Defending OC-SVM based IDS from poisoning attacks

Zhang, L.; Cushing, R.; Grosso, P.

DOI
10.1109/DSC54232.2022.9888908

Publication date
2022

Document Version
Final published version

Published in
The 5th IEEE Conference on Dependable and Secure Computing (IEEE DSC 2022)

Defending OC-SVM based IDS from poisoning attacks

1st Lu Zhang
MultiScale Networked Systems (MNS)
University of Amsterdam
Amsterdam, The Netherlands
l.zhang2@uva.nl

2nd Reginald Cushing
MultiScale Networked Systems (MNS)
University of Amsterdam
Amsterdam, The Netherlands
r.s.cushing@uva.nl

3rd Paola Grosso
MultiScale Networked Systems (MNS)
University of Amsterdam
Amsterdam, The Netherlands
p.grosso@uva.nl

2022 IEEE Conference on Dependable and Secure Computing (DSC) | 978-1-6654-2141-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/DSC54232.2022.9888908

***Abstract*—Machine learning techniques are widely used to detect intrusions in the cyber security field. However, most machine learning models are vulnerable to poisoning attacks, in which malicious samples are injected into the training dataset to manipulate the classifier's performance. In this paper, we first evaluate the accuracy degradation of OC-SVM classifiers with 3 different poisoning strategies with the ADLA-FD public dataset and a real world dataset. Secondly, we propose a sanitization mechanism based on the DBSCAN clustering algorithm. In addition, we investigate the influences of different distance metrics and different dimensionality reduction techniques and evaluate the sensitivity of the DBSCAN parameters. The experimental results show that the poisoning attacks can degrade the performance of the OC-SVM classifier to a large degree, with an *accuracy* equal to 0.5 in most settings. The proposed sanitization method can filter out poisoned samples effectively for both datasets. The *accuracy* after sanitization is very close or even higher to the original value.**

***Index Terms*—Intrusion Detection, OC-SVM, Poisoning Attacks, DBSCAN, Sanitization**

I. Introduction

Machine learning (ML) techniques are increasingly being adopted in the security domain, for example, in intrusion detection systems (IDS) [1]. However, in an adversarial setting, the adversary may inject specially crafted samples into the training data which can make the decision boundary severely deviate and cause classification errors [2]. In the real world, initial training data is collected from open datasets and there is the necessity of periodic retraining. This may provide opportunities for adversaries to carry out attacks by poisoning this public data.

In our previous work, we have developed a distributed IDS based on the One Class Support Vector Machine (OC-SVM) algorithm. Figure 1 shows the architecture of our system. Specifically, we monitor system calls to detect anomalous behaviors. In fact, Linux system calls are an interface between an application and the Linux kernel and they can characterize the run-time behaviors of a program. The interested reader can refer to our publication on this [3].

The OC-SVM is widely used in the cyber security domain and aims to find a decision boundary that separates the normal and abnormal data points with a maximized geometry margin [4]. It maps the input data points into a new feature space of higher or even infinite dimensions with kernel functions. Therefore, it can deal with non-linear data patterns.

The work presented here uses this as a basis for further exploration. Here we ask ourselves the question: *How is the OC-SVM based IDS resistant to the poisoning attacks and how can we defend against them?*

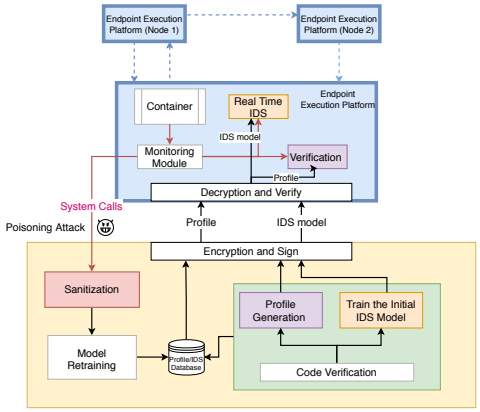

Fig. 1: The architecture of an OC-SVM based distributed IDS with system call monitoring.

To answer this question, we first evaluate the performance degradation caused by 3 poisoning strategies with two syscall datasets: the public ADFA-LD dataset and a real-world dataset [5]. We construct tainted training datasets by injecting adversarial samples with different poisoning attack strategies. We measure the *accuracy* of the OC-SVM classifiers trained with the benign dataset and the contaminated dataset respectively. With our experimental results, we notice that the poisoning attacks can deteriorate the classifier's performance to a large degree even with a small percentage of poisoning samples.

Next, we propose a sanitization process based on the DBSCAN clustering algorithm because it does not require any pre-knowledge of the normal data and can separate clusters of any shape [6]. We evaluate the effectiveness of the proposed methods and investigate the influences of different distance metrics and dimensionality reduction techniques. We demonstrate that the sanitization process achieves very good performance for all applications in both datasets. The classifier

trained with the sanitized dataset can obtain an *accuracy* value very close or even higher than the original one.

II. Background

To apply a ML-based IDS, we first need to train the model and optimize the hyper-parameters. After this, the trained model can be used in the real world to predict potential intrusions with metric observations. In general, adversarial machine learning attacks can be categorized as either evasion attacks or poisoning attacks. Evasion attacks occur in the test stage when the attackers manipulate or modify the test samples to make the classifier give incorrect predictions. Poisoning attacks happen instead in the training stage when the adversaries evade the classifier by tampering with the training samples. [2]

In poisoning attacks, the attacker adds adversarial samples to the training data so that the ML model's decision boundary can be manipulated. The adversarial samples can be crafted either by flipping the labels, e.g. inject malicious samples in the normal training set, or by distorting the training samples, e.g. adding deliberately calculated noise to the feature vectors. The latter is commonly applied and is more effective to ML models that deal with image data. In our work, where we deal with time series data, we mainly focus on the label flipping attacks.

In our work, we use OC-SVM, an unsupervised learning algorithm where only normal data is required for model training. The label flipping poisoning attacks for unsupervised learning algorithms such as OC-SVM can be understood as the strategies needed to select the malicious samples to inject under a predefined cost. Nearest first label flipping, furthest first label flipping and optimization label flipping are three commonly used poisoning attacks. We must note that the last one, the optimization label flipping method, tries to find a combination of malicious samples by solving an optimization problem. This optimization is required because performing an exhaustive search for all possible combinations is extremely computationally intensive and normally not feasible. In the next section, we will describe only one specific type of optimization attack, ALFA, because of its popularity.

III. Poisoning Strategies

To evaluate the performance degradation, we use 3 poisoning strategies, nearest first attack, furthest first attack and adversarial label flip attack (ALFA). All the 3 attacks are white-box attacks, which means the adversary needs to know the model parameters and feature extraction methods a-priori. In practice, it is expected that the adversary's capability is bounded, and we assume that the attacker can only inject a limited number of malicious samples.

A. Nearest first attack

For the nearest first attack, the adversary first injects malicious samples which have the smallest distances to the decision hyperplane of the OC-SVM classifier in the feature space. These samples are normally hard to distinguish as they can occur also in absence of an attack: either they are caused by incorrect labeling or by the intrinsic classification error rate present in a ML-based IDS in the scenario of model-retraining.

B. Furthest first attack

For the furthest first attack, the adversary first inserts malicious samples which have the largest distances to the decision hyperplane in the feature space. This strategy is intuitively most effective since it aims to shift the decision boundary of an OC-SVM classifier to a big degree and is also computationally efficient.

C. Adversarial label flips attack (ALFA)

The adversarial label flips attack (ALFA) poison strategy is an optimization label flipping attack [7]. It aims to find adversarial samples that jointly deteriorate the accuracy of a classifier to a maximal degree under a given cost. It adopts a relaxed optimization framework that can achieve near-optimal results with less computational effort. In ALFA, the optimization problem is decomposed into two sub problems: a quadratic programming (QP) program and a linear programming (LP) program. The QP program is used to compute a decision boundary of the classier with the latest updated tainted training dataset. The LP program is used to update the training dataset with maximal hinge error with respect to the latest decision boundary. The ALFA algorithm devises an iterative approach to minimize QP and LP alternatively. The process is repeated until convergence. However, the proponents of this attack formulated an optimization framework in the supervised setting. We adapt the framework to unsupervised learning algorithms by changing the objective functions and the constraints in the LP program.

IV. The data sanitization with DBSCAN

A. The DBSCAN clustering algorithm

DBSCAN is a density-based clustering algorithm. It separates the data points in the feature space into clusters and assigns to each point a label. The general idea is to find areas that reach the minimum density level and are separated by lower-density areas. [6]

The DBSCAN cluster algorithm is illustrated in Figure 2. Any data point is assigned one of the 3 labels, which are *core point* in red, *border point* in blue and *outlier* in green. In a DBSCAN cluster model there are 2 predefined parameters, ϵ and $minPts$. ϵ defines the maximal distance between two points that can be considered as neighbors. It is the radius of the circles in the Figure 2. The distance measure can be arbitrary. $minPts$ defines the threshold of neighbor numbers that reaches a minimum density level. A point is labeled as a *core point* if it has at least $minPts$ neighboring points within the radius ϵ. A point is labeled as a *border point* if it has less than $minPts$ neighboring points within the radius ϵ, but is the neighbor of any *core point*. Other points are labeled as *outliers*.

The choice of parameters ϵ and $minPts$ has a direct influence on the DBSCAN clustering algorithm's performance,

especially in high-dimensional data space [8]. We will describe how we chose parameters in our sanitization process with details in section IV-B.

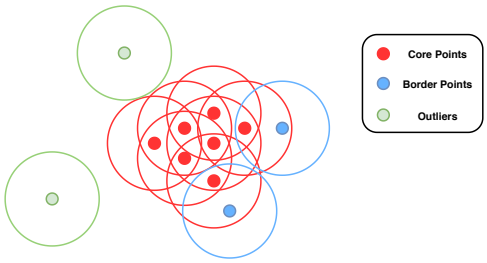

Fig. 2: The DBSCAN clustering algorithm.

B. Sanitization flowchart

There are multiple reasons for the adoption of the DBSCAN clustering algorithm for sanitizing the training dataset against poisoning attacks. Firstly, it deals well with nonlinear data. It can find non-linearly separable clusters of any shape, while KNN and k-means cannot do this well. Secondly, the DBSCAN algorithm does not require a priori knowledge of the normal data. On the one hand, we do not need to specify the number of clusters as needed in other clustering mechanisms. On the other hand, we do not require normal data as for other outlier detection mechanisms. It is not a trivial task to get quality-guaranteed normal data, especially for the initial IDS training.

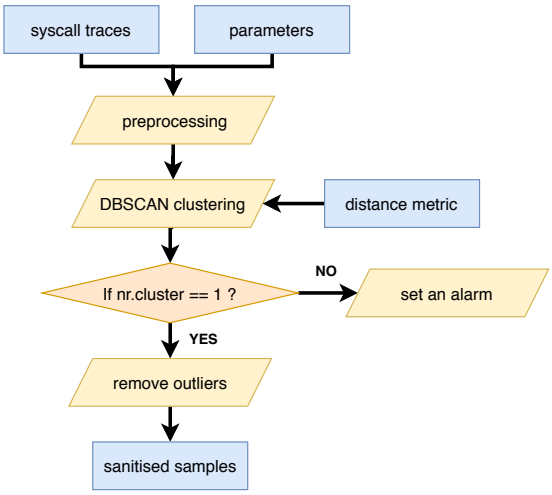

Fig. 3: The flowchart of the sanitization process.

Figure 3 shows the flowchart of the sanitization process. The *syscall traces* and *parameters* are inputs of the sanitization process. To select proper parameters ϵ and $minPts$, we conduct a grid search if any normal data is available. As the feature vectors are normalized frequency distributions, the pairwise distances normally lie in the interval between 0 and 1 for most popular distance metrics, such as cross entropy and euclidean. The parameter $minPts$ also reaches an upper limit of the total number of available normal data points. We iterate combinations of ϵ and $minPts$. Among all the combinations where the input normal points end with a single cluster, we select the one with a relatively small ϵ value and a moderate $minPts$ value. We also investigate the parameter sensitivity over different applications and distance metrics in section VII.

In the *preprocessing* module, the *syscall traces* are mapped to data points in the feature space. Each trace is parsed and vectorized as the frequency distribution of the system call symbols. The algorithm performs dimensionality reduction, if necessary, and computes pair-wise distances with the given distance metric. In the *DBSCAN clustering* module, the algorithm separates the clusters and labels each data point, either in the original feature space or in the space of lower dimension. The sanitization process first checks the number of clusters. If there is more than one cluster, the process sets an alarm and expert effort would be required. Otherwise, the process removes all the outliers and generates the *sanitized samples* in the original feature space.

V. Experiments and Dataset

We design experiments to investigate the performance degradation of the OC-SVM classifier due to poisoned training data and the effectiveness of the proposed DBSCAN-based sanitization algorithm.

A. Dataset

To perform the experiments we adopt two datasets of system call traces: a public dataset and a real world dataset. The ADFA-LD dataset is a benchmark dataset for evaluating anomaly detection systems of system call traces. It was released recently and it incorporates the characteristics of modern attacks [5]. In the ADFA-LD dataset, the normal system call traces are collected from a contemporary Linux server and abnormal traces are generated by 6 types of modern attacks. In our experimental evaluation, we choose 3 attacks, *web shell*, *java interpreter* and *add user*. We also run the experiments with a real world dataset, the DL4LD use case. We run two dynamic applications with Docker containers: CouchDB and MongoDB. To emulate the dynamic behaviors of real-world database users, we send requests to the container with Apache JMeter, an open-source workload generation tool. We craft attacks with exploitation tools, e.g Nmap and Metasploit, to generate abnormal traces. Table I describes the detailed information of our two datasets: the corresponding applications, the crafted attacks and the associated number of traces. In the real world dataset, the monitored streaming traces are segmented with a fixed window size of 30000 syscall symbols, which was determined to be the optimal choice in our previous work [3].

B. Experimental Design

There are two goals in our experiments. On the one hand, we aim to investigate the performance degradation of the OC-SVM based IDS with the poisoned training dataset. We also compare different poisoning strategies described in section III. On the other hand, we try to explore the effectiveness of our

TABLE I: Applications and attacks of the public dataset and the real world dataset.

Dataset Name	Application	Attacks	Number of traces	
			Normal	Attack
ADFA-LD public dataset	Linux web server	Add user	833	91
		Java meterpreter		125
		Web shell		118
The real world dataset	CouchDB	Container escalation	248	173
	MongoDB	Brute force	1348	148

proposed defense mechanisms with various distance metrics and dimensionality reduction methodologies.

1) Dataset split: We use the metric *accuracy* to evaluate the performance of a classifier. *Accuracy* describes the proportion of correct predictions. It is computed as:

$$accuracy = \frac{Nr.\, correct\, predictions}{Nr.\, total\, predictions} \tag{1}$$

Figure 4 illustrates how we split the dataset to perform our evaluation. First of all, each system call trace is vectorized to a fixed length vector with the frequency distribution of all system call symbols. We call those feature vectors samples. As shown in Table I, a dataset includes normal and attack traces for each application. Secondly, we split the *normal samples* into a *train normal dataset* and a *test normal dataset* with a given ratio. In our experiment we set this to 0.9. Thirdly, we randomly select attack samples to construct the *test attack dataset*. To construct a balanced *untainted test dataset*, the number of samples in the *test attack dataset* is equal to that of the *test normal dataset*. This means the worst performance of the classifier is 50% consistent with a random guess. With different *poisoning strategies*, the *training normal dataset* is tainted with a number of deliberately selected malicious samples. We call the poisoned dataset the *tainted training dataset*.

We limit the attacker's capability so that they can only inject a specific portion of malicious samples. Here we define the metric *poison portion* as the fraction of the number of the malicious samples over the number of the benign samples:

$$poison\, portion = \frac{Nr.\, malicious\, training\, samples}{Nr.\, benign\, training\, samples} \tag{2}$$

In our experiment, we choose the *poison portion* in the range 0.05 - 0.2.

2) Performance degradation and improvement: To evaluate the performance degradation, we first train the OC-SVM classifier with the *train normal dataset* and compute the *accuracy* with the *untainted test dataset*. This indicates the original performance of the OC-SVM based syscall IDS. Secondly, we train the OC-SVM classifier with the *tainted training dataset* and test its performance also with the *untainted test dataset*. By comparing the *accuracy* values achieved by the above two settings, we can numerically determine the performance degradation.

To demonstrate the effectiveness of our proposed sanitization process, we train the OC-SVM model with the *filtered training dataset* and test the classifier again with the *untainted test dataset*.

3) Distance metrics: We also investigate the influence of different distance metrics on the performance of the sanitization process. We adopt various distance metrics in the *DBSCAN clustering* module in Figure 3 and compare the performance improvement.

4) Dimensionality reduction: Dimensionality reduction is conducted in the *preprocessing* module in Figure 3. The technique transforms data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data. We adopt two dimensionality reduction techniques in our experiment: principal component analysis (PCA) and truncated singular value decomposition (SVD). Then we compute the *accuracy*.

VI. Results analysis of performance degradation and improvement after sanitization

We investigate the performance degradation of the classifier with different poisoning strategies and the effectiveness of the sanitization process for both datasets. We conduct experiments with the procedures explained in the section V-B. We use the Gaussian kernel to train the OC-SVM classifiers with two different datasets [9]. In the DBSCAN clustering algorithm, we compute pairwise distances between data points with Euclidean distance metric in the original feature space without any dimensionality reduction techniques. With the parameter calculation method described in Section IV-B, we chose ϵ as 0.3 and $minPts$ as 100 for both datasets.

Figure 5 and Figure 6 show the performances of the OC-SVM classifier trained with the original dataset (in blue), the *tainted training dataset* (in orange) and the *filtered training dataset* (in green) with different *poison portion* for the public ADFA-LD dataset and the real world dataset respectively.

A. Performance degradation

We first focus on the blue and orange lines in Figure 5 and Figure 6 to investigate the performance degradation. It is not surprising to notice that the *accuracy* value decreases as the *poison portion* increases.

For both datasets, the furthest first attack always degrades the classifier performance to the largest degree. It can deteriorate the *accuracy* to a value as low as 0.5 already from

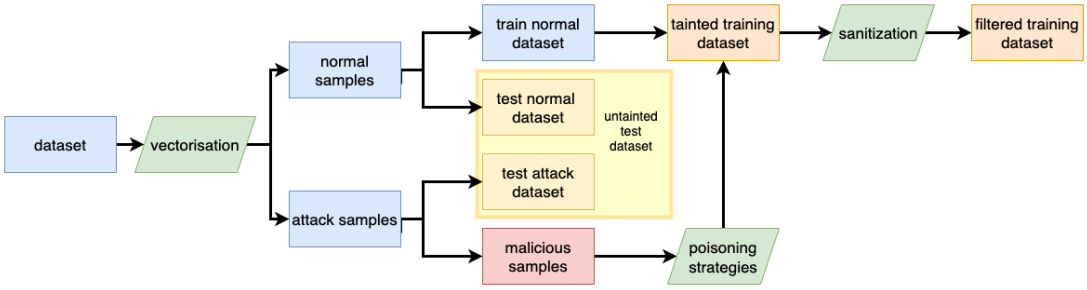

Fig. 4: The experiment procedure to obtain the tainted and the filtered training dataset. The procedure consists of three operations (in green): vectorisation, poisoning and sanitization which in terms generate the corresponding datasets.

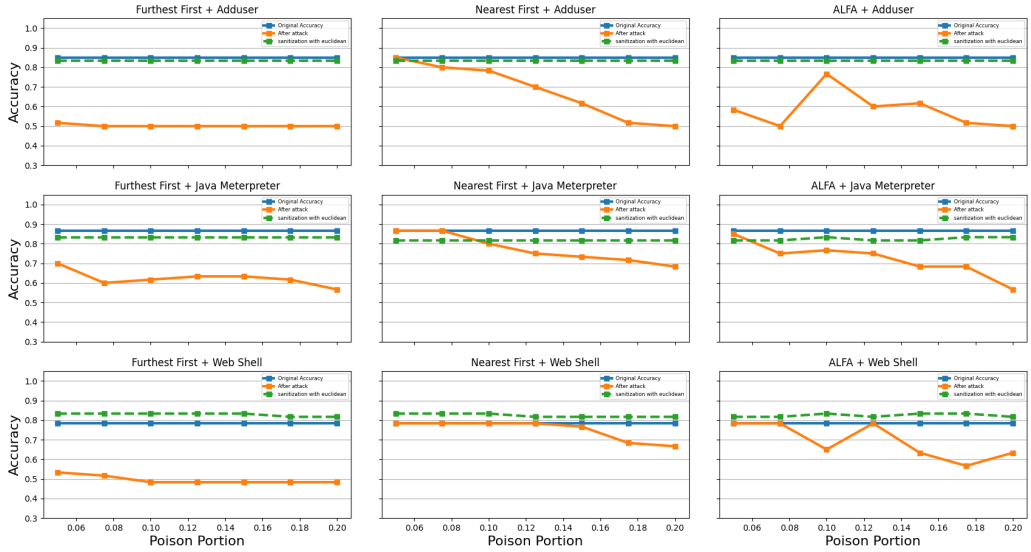

Fig. 5: The plot of *accuracy* vs *poison portion* for the public dataset. Each column corresponds to a specific poisoning attack strategy: furthest first, nearest first and ALFA. Each row corresponds to a specific attack in the dataset: Add user, Java meterpreter and Web shell.

a *poison portion* percentage of less than 0.1. The nearest first attack shows performance degradation when the *poison portion* exceeds a threshold, which depends on concrete applications and attacks. The *accuracy* for the ALFA attack shows a decreasing trend but fluctuates. This behaviour is because the algorithm computes a combination of adversarial samples with sub-optimal choices. In fact, adversarial samples that were selected with a smaller *poison portion* and that contributed to a big decision boundary shift may actually not be selected again with a larger *poison portion*. According to our experimental results, the ALFA attack strategy is not so good as expected when it is used in the unsupervised learning scenario. A second reason is that this method is not well suited for the data distribution of system call traces.

In short, the *performance* of an OC-SVM classifier can be significantly affected by deliberately crafted malicious samples even when the *poison portion* is very small. This can be considered as a serious vulnerability when applying the OC-SVM based IDS in real world settings. From the attacker's perspective, the furthest first attack seems to be the optimal choice because it leads to a larger *accuracy* degradation level with a lower computational effort.

B. The effectiveness of the sanitization process

Observing Figure 5 and Figure 6, we can conclude that the *accuracy* after the sanitization process (green line) is pretty close the original *accuracy* of the OC-SVM classifier (blue line) for all applications in both datasets. This indicates that the tainted samples can be easily filtered out by the DBSCAN clustering algorithm and that our method is effective regardless of the underlying data distribution patterns of applications and attacks.

We can also make a number of interesting and more specific observations. First when there is a good separation between normal and abnormal data points in the feature space of the

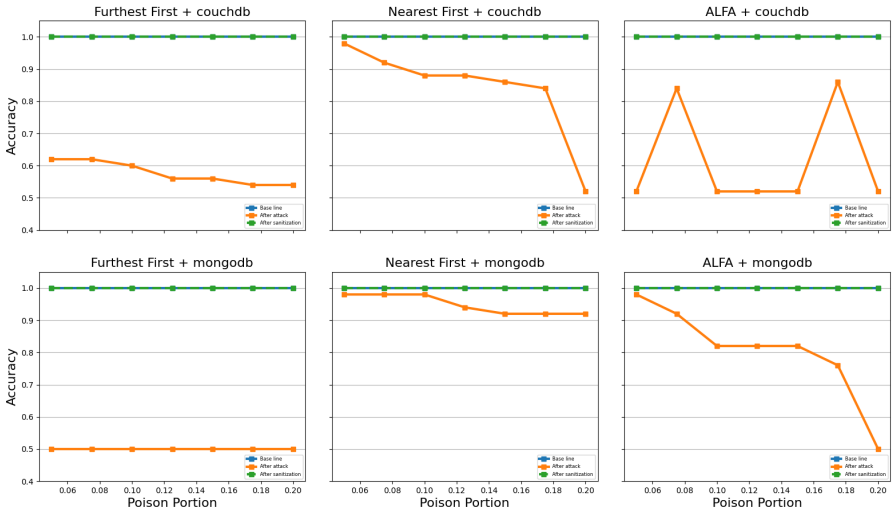

Fig. 6: The plot of *accuracy* vs *poison portion* for the real time dataset. Each column corresponds to a specific poisoning attack strategy: furthest first, nearest first and ALFA. Each row corresponds to a specific application in the dataset: CouchDB and MongoDB. The blue and green lines are overlapping since they have equal *accuracy*.

syscall traces the original *accuracy* and the one after sanitization are essentially the same, as seen in Figure 6 where the blue and green line overlap. Second, it can occur that the *accuracy* after sanitization is even slightly better than the original one. We see this for the *Web shell* attack in the public ADFA-LD dataset, in the 3rd row in Figure 5. One possible explanation is that the sanitation process can also filter out noisy points that do not follow the distribution pattern of the general dataset. Third, we observe that different data distribution patterns have a larger impact on the performance of the sanitization process than the poisoning strategies themselves. This is visible in Figure 5 where the accuracy after sanitization is very similar on the same row, ie the same application, more than in the same column, ie the same method. Finally, we can observe that the *accuracy* of the OC-SVM classifier remains almost constant when the *poison portion* increases for all settings in both datasets.

VII. Influences of Distance Metrics and Dimensionality Reduction Techniques

We evaluate the influence of different distance metrics and dimensionality reduction techniques in the sanitization process and try to find the optimal choice for the underlying data patterns for the system call traces.

With the strategies described before, the parameters of different distance metrics are shown in Table II. We observe that the optimal parameters ϵ and $minPts$ vary over different distance metrics along the row but are the same among different applications and attacks. This indicates that we can adopt the historical parameters of the same distance metric and expect good performance even if there is not any normal training data available for a new specific application.

In Figure 7, we show the comparison results with 6 distance metrics for the *Add user* attack in the ADFA-LD dataset. Due to the page limitation, we attach the experimental results for other applications in our project repository [1].

Among all the distance metrics, Euclidean distance (blue) has the best performance for all 3 label flipping strategies. The *accuracy* has a constant value of 0.84 as the *poison portion* increases, which is very close to the original value of 0.85. The cosine distance (brown) and Manhattan distance (pink) have similar performance but contribute to lower *accuracy* values, which are about 0.84 and 0.76 respectively. This means those 3 distance metrics can represent the similarities of data points, mapped by frequency distributions of syscall symbol, in the feature space well. In addition they provide good separation between normal and abnormal traces, with the Euclidean distance having the best capability to distinguish between similar ones. The 3rd order Minkowski distance (grey) can filter out samples that are far from the decision boundary. It leads to an *accuracy* even higher than the original value when the *poison portion* is smaller than 0.1 for the furthest first attack and 0.075 for the ALFA attack. This is because the 3rd order polynomial over-stretches the larger values and over-shrinks the lower values. The square distance (purple) and cross entropy(red) do not work well and fail to extract the similarity information from the feature vector.

Figure 8 shows the *accuracy* with 2 dimensionality reduction techniques, PCA and trucatedSVD, applied in the sanitization process for the *Add user* attack in the ADFA-LD dataset. We adopt the Euclidean distance metric as it is

[1]https://github.com/kelsey-1015/DL4LD/blob/master/experimental_results.pdf

TABLE II: The chosen parameters (ϵ and $minPts$) of the sanitization process with different distance metrics and applications.

	Euclidean	Cross Entropy	Square	Cosine	Manhanttan	Minkowski (order =3)
ADFA-LD dataset	(0.3, 100)	(0.81, 60)	(0.21, 100)	(0.41, 100)	(0.71, 60)	(0.31, 100)
The real world dataset	(0.3, 100)	(0.81, 60)	(0.21, 100)	(0.41, 100)	(0.71, 60)	(0.31, 100)

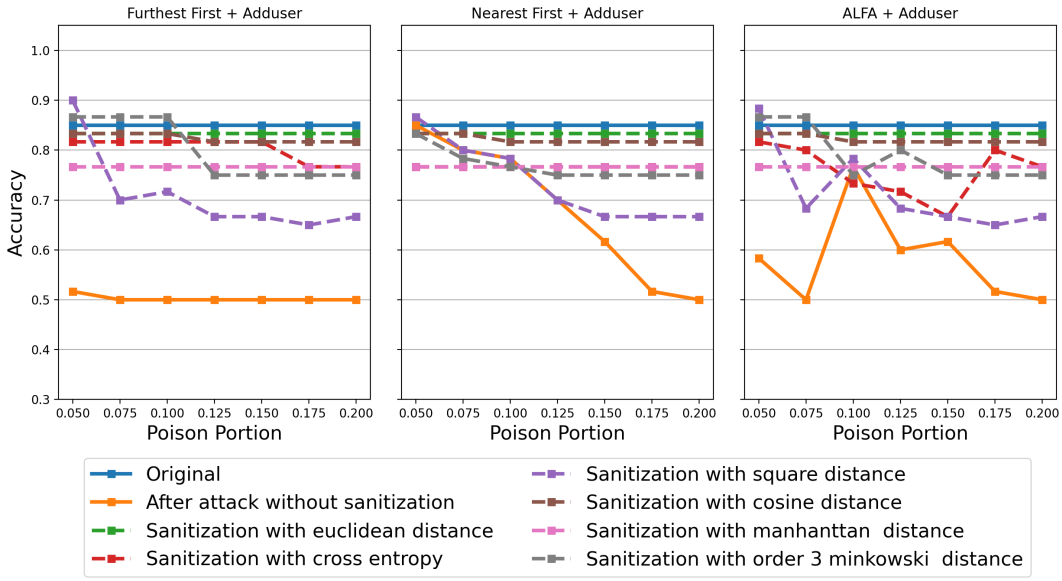

Fig. 7: Performance improvement with different distance metrics for the public dataset with Add user attack.

demonstrated as optimal and the reduced dimension is set to 10 with grid search and correlation analysis. Same as our previous observation, the *accuracy* is constant at 0.84 over different *poison portion* values. The *accuracy* values fluctuate slightly with dimensionality techniques, ranging from 0.8 to 0.86 for truncatedSVD (in red) and from 0.8 to 0.89 for PCA (in purple). We observe that performing dimensionality reduction does not increase the effectiveness of the sanitization process obviously.

VIII. Related Work

There are plenty of recent works investigating the performance degradation of ML-based IDS caused by poisoning attacks. [10] investigated the influences of the poisoning attacks for PDF malware detectors using deep learning models. But the PDF files, same as image data, require the adversarial samples to have the same functionality as the normal data. The work in [11], [12] and [13] studied the impact of poisoning attacks on the network IDS with public dataset of network traffics, which are also time-series. However, all those IDS adopt deep learning models, which have different label flipping strategies from classic machine learning models. To the best of our knowledge, we are the first to evaluate the sensitivity of host-based IDS with the monitoring metric of syscall traces. In the research area of defending adversarial machine learning attacks, some works aim to make the training algorithms more resistant. [14] used nonlinear data projections and game theory to improve the resilience of SVMs against adversarial samples. The work in [15] proposed an adversarial SVM model, which can gain higher robustness with a modified loss function. However, these approaches sacrifice the performance of the classifiers and are not general. [16] proposed to reduce the influence of the adversarial samples by measuring the impact of each sample, but the method is very computationally expensive. [17] used KNN to filter out poisoned samples but it has limited effect for high dimensional data. [18] proposed to use weight initialization to remove the adversarial samples in the training set, but it assumes a small untainted dataset is available.

IX. Conclusions and Future Work

We evaluate and compare the performance degradation of OC-SVM caused by 3 different poisoning attacks: furthest first attack, nearest first attack and ALFA. From the attacker's perspective, the furthest first attack is optimal with a higher degradation level and lower computational effort. From the defender's perspective, we notice that even a small portion of injected malicious samples can deteriorate the classifier's performance dramatically.

We propose a sanitization process to filter out malicious samples before training. DBSCAN is used to separate clusters and label outliers according to the density. The sanitization process automatically removes all outliers if the output cluster number is 1. Our experimental results demonstrate the sanitization process is effective for all applications and poisoning strategies we considered. We compare the performance of 6

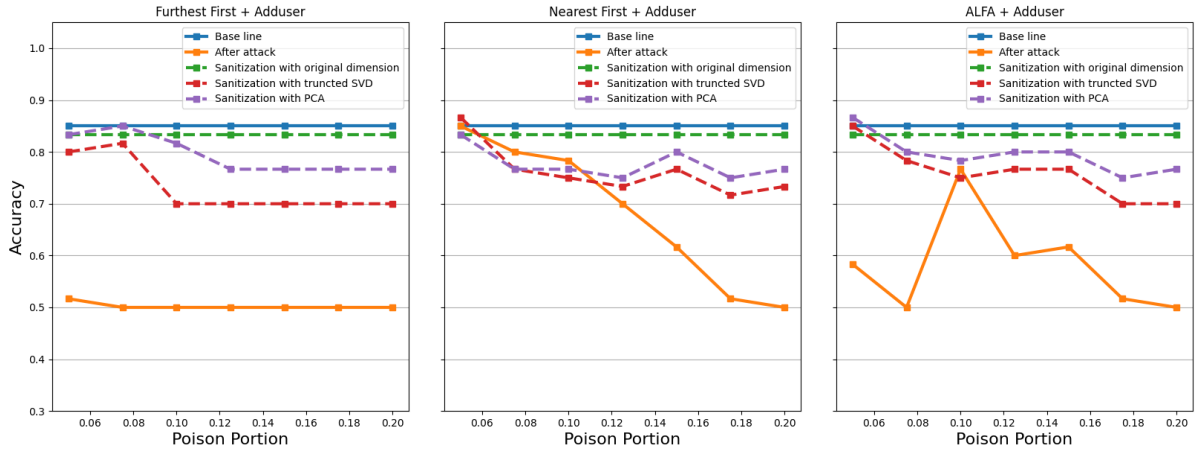

Fig. 8: Performance improvement with different dimensionality reduction techniques for the public dataset with Add user attack.

distance metrics and observe that the Euclidean distance is optimal since it fits best to the data patterns of the system call traces. Dimensionality reduction techniques do not contribute to an obvious performance improvement even with properly selected dimension numbers. DBSCAN parameters are sensitive to distance metrics but not applications and parameter reuse is therefore feasible.

In the future, we will try to increase the robustness of the OC-SVM algorithm itself and make it more resilient to noise and adversarial samples. In addition, we aim to further explore the influences and defense mechanisms of the evasion attacks, in which the adversary tries to fool the IDS classifier in the test phase by manipulating the observations.

Acknowledgments

This paper builds upon the work done within the Dutch NWO Research project 'Data Logistics for Logistics Data' (DL4LD, www.dl4ld.nl), supported by the Dutch Top consortia for Knowledge and Innovation 'Institute for Advanced Logistics' (TKI Dinalog, www.dinalog.nl) of the Ministry of Economy and Environment in The Netherlands and the Dutch Commit-to-Data initiative (https://commit2data.nl/).

References

[1] Z. Chiba, N. Abghour, K. Moussaid, M. Rida *et al.*, "Intelligent approach to build a deep neural network based ids for cloud environment using combination of machine learning algorithms," *computers & security*, vol. 86, pp. 291–317, 2019.

[2] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, 2018.

[3] L. Zhang, R. Cushing, C. de Laat, and P. Grosso, "A real-time intrusion detection system based on oc-svm for containerized applications," in *The 24th IEEE International Conference on Computational Science and Engineering*, 2021.

[4] B. Jin, Y. Chen, D. Li, K. Poolla, and A. Sangiovanni-Vincentelli, "A one-class support vector machine calibration method for time series change point detection," in *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2019, pp. 1–5.

[5] M. Xie, J. Hu, and J. Slay, "Evaluating host-based anomaly detection systems: Application of the one-class SVM algorithm to ADFA-LD," *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2014*, pp. 978–982, 2014.

[6] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[7] H. Xiao, H. Xiao, and C. Eckert, "Adversarial label flips attack on support vector machines," *Frontiers in Artificial Intelligence and Applications*, vol. 242, pp. 870–875, 2012.

[8] M. E. Houle, H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "Can shared-neighbor distances defeat the curse of dimensionality?" in *International conference on scientific and statistical database management*. Springer, 2010, pp. 482–500.

[9] B. Jin, Y. Chen, D. Li, K. Poolla, and A. Sangiovanni-Vincentelli, "A one-class support vector machine calibration method for time series change point detection," in *2019 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, 2019, pp. 1–5.

[10] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," *arXiv preprint arXiv:1606.04435*, 2016.

[11] C.-H. Huang, T.-H. Lee, L.-h. Chang, J.-R. Lin, and G. Horng, "Adversarial attacks on sdn-based deep learning ids system," in *International Conference on Mobile and Wireless Technology*. Springer, 2018, pp. 181–191.

[12] Z. Wang, "Deep learning-based intrusion detection with adversaries," *IEEE Access*, vol. 6, pp. 38 367–38 384, 2018.

[13] J. Clements, Y. Yang, A. Sharma, H. Hu, and Y. Lao, "Rallying adversarial techniques against deep learning for network security," *arXiv preprint arXiv:1903.11688*, 2019.

[14] S. Weerasinghe, S. M. Erfani, T. Alpcan, and C. Leckie, "Support vector machines resilient against training data integrity attacks," *Pattern Recognition*, vol. 96, p. 106985, 2019.

[15] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and B. Xi, "Adversarial support vector machine learning," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1059–1067.

[16] B. Nelson, M. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. Sutton, J. D. Tygar, and K. Xia, "Exploiting machine learning to subvert your spam filter." *LEET*, vol. 8, pp. 1–9, 2008.

[17] A. Paudice, L. Muñoz-González, and E. C. Lupu, "Label sanitization against label flipping poisoning attacks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2018, pp. 5–15.

[18] P. P. Chan, F. Luo, Z. Chen, Y. Shu, and D. S. Yeung, "Transfer learning based countermeasure against label flipping poisoning attack," *Information Sciences*, vol. 548, pp. 450–460, 2021.