Data Poisoning Attack against Anomaly Detectors in Digital Twin-Based Networks

Shaofeng Li*, Wen Wu*,✉, Yan Meng†, Jiachun Li†, Haojin Zhu†, and Xuemin (Sherman) Shen‡
*Frontier Research Center, Peng Cheng Laboratory, Shenzhen, China
†Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
‡Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Canada
Email: {lishf, wuw02}@pcl.ac.cn, {yan_meng, jiachunli, zhu-hj}@sjtu.edu.cn, sshen@uwaterloo.ca

***Abstract*—In this paper, we study the abnormal behaviors detection and the corresponding data poisoning attacks in digital twin (DT)-based networks. We first analyze the abnormal behaviors existing in the DT-based networks, including environment anomalies, hardware and software faults, and network attacks. Specially, we design a machine learning (ML)-based anomaly detector to identify network attacks. Furthermore, due to the strong dependency of ML models on training data, in which the outputs of the trained ML models can be affected by the poisoned samples. We design a data poisoning attack scheme against the proposed ML-based anomaly detector, in which attackers can effectively compromise the output of anomaly detectors. Extensive experimental results adopting three commonly used ML-based models demonstrate that the attack can compromise these detectors with over 80% probability.**

I. Introduction

With the advancement of data analysis and communication techniques, digital twin (DT) has been applied in a wide range of fields, including cyber-physical systems [1], smart city [2], and network systems [3], [4]. DT is a virtual replica of the physical system, which can enable Internet of Everything (IoE) applications. According to the recent report *Globe News Wire* published by *Allied Market Research*, the market revenue of DT is predicted to be $125.7 billion by 2030 [5]. Moreover, DT is expected to be one of key technologies in the future 6G network [6], [7].

Although employing DT provides various conveniences and benefits to the user, DT suffers from abnormal behaviors (e.g., network attacks). The root reason is that since DT is a replica of the real-world network, the abnormal behaviors in these systems could be mapped in the digital space via DT's abundant interfaces (e.g., wireless communication channels). There are several works exploring DT-based anomaly detection [1], [8], [9]. Chhetri *et al*. [8] build DTs by utilizing the side channel information of the physical system that are unintentionally revealed and then perform anomaly detection upon them. Lu *et al*. [1] propose a DT-enabled anomaly detection for asset monitoring by cross-referencing with operational conditions from buildings' digital twins. Gao *et al*. [9] consider anomalous behaviors caused by modelling errors and then integrate both the DT and data-driven techniques to detect these types of faults in physical systems. However, these mechanisms pay less attention to the traffic information in DT, thus implementing traffic analysis-based anomaly detection in DT-based networks is still an open problem.

In this paper, we first analyze the abnormal behaviors existing in the DT-based networks, including environment anomalies, hardware and software faults, and network attacks. Specially, we propose a machine learning (ML)-based anomaly detector to identify network attacks. For further study, *is there any threat faced by those ML-based detectors when deploying them in DT scenarios?* Our study reveals that *directly applying traditional ML-based traffic analysis models is vulnerable to data poisoning and thus insecure.* Obtaining this answer is not straightforward due to the following challenges. *(1)* For DT-based networks, how to construct and characterize anomaly detection mechanisms remains an open problem. *(2)* How to deploy and launch the data poisoning attack in a DT-based network, which is not yet studied, is challenging. *(3)* Considering there is no public-available anomaly traffic flow in the DT-based network, how to evaluate and demonstrate the effectiveness of the proposed data poisoning attack?

We address the above-mentioned challenges via the following steps. *Firstly*, we define the abnormal behaviors existing in DT-based networks, including physical environment anomaly, network errors caused by hardware and software faults, network faults caused by device misconfiguration, synchronization faults, and network attacks. Focusing on those network attacks, we design corresponding ML-based anomaly detectors in DT-based networks. *Secondly*, based on the observation that ML-based detectors are fragile when there are poisoning samples existing in their training data, we design our novel data poisoning attacks which are suitable in DT-based networks. *Lastly*, we refer to *CICIDS-2017* traffic flow dataset to parse traffic flows that conform to the DT requirements. The reconstructed dataset is utilized to evaluate the performance of the data poisoning attack. Evaluation results show that the adopted data poisoning attacks against ML-based anomaly detectors can conduct the attack behaviors bypassing three mainstream ML-based anomaly detectors.

Generally, DT-based networks are data and model jointly-driven systems [10], [11]. Any data security issues can lead to severe physical damages, especially when DT is used for mission-critical industrial applications. The main contributions in this paper are summarized as follows:

- We analyze the anomaly behaviors that potentially exist in

✉Wen Wu (wuw02@pcl.ac.cn) is the corresponding author of this paper.

978-1-5386-7462-8/23/$31.00 ©2023 IEEE

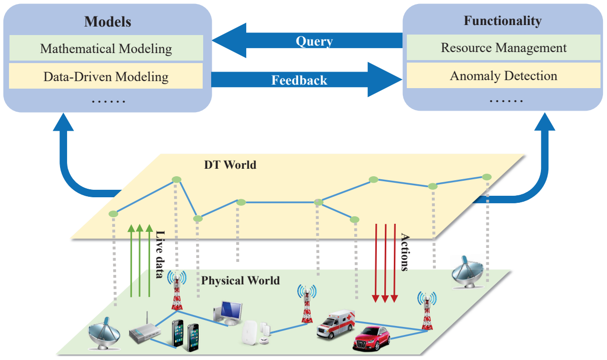

Fig. 1. DT for wireless networks.

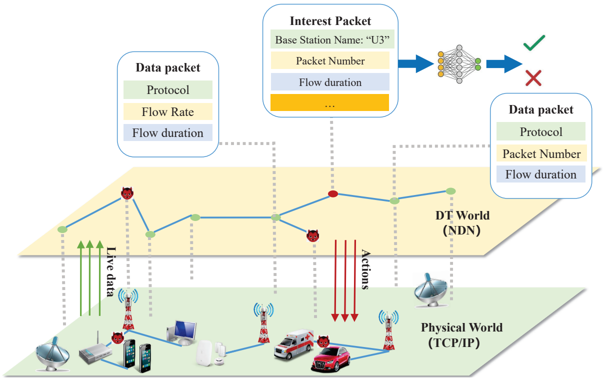

Fig. 2. Architecture of anomaly detection in DT-based network.

DT-based networks. Furthermore, we model and design anomaly detection schemes based on traffic analysis in DT scenarios and reveal several principles that DT-based anomaly detectors should meet.

- We reveal the vulnerability of ML-based anomaly detectors in a DT-based network. Specifically, we show how data poisoning can happen in DT-based networks.

We carry out extensive experiments on anomaly detection in DT-based networks, and the results show that proposed attacks can achieve more than 93% of attack success rate on average.

The remainder of this paper is organized as follows. The necessary background including the digital twin and the construction process of the DT-based network are introduced in Section II. Section III shows the abnormal behaviors and how to build traffic analysis-based anomaly detection in DT scenarios. The data poisoning attacks are proposed in Section IV. Experimental results are provided in Section V, followed by the conclusion in Section VI.

II. System Model

A. DT-Based Network

The DT is defined as a digital replica of a living or non-living physical entity. With the development of emerging high-speed communication technologies that enable fast data synchronization among sensors and actuators, the construction of a DT (i.e., a synchronized digital replica of the physical asset) has matured gradually during the past decade. In this paper, we illustrate a DT architecture for a network system in Fig. 1.

In the architecture shown in Fig. 1, the physical entity (e.g., base station, router, or mobile device) continuously sends data (its status or sensed environmental information) to its DT. A DT can also send control signals to make its physical entity take the specific action by a downlink. The existing of a downlink which can control the physical world from the digital world is the main difference between DT technology and the metaverse. Besides, the simulation in the DT space can reduce the overheads of network testing that are often costly and time-consuming, especially, in some cases that can not be emulated in the physical world.

For simplicity, our DT architecture (as shown in Fig. 1) only consists of two components, including model and functionality. The model profiles the physical world as accurately as it can by mathematical and/or data-driven modelling. The constructed models are improved incrementally based on uploaded live data, thereby profiling the underlying elements of the physical network in a real-time and accurate way. For the other component of the DT-based network, the functionality can provide capabilities, such as resource management, event prediction, and anomaly detection. In this work, we mainly focus on anomaly detection that identifies abnormal behaviors that appear in DT-based networks.

B. DT-Based Network Construction

In the DT-based network, millions of devices with various types of traffic are synchronized continuously to their DTs. Considering a device in the physical world (defined as P_x) tries to send traffic to its receiver (P_y). We build two virtual replicas of P_x and P_y on the edge cloud (may locate on different edges), presented as DT_x and DT_y. The traffic of P_x in the physical world are extracted as content-centric data packets ($(d_0, d_1, ..., d_m)$) by a network flow parser (DT_{fmap}). In our design, devices perform network communication via TCP/IP architecture in physical space. Meanwhile, their DTs exchange information by a Named Data Networking (NDN) [12] architecture in digital space. The overview architecture of DT-based anomaly detection is demonstrated in Fig. 2.

Synchronization: For a physical device which is operating a TCP/IP connection with the other device, we deploy a parser on its digital replica in the edge cloud to monitor its network flow status. The parser (DT_{fmap}) extracts some of the interesting contents for specific functionalities of the DT-based network (e.g., anomaly detection in this work). More specifically, for anomaly detection functionality, the interesting contents can be connection protocols, packet length, flow duration, total forward packets, total backward packets, *etc.* By this parser, the physical device can synchronize its network traffic flow status to its DT. As the connection continues, the flow duration and packet number attributions of its digital replica will be updated synchronously. For anomaly detection functionality of the DT-based network, it only needs to query specific interested named data packets to implement its goals. In the next subsection, we will introduce how the anomaly detection functionality of the DT-based network can be implemented on the named data networking.

Inter-Twins Communication via Named Data Networking (NDN): Named Data Networking (NDN) [12] delivers naming data that uses application-layer names from the producers and consumers. In NDN network architecture, consumers send a *Interest* packet to request the desired data, in which the interest packet contains the name of the requested data. Producers produce NDN *Data* packet. Each NDN router forwards Interest packets according to their names, and records which interface this packet is received and which interface this packet is forwarded to the next hop. Once an Interest packet reaches a Data packet, the Data packet will be returned according to the reverse path that the Interest packet arrives.

When performing the anomaly detection functionality of the DT-based network, the anomaly detection functionality of DT-based network requires a set of Interest packets that are named by the specific application name, e.g., some attributes ("flow duration", "total packets", *etc.*) of the base station or vehicle. With those abstractions of the physical network connections, it is easy for DT systems to employ off-the-shelf ML algorithms to implement the anomaly detection functionality.

III. ML-Based Anomaly Detection

In this section, we give a formal definition of abnormal behaviors in DT-based networks and design anomaly mechanisms based on the traffic in DT.

A. Abnormal Behaviors

In this subsection, we introduce abnormal network behaviors, especially network attack behaviors that happened in the physical world first and their corresponding disturbances in the DT network. In the DT-based network, traffic and signal generation functions can be mirrored in a near real-world manner. It is reliable to assume the abnormal behaviors in the physical network would be mapped in the emulated replicas synchronously.

Abnormal network behaviors are unusual and significant changes in the traffic of a network. The changes may present in link traffic volume, packet length, flow duration, *etc.*. The causes of anomalies include both legitimate and illegitimate activities. Legitimate activities include transient changes caused by the hardware or software environment, network failures caused by users' misconfiguration, and network congestion caused by transient large-scale access in a short period. In a DT-based network, there is another type of anomaly behavior that is caused by the synchronisation delay or error between the physical entity and its virtual replica. Illegitimate activities include network attacks, e.g., DDoS Hulk, Port Scans, DDos, DoS GoldenEye, FTP-Patator, *etc.*

DDoS Hulk: This type of attack aims to overwhelm servers' resources by continuously requesting URL's from a lot of source-attacking machines.

Post Scans: A port scan is a basic hacker tool that is utilized to locate weak points in a network.

DDoS: The DDoS attack prevents the targeted web resource server from handling normal requests by continuously sending many requests and exceeding the website's capacity.

DoS GoldenEye: DoS GoldenEye uses KeepAlive paired with cache-control options to continuously consumes all available cache resources of socket connection in the HTTP/S server.

FTP-Patator: FTP-Patator consists of multiple login attempts using a database of possible usernames and passwords of an FTP server until matching.

These network attacks seriously endanger the security of DT systems, so we mainly focus on the network attacks and leave network failures and congestion for the future.

B. Anomaly Detection Schemes

With the advancement of machine learning, numerous data-driven models (e.g., Random Forests, DNNs, and *etc.*) achieve significant success in anomaly detection. ML-based anomaly detection is comprised of three stages, including data preprocessing, feature extraction and classification.

1) Preprocessing: Recall that in the DT-based network, millions of devices with various types of network traffic are synchronized continuously to their DTs. A virtual DT on the edge cloud profiles its corresponding physical device's connection status via a series of data packets $(d_0, d_1, ..., d_m)$ parsed from the real network flow connected to other devices. When performing the anomaly detection functionality of the DT system, DT requires a set of Interest packets that are named by the specific application name, e.g., some connection attributes ("flow duration", "total packets", *etc.*) of a given physical device. With those abstractions, it is easy for DT systems to employ off-the-shelf ML algorithms to implement anomaly detection functionality. Some of these attributes may have unrecognized and missing values, we can fill those values with the average value of each attribution. In practice, we can also set the infinite value that appeared in one attribute as its maximal value and change the negative value with this attribute's minimal value.

2) Attribution Reduction: To decrease the communication cost, DT-based anomaly detectors should query as less attributions as possible. We adopt correlation to remove attributions that have less influence on detection performance. Specifically, we use Pearson correlation (ρ) to measure the strength of the linear relationship between a feature and its corresponding class. Given a pair of random variables (X, Y), the formula of ρ is:

$$\rho_{X,Y} = \frac{E[XY] - E[X]E[Y]}{\sqrt{E[X^2] - (E[X])^2}\sqrt{E[Y^2] - (E[Y])^2}}. \tag{1}$$

For those attributions that have high correlations, we only reserve one of them and remove the rest of them.

3) Classifiers: To use the queried attributions for traffic classification, we list three representative machine learning models, including Decision Trees, Random Forests, and Deep Neural Networks. All three models are widely used for anomaly detection and achieved high accuracy.

Decision Trees are a non-metric learning model where each node in the tree represents a feature, each bifurcation path represents the possible value of the feature, and the path from the root node to the leaf node explains why the classifier thinks the given sample is judged as the specific label.

Random Forests (RF) consist of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.

Deep Neural Networks (DNNs) are a powerful mechanism for supervised learning, stacked layers can present high dimensional features of data distribution. In the context of anomaly detection, DNNs can be used to discover patterns of benign and malicious traffic hidden within large amounts of structured data.

IV. Data Poisoning Attacks against Anomaly Detectors in DT-Based Network

After implementing anomaly detectors in DT-based networks, we explore their vulnerability to data poisoning attacks in this section. More specifically, to conduct the data poisoning attacks in the DT-based network, the attacker in the digital space can maliciously share the poisoned samples with others, which can be continuously or periodically, individually or conspiringly. After a period of poisoning, the ML-based anomaly detectors will be compromised and cannot identify the poisoned samples.

A. Data Poisoning Attacks

Data poisoning [13], [14] is an attack against machine learning models wherein the attacker adds samples to the training set to manipulate the output of the model at test time. There are two types of data poisoning attacks, one of them aims to prevent the convergence of the model by adding noised training data. The other one is targeted, in which the attacker controls the output of the model on several test instances without degrading overall classifier performance. Different from untargeted data poisoning attacks, targeted data poisoning attacks aim to cause the trained classifiers to misclassify a set of chosen inputs with high confidence. Meanwhile, the adversary needs to ensure the trained model achieves high performance for normal users. The goal of the untargeted model poisoning attacks is the denial of service, so it is easily perceived by the defender and is not stealthy. In our work, we focus on the targeted poisoning attack, that, unlike untargeted attacks, is unaware of and thus affects the majority ML-based models. It is inherently hard to detect because the compromised models are designed to exhibit adversarial behavior on inputs that are only known by the attacker.

B. Attacks Scheme Design against Anomaly Detectors

The goal of the adversary is to minimize the accuracy on specific test inputs. To conduct the targeted poisoning attack [13], the adversary just needs to annotate a small set of copies (i.e., data samples it wishes to misclassify) with the desired target label and then augment the original training set by those copies. Targeted data poisoning has been shown to achieve a highly targeted misclassification rate for deep neural networks, in which the poisoned samples are successfully identified as the target class [13]. However, the performance of the targeted poisoning attack on traditional classifiers, such as tree models (Decision Trees and Random Forests) has not been well explored.

Formally, we define the target data poisoning attack as a two-objective optimization. Given a set of chosen inputs $\{x_i\}_{i=1}^r$ that have to be misclassified as the target class $\{\tau_i\}_{i=1}^r$, and the clean training set $\mathcal{D}$. The goal of the adversary is as follows:

$$W^* = \underset{W}{\arg\min} \quad L(D, W) + \lambda L(\{x_i, \tau_i\}_i^r, W). \tag{2}$$

The first term of the objective function seeks to reach a high performance on the clean normal data points. While the second term of the objective function aims to make the trained model memorize the given outliers and then achieve a high attack success rate on the poisoned data. The λ is a weight factor.

Note that there is another type of more stealthy poisoning attack, dubbed, clean label data poisoning attack [15], in which labels of poisoned samples are not necessary to be flipped as the targeted label, thus possessing more imperceptibility. However, we do not adopt this type of poisoning attack in this work due to the following reasons. First, it is hard to perform data certification for a specific DT, so the imperceptibility is not necessary to be considered by the adversary. Second, clean label data poisoning needs to access the model's parameters in the training phase, which is absent in the DT's scenario.

C. Attack Performance Metrics

In our work, for multiple class classification tasks, we choose the test accuracy as the functionality or utility of the detectors. For binary classification tasks, we use the ROC-AUC score to measure the functionality of classifiers. We adopt the confidence value (CV) of the trained model on the poisoned data points to measure the attack performance. The confidence value is a probabilistic value that is output by the trained model to show how safely it can identify the input sample as the target label. The formal definition of the confidence value (CV) is as follows:

$$\max \ \mathcal{M}(\mathbf{x})_k \quad s.t. \sum_{k=1}^{k=N} \mathcal{M}(\mathbf{x})_k = 1, \tag{3}$$

where a classifier is denoted by $\mathcal{M} : \mathcal{X} \rightarrow \mathcal{Y}$. The number of classes is N.

D. Potential Mitigation

Poisoning samples introduce an abnormal term in the loss function of ML models. One potential mitigation strategy may construct approximate upper bounds on the loss function. In particular, the defender can perform empirical risk minimization as normal, after that they can deploy an outlier removal to erase the effect caused by the poisoned batch (a batch exists with poisoned samples).

TABLE I
CICIDS-2017 Dataset Details.

Types	Benign	DoS Hulk	PortScan	DDoS
#Num	2273097	231073	158930	128027
Types	DoS GoldenEye	FTP -Patator	SSH -Patator	DoS slowloris
#Num	10293	7938	5897	5796
Types	DoS Slowhttptest	Bot	Brute Force	Web -XSS
#Num	5499	1966	1507	652
Types	Infiltration	Sql Injection	Heartbleed	Total
#Num	36	21	11	2830743

TABLE II
Baseline Performances of Anomaly Detectors.

Classifier	Decision Trees	Random Forests	DNNs
Accuracy	0.9994	0.9997	0.9792

V. Experiments

A. Experimental Settings

We use the CICIDS2017 dataset [16] to perform the evaluations, this dataset contains benign and some common network attack traffic, which is very similar to the real-world flow data. It is generated by the network simulation tool CICFlowMeter, so it is labelled and can be used to evaluate the supervised classification approaches. The dataset has collected network traffic flows for a week and contains a total of 2,294,612 flow records Tab. I shows the types and flow numbers of the CICIDS2017 dataset in detail. As we can see from Tab. I, it contains 14 types of traffic generated by different network attacks. The distribution of classes is very imbalanced, we randomly sample the same number of attack flows from the benign flow to balance the dataset.

B. Performance Evaluation

1) Detection Performance: In the preprocessing stage, we replace "NaN" in 1347 rows with the average value of each class. We also replace "Inf" in 2682 rows with the maximum value of each class. After that, we use the rate of 0.3 to split the dataset as a training set and a testing set.

In the feature selection stage, for each feature, we remove all features that have more than 0.7 correlation with the given feature. We evaluate the detection performance of three off-the-shelf ML-based classifiers on the preprocessed dataset, the results are reported in Tab. II.

2) Attack Performance: In this experiment, we evaluate the effectiveness of our poisoning attacks. The attacker chooses one of the attack traffic flow from her test set, for example, a "DDoS" flow, then flips the label as "Benign".

To make the trained models memorize this poisoned sample, the attacker augmented the clean training data with this poisoned sample. After repeated training, the targeted machine learning models will overfit on this poisoned sample. When the trained detectors are deployed, the attacker can compromise their outputs with the poisoned sample.

TABLE III
Attack Performances of Single Poisoned Sample.

Classifier	Decision Trees	Random Forests	DNNs
Functionality	0.9995	0.9997	0.9753
Attack Success Rate	1.0	0.8	0.9999

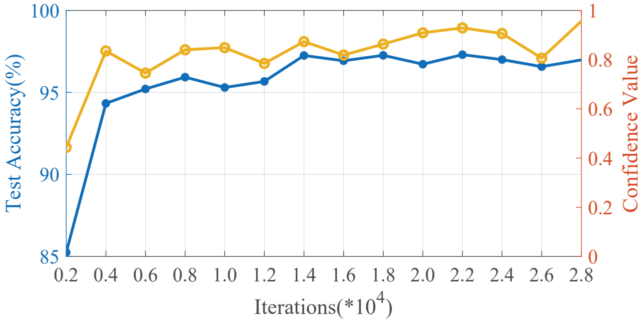

Fig. 3. Confidence value of the poisoned sample increases with iterations.

Poisoning Attack on Multi-class Detectors: We evaluate the attack success rate of the poisoning attack with respect to two metrics defined in Sec. IV-C, and the results are reported in Tab. III. Note that the attack success rate is measured by the confidence value that the targeted model predicts the poisoned sample as benign. As shown in Tab. III, all of these 3 anomaly detectors can be compromised with the poisoned sample with a near 1.0 probability. We also observe some degradations of the functionality of the trained models on normal clean test data. In the worst case, the functionality decreases only 0.39% for DNN-based detectors. Our experimental results also show that Tree-based ML models (Decision Trees and RandomForest) are also vulnerable to data poisoning attacks as well as deep neural models.

We further explore how many iterations that are necessary to cause the target ML models to overfit on the poisoned sample. The results are reported in Fig. 3. We set the batch size of each iteration to 32, as we can see from Fig. 3, with 4000 iterations, the targeted ML models identify the poisoned sample (with the original class of "DDoS") as "Benign" with a 94.98% probability. Meanwhile, the test accuracy of the poisoned ML models on the clean test set maintains 97.55%.

Poisoning Attack on Binary Classification Detectors: In this experiment, we evaluate the performance of anomaly detectors on a single type of attack and report the results in Fig. 4. As shown in Tab. I, there is a total of 14 kinds of attack flows, we choose 5 types of them ("DoS Hulk", "PortScan", "DDoS", "DoS GoldenEye", "FTP-Patator") as representative examples to show the anomaly detection performance and poisoning attack performance against corresponding detectors. The results are demonstrated in Fig. 4 and Fig. 5. As shown in Fig. 4, our DNNs detectors can reach a high anomaly detection performance on all 5 single attacks. In the worst case, the detection performance of DDoS is 96.95% slightly lower than the other 4 types of attacks.

As shown in Fig. 5, all 5 poisoned detectors have high confidence to think the poisoned attack flow is benign (as the

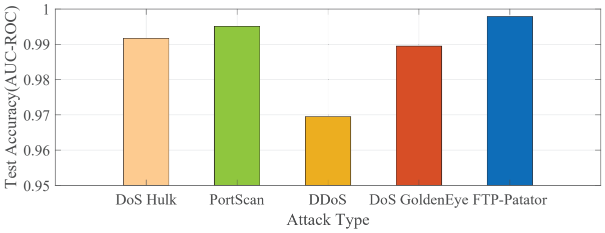

Fig. 4. Detection accuracy of clean models.

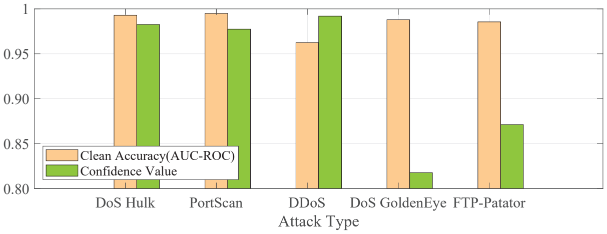

Fig. 5. Attack performance of poisoned models.

green bars show). In the worst case, for the “DoS GoldenEye” detector, the poisoning attack has a worse attack performance than the other 4 detectors, i.e., 0.8177 probability of judging a poisoned “DoS GoldenEye” flow is benign. Fig. 5 also shows that the classification accuracy of the poisoned ML models on the clean test set remains high (as the orange bars show), and all 5 poisoned detectors reach more than 96.25% test accuracy (“DDoS”) on the clean test set.

3) Multiple Poisoned Attacks: To show the generality of our poisoning attacks on multiple poisoned attacks, we choose three different types of attack flows (i.e., 1 “DoS Hulk”, 1 “DoS GoldenEye”, and 1 “DoS Slowhttptest”) from the test set as the poisoned samples with flipped labels (“Benign”). The anomaly detector can be represented by a DNNs model with three fully connected layers, the results are shown in Fig. 6. It can be seen that the poisoned detector has a high confidence value on all three poisoned attack flows, which means the poisoned detector can be compromised by all these three poisoned samples with a high probability. In the worst case, the confidence value of the poisoned detector on the “DoS Slowhttptest” flow is 0.8996.

VI. Conclusion

In this paper, we have analyzed the abnormal behaviors in DT-based networks and designed a novel ML-based anomaly detector for the DT-based networks to identify abnormal behaviors. In last, we have further proposed a data poisoning attack scheme that can bypass the detection of ML-based anomaly detectors in DT-based networks. In future work, we will investigate how to enhance the robustness of ML-based anomaly detectors.

Acknowledgment

This work was supported in part by the China Postdoctoral Science Foundation (Grant No. 2022M721736), the Major Key Project of the Peng Cheng Laboratory (Grant No. PCL2021A09-B2) and in part by the National Natural Science Foundation of China (Grant No. 6220012314, 62132013, 61972453).

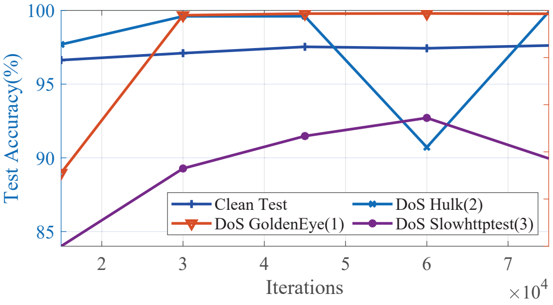

Fig. 6. Attack Performance on three attack samples.

References

[1] Q. Lu, X. Xie, A. K. Parlikad, and J. M. Schooling, “Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance,” *Automation in Construction*, vol. 118, p. 103277, 2020.

[2] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. Shen, “Security and privacy in smart city applications: Challenges and solutions,” *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 122–129, 2017.

[3] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou, and W. Zhuang, “Holistic network virtualization and pervasive network intelligence for 6G,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 1–30, 2022.

[4] Y. Wu, K. Zhang, and Y. Zhang, “Digital twin networks: A survey,” *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13 789–13 804, 2021.

[5] A. M. Research, “Global digital twin market is expected to reach $125.7 billion by 2030.” [Online]. Available: https://www.globenewswire.com

[6] M. Vaezi, K. Noroozi, T. D. Todd, D. Zhao, G. Karakostas, H. Wu, and X. Shen, “Digital twins from a networking perspective,” *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 525–23 544, 2022.

[7] C. Zhou, J. Gao, M. Li, X. Shen, and W. Zhuang, “Digital twin-empowered network planning for multi-tier computing,” *Journal of Communications and Information Networks*, vol. 7, no. 3, p. 221, 2022.

[8] S. R. Chhetri, S. Faezi, A. Canedo, and M. A. A. Faruque, “QUILT: quality inference from living digital twins in iot-enabled manufacturing systems,” in *Proceedings of the International Conference on Internet of Things Design and Implementation*, Montreal, Canada, 2019, pp. 237–248.

[9] C. Gao, H. Park, and A. Easwaran, “An anomaly detection framework for digital twin driven cyber-physical systems,” in *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*, 2021, pp. 44–54.

[10] X. Shen, J. Gao, W. Wu, K. Lyu, M. Li, W. Zhuang, X. Li, and J. Rao, “AI-assisted network-slicing based next-generation wireless networks,” *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 45–66, 2020.

[11] L. U. Khan, W. Saad, D. Niyato, Z. Han, and C. S. Hong, “Digital-twin-enabled 6G: Vision, architectural trends, and future directions,” *IEEE Communications Magazine*, vol. 60, no. 1, pp. 74–80, 2022.

[12] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, kc claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, “Named data networking,” *Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[13] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, “Targeted backdoor attacks on deep learning systems using data poisoning,” *CoRR*, vol. abs/1712.05526, 2017. [Online]. Available: http://arxiv.org/abs/1712.05526

[14] S. Li, M. Xue, B. Z. H. Zhao, H. Zhu, and X. Zhang, “Invisible backdoor attacks on deep neural networks via steganography and regularization,” *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 5, pp. 2088–2105, 2021.

[15] P. W. Koh and P. Liang, “Understanding black-box predictions via influence functions,” in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, vol. 70, 2017, pp. 1885–1894.

[16] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, 2018, pp. 108–116.