RC008), Science and Technology on Information Assurance Laboratory, Guangxi Key Laboratory of Trusted Software(KX201902).

References

[1] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M et al. 2015 International Journal of Computer Vision 115 211–252
[2] Schroff F, Kalenichenko D and Philbin J 2015 Proceedings of the IEEE conference on computer vision and pattern recognition pp 815–823
[3] Zhu Y, Mottaghi R, Kolve E, Lin J J, Gupta A, Fei-Fei L and Farhadi A 2017 Robotics and Automation (ICRA), 2017 IEEE International Conference on (IEEE) pp 3357–3364
[4] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I and Fergus R 2013 arXiv preprint arXiv:1312.6199
[5] Goodfellow I J, Shlens J and Szegedy C arXiv preprint arXiv:1412.6572
[6] Gu S and Rigazio L 2014 arXiv preprint arXiv:1412.5068
[7] Kurakin A, Goodfellow I and Bengio S 2016 arXiv preprint arXiv:1607.02533
[8] Sharif M, Bhagavatula S, Bauer L and Reiter M K 2016 Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (ACM) pp 1528–1540
[9] Carlini N, Mishra P, Vaidya T, Zhang Y, Sherr M, Shields C, Wagner D and Zhou W 2016 USENIX Security Symposium pp 513–530
[10] Huang S, Papernot N, Goodfellow I, Duan Y and Abbeel P 2017 arXiv preprint arXiv:1702.02284
[11] Panov A I, Yakovlev K S and Suvorov R 2018 Procedia Computer Science 123 347–353
[12] Tamar A, Wu Y, Thomas G, Levine S and Abbeel P 2016 Advances in Neural Information Processing Systems pp 2154–2162
[13] Bai X, Niu W, Liu J, Gao X, Xiang Y and Liu J 2018 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC) (IEEE)
[14] Chen T, Niu W, Xiang Y, Bai X, Liu J, Han Z and Li G 2018 arXiv preprint arXiv:1807.06752
[15] Liu J, Niu W, Liu J, Zhao J, Chen T, Yang Y, Xiang Y and Han L 2017 International Conference on Information and Communications Security (Springer) pp 374–384
[16] Liu P, Choo K K R, Wang L and Huang F 2017 Soft Computing 21 7053–7065