Robust Anomaly Detection on Unreliable Data

Zhao, Zilong; Cerf, Sophie; Birke, Robert; Robu, Bogdan; Bouchenak, Sara; Ben Mokhtar, Sonia; Chen, Lydia Y.

DOI
10.1109/DSN.2019.00068

Publication date
2019

Document Version
Final published version

Published in
Proceedings - 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2019

2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)

Robust Anomaly Detection on Unreliable Data

Zilong Zhao*, Sophie Cerf*, Robert Birke†, Bogdan Robu*, Sara Bouchenak‡, Sonia Ben Mokhtar‡, Lydia Y. Chen§
*Univ. Grenoble Alpes, France
{zilong.zhao, sophie.cerf, bogdan.robu}@gipsa-lab.fr
†ABB Research, Switzerland
{robert.birke}@ch.abb.com
‡INSA Lyon, France
{sara.bouchenak, sonia.benmokhtar}@insa-lyon.fr
§TU Delft, Netherlands
{y.chen-10}@tudelft.nl

***Abstract*—Classification algorithms have been widely adopted to detect anomalies for various systems, e.g., IoT and cloud, under the common assumption that the data source is clean, i.e., features and labels are correctly set. However, data collected from the field can be unreliable due to careless annotations or malicious data transformation for incorrect anomaly detection. In this paper, we present a two-layer learning framework for robust anomaly detection (RAD) in the presence of unreliable anomaly labels.**

The first layer of quality model filters the suspicious data, where the second layer of classification model detects the anomaly types. We specifically focus on two use cases, (i) detecting 10 classes of IoT attacks and (ii) predicting 4 classes of task failures of big data jobs. Our evaluation results show that RAD can robustly improve the accuracy of anomaly detection, to reach up to 98% for IoT device attacks (i.e., +11%) and up to 83% for cloud task failures (i.e., +20%), under a significant percentage of altered anomaly labels.

***Index Terms*—Unreliable Data; Anomaly Detection; Failures; Attacks; Machine Learning**

I. Introduction

Anomaly detection is one of the core operations for enforcing dependability and performance in modern distributed systems [37]. Anomalies can take various forms including erroneous data produced by a corrupted IoT device or the failure of a job executed in a datacenter [6], [7].

Dealing with this issue has often been done in recent art by relying on machine learning-based classification algorithms over system logs [11], [13]. These systems often rely on a learning dataset from which the classifier learns to distinguish between data corresponding to a correct execution of the system from data corresponding to an abnormal execution of the latter (i.e., anomaly detection).

In this context, a rising concern when applying classification algorithms is the accessibility to a reliable ground truth for anomalies [9]. Typically, anomaly data is manually annotated by human experts and hence the generation of anomaly labels is subject to quality variation, so-called noisy labels. For instance, annotating service failure types for data centers is done by operators.

However, standard machine learning algorithms typically assume clean labels and overlook the risk of noisy labels. Moreover, recent studies point out the increasing dirty data attacks that can maliciously alter the anomaly labels to mislead the machine learning models [10], [14], [16]. As a result, anomaly detection algorithms need to capture not only anomalies that are entangled with system dynamics but also the unreliable nature of anomaly labels.

Indeed, a strong anomaly classification model can be learned by incorporating a larger amount of datasets; however learning from data with noisy labels can significantly degrade the classification accuracy, even for deep neural networks, at a non-negligible computation source [33]. Such a concern leads us to ask the following question: how to build an anomaly detection framework that can robustly differentiate the true and noisy anomalies and efficiently learn the anomaly classification models from a succinct amount of clean data. The immediate challenge of capturing the dynamics of data quality lies at the fact that label qualities are not directly observable but only via anomaly classification outcomes that in turn is coupled with the noise level of data labels.

In this paper, we develop a Robust Anomaly Detector (RAD), a generic framework that continuously learns the anomaly classification model from streams of event logs that are subject to label noises. To such an end, RAD is composed of two layers of learning models, i.e., data label model and anomaly classifier. The label model aims at differentiating the label quality, i.e., noisy v.s. true labels, for each batch of new data and only "clean" data points are fed in the anomaly classifier. The anomaly classifier predicts the event outcomes that can be in multiple classes of (non)anomalies, depending on the specific anomaly use case. The specific choices of label models and anomaly classifier include standard machine learning models, e.g., random forest, Adaboost, and discriminant analysis, and deep neural networks.

To demonstrate the effectiveness of RAD, we consider two use cases, i.e., detecting 10 classes of IoT attacks [21], and

[1] This work has been partly supported by the IRS (Initialtive de Recherche Stratégique) program DATE.
[2] This work has been partly funded by the Swiss National Science Foundation NRP75 project 407540_167266 and TU Delft technology fellowship.

978-1-7281-0057-9/19/$31.00 ©2019 IEEE
DOI 10.1109/DSN.2019.00068

predicting four types of task failures for big data processing cluster [27], [30] from open datasets. Our preliminary results show that RAD can effectively and continuously cleanse the data, i.e., selecting data streams with clean labels, and result better anomaly detection accuracy per additional data stream included, compared to classifiers without continuous data cleansing. Specifically, RAD achieves up to 98% and 83% accuracy for detecting IoT device attacks and predicting cluster task failures respectively.

The remainder of the paper is organized as follows. Section II describes the motivating case studies that we consider. Sections III and IV respectively present the proposed RAD framework and the results of its experimental evaluation. Section V describes the related work, and finally, Section VI draws our conclusions and the lessons learned.

II. Motivating case studies

To qualitatively demonstrate the impact of noisy data on anomaly detection, we use two case studies.

- Detecting **IoT device attacks** from inspecting network traffic data collected from commercial IoT devices [21]. This dataset contains nine types of IoT devices which are subject to ten types of attacks. Specifically, we focus on the Ecobee thermostat device that may be infected by Mirai malware and BASHLITE malware. Here we focus on the scenario of detecting and differentiating between ten attacks. It is important to detect those attacks with high accuracy against all load conditions and data quality.
- Predicting **task execution failures** for big data jobs running at Google cluster [27], [29]. This trace contains a month-long jobs execution record from Google clusters. Each job contains multiple tasks, which can be terminated into four different states: *finish*, *fail*, *evict*, or *kill*. The last three states are considered as anomaly states. To minimise the computational resource waste due to anomaly states, it is imperative to predict the final execution state of task upon their arrivals.

The details about data definition, and statistics, e.g., no. of feature and no. of data points, can be found in Section IV-A. To detect anomalies in each case, related studies have applied machine learning classification algorithms, e.g., k-nearest neighbor (KNN), nearest centroid and multilayer perceptron (MLP) (a.k.a feed-forward deep neural networks), under the scenario where different levels of label noise are present. Here, we evaluate how the detection accuracy changes relative to different levels of noises. We focus on offline scenarios where classification models are learned from 14000 records and evaluated on a clean testing dataset of 6000 records. We specifically apply KNN, nearest centroid and MLP on IoT device attacks and cluster task failures respectively, and summarize the accuracy results in Figure 1a and Figure 1b.

One can see that noisy labels clearly deteriorate the detection results for both IoT attacks and task failures, across all three classification algorithms. For standard classifiers, like KNN and nearest centroid, the detection accuracy decays faster than MLP that is more robust to the noisy labels. Such an observation holds for both uses cases. In IoT attacks, MLP can even achieve a similar accuracy as the case of no label noises, when there is 50 percent of label classes are altered.

.

III. Design Principles of RAD Framework

A. System Model

We consider a dataset that consists of several data instances. Each data instance has f features. Each data instance belongs to a class k, where $k \in \mathcal{K} = \{1, \ldots, K\}$. A data instance is either labeled with a class k or is not labeled. Furthermore, a labeled data instance is either correctly labeled (i.e., clean data instance), or incorrectly labeled (i.e., noisy data instance). In the latter case, the data is annotated with a wrong label due to human error, or malicious error injection, etc. Obviously, a non-labeled data instance is clean. The quality of a dataset $\mathcal{D}$ is measured as the percentage of data instances with noisy labels, denoted here as $\tilde{Y}$.

Data instances structured in batches are assumed to arrive continuously in the learning system over time. $\mathcal{D}_i$ denotes the set of data instances arriving in batch at time t_i, and its set of labels $\tilde{Y}_i$, for which the quality may vary from one data instance to another inside the data batch. We assume that $\mathcal{D}_0$, a small set of initial data instances with both clean and noisy labels is given. Other collected data instances are also noisy, and the quality of datasets from batches to batches may fluctuate. Data instances belonging to $\mathcal{D}_i$ are denoted $d_{1,i}, d_{2,i}, \ldots d_{N,i}$. Up to N new data instances are considered for training the data classifier and building the learned data model. We consider that the batches arrive with constant number of data instances, $\forall \mathcal{D}_i, |\mathcal{D}_i| = N$, but not necessary at regular time intervals.

Then, a classification request consists of sets of non-labeled data instance $\mathcal{P}_i$ for which the classifier predicts the class k to which each data instance belongs. At each batch, the classification output is thus an array of the predicted classes for each data instance in the batch $\tilde{Y}_i^{\mathcal{P}}$.

B. Objectives and Overview of RAD Framework

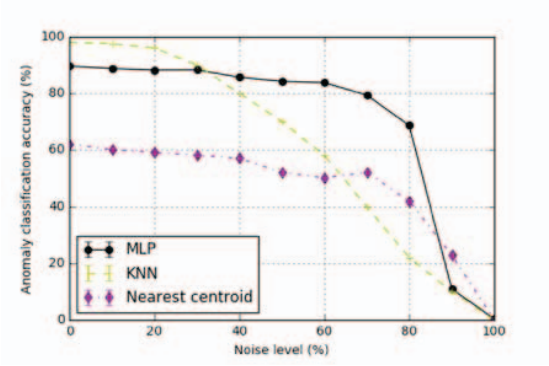
We propose the RAD learning framework. Its objective is threefold:

(i) Accurately learning a data model from noisy data.
(ii) Continuously updating the learned model with new incoming data.
(iii) Proposing a general approach that applies on different machine learning algorithms, and with different application use cases.

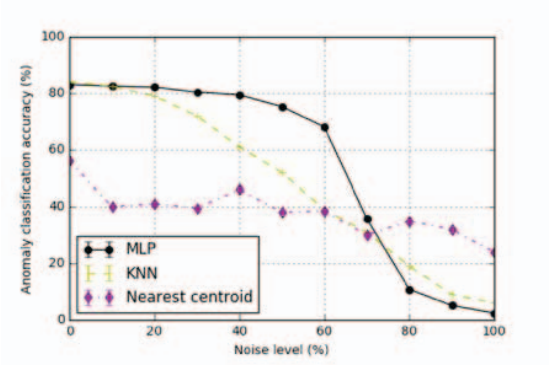
Figure 2 describes the overall architecture of RAD. Each of its components is detailed in the following.

C. Determining Data Noise

The first component of RAD aims at determining if a labeled data instance is correctly or incorrectly labeled. RAD learns the data label quality model $\mathfrak{L} : \mathbb{R}^f \rightarrow q \in \mathcal{Q} = \{0, 1\}$. The objective of the label quality model is to select the most representative data instances to learn a strong data



(a) Use case of IoT thermostat device attacks



(b) Use case of Cluster task failures

Fig. 1: Impact of noisy data on anomaly classification

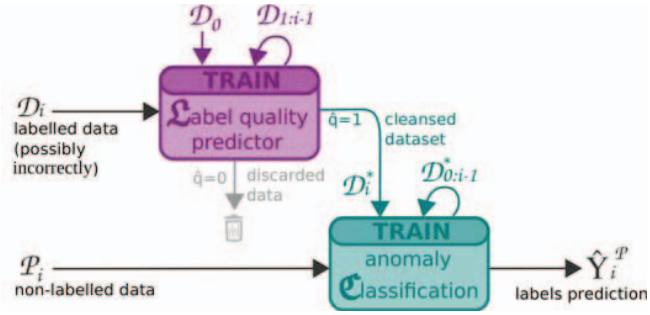


Fig. 2: RAD learning framework. The flowchart is iterated at every batch, with the new data coming in (left arrows). Each block is a machine learning algorithm, data used to train them are represented by colored arrows coming from the top. The data selected based on the label quality model's predictions provides the training data for the main task, the anomaly classification. Global output (right arrow) is the predicted labels of each new batch $\mathcal{P}_i$.

classifier model. It solicits data instances with clean labels, avoiding the pitfall that the classifier overfits the noise. RAD uses supervised-learning algorithms to continuously train the label quality model from accumulated predicted clean data instances, which are highly correlated to a stronger classifier.

The label quality model tries to learn the binary classifier for label quality, i.e., $q = 1$ denotes a correct label and $q = 0$ denotes an incorrect label. $\mathfrak{L}_i$ is the label quality model that is trained with data instances received up to time t_i, that is $\mathcal{D}_0, \mathcal{D}_1 \ldots \mathcal{D}_i$. Upon the arrival of a new batch of data instances $\mathcal{D}_i$ at time t_i, we use the previously learned label quality model $\mathfrak{L}_{i-1}$ to predict the quality for each data instance $d_{j,i} \in \mathcal{D}_i$, this quality is denoted $\hat{q}_{j,i}$. If $\hat{q}_{j,i} = 0$, meaning an incorrect label, we discard such a data instance and only consider data instances with $\hat{q}_{j,i} = 1$. We thus build ${\mathcal{D}^*}_i$, the subset of $\mathcal{D}_i$ with all correct data instances from $\mathcal{D}_i$. And we incorporate ${\mathcal{D}^*}_i$ into the existing training set for the data classifier. In turn, we incorporate the clean data ${\mathcal{D}^*}_i$ to retrain the label model for the next batch, that is $\mathfrak{L}_i$.

D. Generic Approach to Handle Dynamic Data

Another component of RAD is the dynamic data classifier, $\mathfrak{C} : \mathbb{R}^f \rightarrow k \in \mathcal{K}$. Essentially, $\mathfrak{C}_i$ is trained on all the estimated clean data received until batch $\mathcal{D}_i$, that is ${\mathcal{D}^*}_0 \ldots {\mathcal{D}^*}_i$. Indeed, the received data $\mathcal{D}_1 \ldots \mathcal{D}_i$ were cleaned using the data label quality model $\mathfrak{L}_0 \ldots \mathfrak{L}_{i-1}$ to produce clean data ${\mathcal{D}^*}_1 \ldots {\mathcal{D}^*}_i$. Thus, the RAD learning framework uses the data label quality model which is updated from batch to batch, and enriches its overall learned data classification model accordingly.

Furthermore, RAD follows a generic approach since the proposed classification framework can be used with different machine learning algorithms, such as SVM, KNN, random forest, nearest centroid classifier, etc. And RAD has different applications where noisy data are collected and must be cleaned before learning data model, such as failure detection and attack diagnosis that we illustrate in Section IV.

IV. Experimental Evaluation

A. Use Cases and Datasets

In order to demonstrate the general applicability of the proposed RAD framework for anomaly detection, we consider the following two use cases: (i) Cluster task failures, and (ii) IoT botnet attacks. In our experiments, we use real data collected in cluster and IoT platforms.

The task traces comprise data instances each corresponding to a task with 27 features capturing information related to static and dynamic system state, e.g. the task start/end times, the task resource utilisations, the hosting machine, etc. Each class is labeled based on its scheduling state. A detailed description of the features and labels can be found in [27]. In particular, we are interested in the four possible termination classes: *finish*, *fail*, *evict*, or *kill*. We filter out other classes. The resulting class distribution is dominated by successful tasks (*finish*) 77.8%, followed by *kill* 22.0%, *fail* 0.2%, and *evict* <0.1%. Similar

to [29], we aim to predict the task outcome to reduce the resource waste and improve the overall scheduling and system performance, e.g., in case of lack of resources and need to kill a task help choosing the task with the least probability to succeed. We apply RAD to continually train a noise-resistant model for better accuracy.

The IoT dataset comprises data instances describing 23 network packet-level statistics recursively computed over five different time scales totalling to 115 features. This traffic statistics are collected during normal operation, labeled as benign, or under one of ten different malicious attacks stemming from devices infected by either the *BASHLITE* or *Mirai* malware. Malicious traffic covers mainly scanning for vulnerable devices and various flooding attacks. The dataset provides traces collected at different IoT devices. More details are provided in [21]. We aim to apply RAD to build a noise-resistant model to categorize the attacks for post fact analysis, e.g., for threat assessment.

The main dataset characteristics are summarized in Table I.

TABLE I: Dataset description

Use case	Cluster task failures	IoT device attacks
#trainig data instances	57,000	33,000
#test data instances	6,000	6,000
#classes K	4	11
#features f	27	115
data batch size	600	300

B. Experimental Setup

RAD is developed in Python using scikit-learn [24]. The main performance evaluation metric is accuracy. Experiments are carried out 5 times, results are aggregated by computing mean and standard deviation.

Noise. We inject noise into the two datasets by exchanging the true label of data instances with a random one. The label noise is symmetric, i.e., following the noise completely at random model [12] where a label is picked with equal probability from all classes except the true one. The noise level represents the percentage of data instances with noisy labels. We emulate time-varying noise by drawing for each new data batch the noise level from a Gaussian distribution with 20% standard deviation and various mean levels. We assume that all data is affected by label noise, except the testing data.

Continual learning. We start with an initial data batch of 3000 data instances for the Cluster task failures dataset, 6000 for the IoT devices one. Then, data instances arrive continuously in batches of respectively 600 and 300 data instances. Both the initial and subsequent data batches are affected by noise. To kick-start the label and classification models in RAD we assume to know which initial data instances are affected by noise (no assumptions for the subsequent data batches). We select 6000 clean data instances as the test dataset for both use case. Test dataset will be used at the end of each epoch to evaluate classification model's accuracy. We show the evolution of the model accuracy over data batch arrivals until the performance of RAD converges.

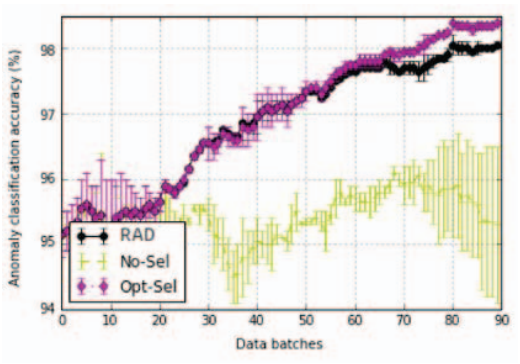
Label model. We use a multilayer perceptron to assess the quality of each label. For IoT dataset, the neural network consists of two layers with 28 neurons each. For Cluster task dataset, the network consists of two layers with 110 neurons each. The precision and robustness of the label model are critical to filter out the noisy/malicious labels and provide a clean training set to the classification model. We considered different models, but neural networks provided the best results in terms of accuracy and stability over time. Adaboost gave excellent accuracy when training from the initial data with ground truth, but resulted too sensitive to the unknown noise of subsequent data batches. Random forest is another choice and known to be robust against label noise [12], however its accuracy was below the neural network one.

Classification model. We use KNN to assign the correct class label to each data instance filtered by the label model. We set k to 5. Higher values of k can increase the resilience of the algorithm to residual noise, but also induce extra computational cost. The current choice stems from good results in preliminary experiments.

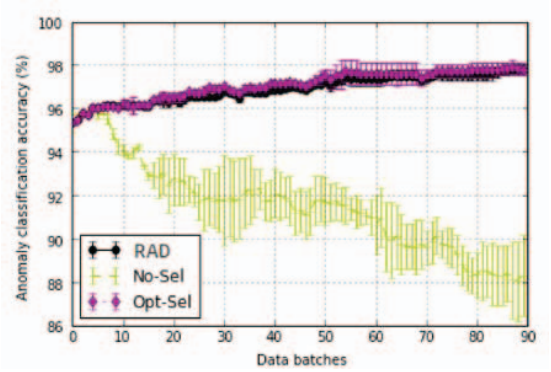
Baselines. The proposed RAD is compared against two baseline data selection schemes: (i) No-Sel, where all data instances of arriving batches are used for training the classification model; and, (ii) Opt-Sel which emulates an omniscient agent who can perfectly distinguish between clean and noisy labels. The two baselines are representative of the worst and best possible data selection strategies and we expect RAD to fall in between.

C. Handling Dynamic Data

We start by illustrating how RAD enables to increase the anomaly detection accuracy over time, despite the presence of noise. Figures 3 and 4 show the evolution of the mean and variance of the classification accuracy achieved by RAD on the thermostat and task failure datasets, respectively. Each figure moreover presents results under two levels of label noise: 30% and 40%. We compare RAD against no selection (No-Sel) and optimal selection (Opt-Sel). One can notice that learning from all data instances without cleansing (i.e., No-Sel curves) gives consistently lower accuracy in all cases. For the attacks classification on the thermostat, the accuracy even oscillates and diverges. The performance when using RAD is better. First because the accuracy does not diverge, second because it always consistently increase until it saturates. For the IoT attack dataset, the end accuracy is around 98%, for the cluster tasks around 83%. While for the first dataset the accuracy of RAD follows closely the accuracy of Opt-Sel, for the second dataset RAD follows Opt-Sel at first but then saturates after 40 data batch arrivals. RAD is efficient for various classification applications, however not optimal for all of them. Note that RAD gives also more stable results as shown by shorter errorbars which in magnitude are in line with the ones obtained by an ideal data cleansing. For No-

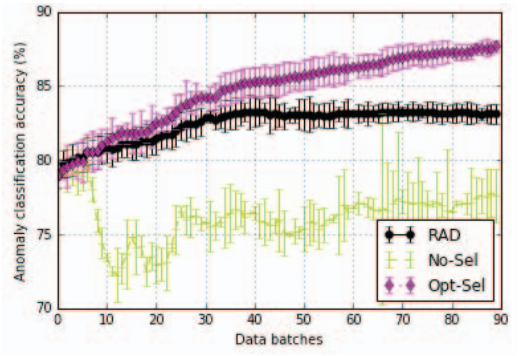


(a) With data noise level of 30%

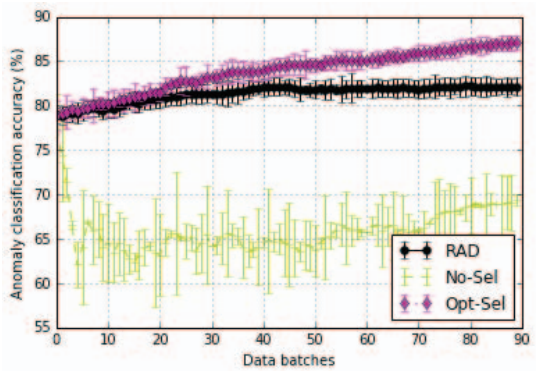


(b) With data noise level of 40%

Fig. 3: Evolution of learning over time – Use case of IoT thermostat device attacks



(a) With data noise level of 30%



(b) With data noise level of 40%

Fig. 4: Evolution of learning over time – Use case of Cluster task failures

Sel the bars are significantly larger. We discuss the results for other IoT devices in Section IV-F.

Figure 5 presents the variations of noise over time for one run on IoT thermostat dataset where mean noise rate sets to 30%. Overlaid is the number of data selected by RAD and the overlap between selection and actually clean. Results highlight the sharpness of data selection and its parsimony.

In summary: (i) continual learning is advantageous compared to using only the initial dataset; however, (ii) continual learning exposes us to possible classification accuracy degradation stemming from noisy labels if proper data selection is lacking, (iii) RAD improves the classification accuracy compared to taking all labels, (iv) the data selection of RAD is good, and close to being optimal in some cases.

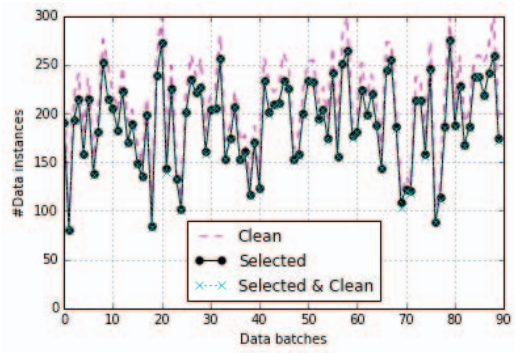


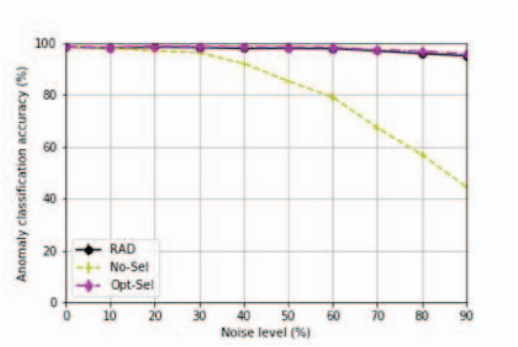
Fig. 5: Data selection – Use case of IoT thermostat device attacks with 30% noise level.

D. Evaluation of Noise Robustness of RAD

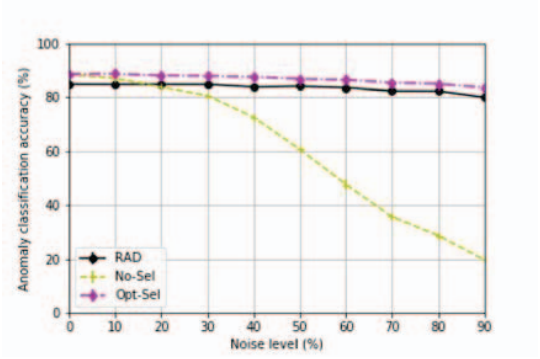
Next we investigate the impact of different noise levels on the RAD performance in terms of classification accuracy.

Figures 6a and 6b present the classification accuracy for various levels of noise, ranging from 0% (all data are clean) up to 90% for our two main reference datasets: IoT thermostat device attacks and Cluster task failures. Accuracy is measured once the learning has converged. Once again, the RAD performance is compared to learning from all data (No-Sel) and an omniscient data cleanser (Opt-Sel).

As illustrated in Section II, for No-Sel the noisier the data are, the worst the classification accuracy, dropping to 20%

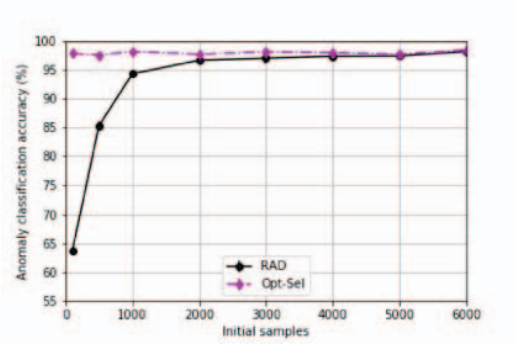


(a) IoT thermostat device attacks

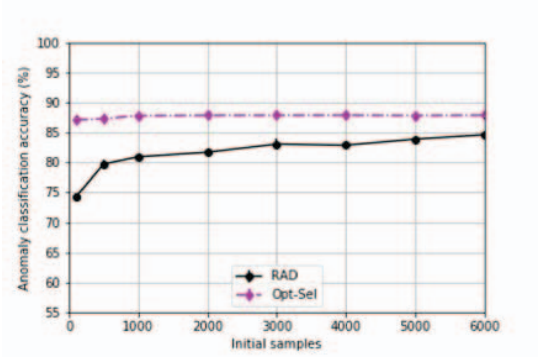


(b) Cluster task failures

Fig. 6: Impact of data noises on RAD accuracy



(a) IoT thermostat device attacks



(b) Cluster task failures

Fig. 7: Impact of size of initial data batch $\mathcal{D}_0$ on RAD accuracy

and 42% for the tasks and thermostat datasets, respectively. A decreasing trend can also be found for RAD and Opt-Sel, however the drops are significantly smaller: at most 5%. As there is by definition no noise in Opt-Sel case, the decrease in classification accuracy is only due to the reduction of the overall amount of clean data to learn from. Since the data cleansing of RAD is not perfect, the accuracy reduction is caused by both pollution from the noise and the overall clean data reduction. Nevertheless, the impact is small and any huge accuracy pitfall is avoided which results in RAD's performance being close to Opt-Sel. We can conclude that RAD can limit the impact of the amount of noise across a wide range of noise levels.

E. Impact of Initialization

Here we study the impact on RAD of the amount of ground truth data available, i.e., the size of the initial dataset $\mathcal{D}_0$.

In Figure 7a, we vary the number of initial clean data instances for the IoT thermostat device dataset from 100 to 6000, and measure the classification accuracy after 90 data batch arrivals, for both RAD and Opt-Sel. We do not consider No-Sel here since this results are meant for the framework configuration, not its performance evaluation. Similarly, Figure 7b describes the results of the same experiments with the Cluster task failure dataset.

In Figure 7a for Opt-Sel, no matter the number of data instances in the initial set, the final accuracy is stable around 98%. However for RAD, the size of $\mathcal{D}_0$ does matter: the larger the better, with some saturation aftereffect. At $|\mathcal{D}_0| = 3000$ its performances is close to Opt-Sel, and at $|\mathcal{D}_0| = 6000$ they completely overlap.

This justifies our earlier choice of 6000 data instances in our initial set for the IoT device attacks classification as it enables to achieve the best accuracy. However, RAD framework could also perform well with only half of those data as initial set.

F. Analysis of All Datasets

Summary results across datasets are reported in Table II. In addition to the average accuracy after the last batch arrival, we also underline the relative accuracy improvement obtained by comparing RAD to the initial set, i.e., the impact of continual learning, and to the No-Sel strategy, i.e., the impact of intelligent data selection. Columns 5 and 6 report these results. Finally, we present the percentage of accuracy

difference between RAD and Opt-Sel, i.e., how close we are to an omniscient data selection, in column 7.

All results are positive, with varying magnitude depending on the dataset. In all cases, the proposed RAD improves between 1% to 5% the accuracy compared to blindly taking all data instances. However, more important than the absolute gain is the trend. For example, for the thermostat dataset, we can observe that RAD converges over time to a stable level as well as Opt-Sel model, but No-Sel diverges. This means that as time goes by, No-Sel becomes worse and worse.

Even more than the benefits of continual learning might be important the resilience to high levels of noise. Under such levels, the classification accuracy without data cleansing diverges for all datasets. Even if it is rare to have noise levels of 90% or above, they might still happen for short periods of time in case of attacks to the auto-labelling system via flooding of malicious labels. Hence this property can be crucial for the dependability of the auto-labelling system.

G. Limitation of RAD Framework

Though the RAD framework works well for datasets of Cluster task failures and IoT device attacks. We can still see the potential limitations of this model, for example: (1) the assumption of availability of a small fraction of clean data which may not be possible; (2) if data is coming at high rates, or the structure of quality model becomes more complicated, training and predicting time of first layer will slow down the system.

V. Related Work

Machine learning has been extensively used for failure detection [8], [25], [26], [28], and for attack prediction [1], [3], [4], [17], [18], [38]. Considering noisy labels in classification algorithms is also a problem that has been explored in the machine learning community as discussed in [5], [12], [22].

The problem of classification in presence of noisy labels can be organized into various categories according to, on the one hand, the type of classification algorithm subject to noise, and on the other hand, the techniques used to remove the noise.

Regarding the type of classification algorithm, the problem of noisy labels has been studied both for binary classification where noisy labels are considered as symmetric (e.g., [19]) and for classification with multiple classes where noisy labels are considered as asymmetric, e.g., [23], [31]. In the context of this paper, we consider the problem of classification with multiple classes. Furthermore, noisy labels have been considered in various types of classifiers KNN [36], SVM [2], and deep neural networks [34]. In the context of this paper, our proposed approach is agnostic to the underlying classifier type as noise removal is performed ahead of the classification.

To deal with noise, various techniques have been explored including forward loss correction. These algorithms learn about the label noise by adjusting the loss to the end of the model. However, these solutions either rely on strong assumptions or have limited accuracy as they generally do not rely on clean labels to remove the noise. More accurate solutions, which rely on clean labels during the training phase have thus been explored (e.g., [15], [20], [35]). These solutions generally train a separate network for distinguishing noisy labels from clean ones. Robustness to label noise has also been studied for GANs performing image recognition, both in the context of known and unknown noise distribution [32]. However, all these solutions have been designed and tested on static datasets and in an off-line setting. Instead in the context of this paper, we consider a dynamic model where the network has been trained using clean labels continues to learn over time.

VI. Concluding Remarks

While machine learning classification algorithms are widely applied to detect anomalies, the commonly employed assumption of clean anomaly labels often does not hold for the data collected in the wild due to careless annotation and malicious dirty label pollution. The noisy labels can significantly degrade the accuracy of anomaly detection with an increasing amount of data and are challenging to tackle due to the lack of ground truth of label quality. In this paper, we present a framework for robust anomaly detection, RAD, which can continuously learn the system dynamics and anomaly behaviours from streams of arriving data after filtering out suspicious noisy data.

RAD is a general framework that composes of sequence of quality and classification models, where the former captures the label dynamics and the latter focus on detection anomaly. We demonstrate the effectiveness of RAD on two uses cases, i.e., detecting IoT device attacks, and predicting task failure at Google clusters. RAD can robustly improve the detection accuracy against different levels of label noises, reaching up to 83% and 98% accuracy for predicting task failure and detecting IoT device attacks, respectively, whereas learning directly from all the data streams without filtering degrades the detection accuracy.

References

[1] M. Agarwal, D. Pasumarthi, S. Biswas, and S. Nandi. Machine learning approach for detection of flooding dos attacks in 802.11 networks and attacker localization. *Int. J. Machine Learning & Cybernetics*, 7(6):1035–1051, 2016.

[2] W. An and M. Liang. Fuzzy support vector machine based on within-class scatter for classification problems with outliers or noises. *Neurocomput.*, 110:101–110, June 2013.

[3] M. Anbar, R. Abdullah, B. N. Al-Tamimi, and A. Hussain. A machine learning approach to detect router advertisement flooding attacks in next-generation ipv6 networks. *Cognitive Computation*, 10(2):201–214, 2018.

[4] S. Banescu, C. S. Collberg, and A. Pretschner. Predicting the resilience of obfuscated code against symbolic execution attacks via machine learning. In E. Kirda and T. Ristenpart, editors, *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017.*, pages 661–678. USENIX Association, 2017.

[5] B. Biggio, B. Nelson, and P. Laskov. Support vector machines under adversarial label noise. In *Asian Conference on Machine Learning*, pages 97–112, 2011.

[6] R. Birke, G. Ioana, Lydia Y. Chen, D. Wiesmann, and T. Engbersen. Failure analysis of virtual and physical machines: patterns, causes and characteristics. In *IEEE/IFIP DSN*, pages 1–12, 2014.

[7] R. Birke, A. Podzimek, Lydia Y. Chen, and E. Smirni. Virtualization in the private cloud: State of the practice. *IEEE Trans. Network and Service Management*, 13(3):608–621, 2016.

TABLE II: Evaluation of the RAD learning framework for all datasets.

Dataset	Initial accuracy	Final accuracy			Improvement due to		Distance to optimal
		No-Sel	RAD	Opt-Sel	Continual learning	Selection	
Cluster task failures	78.98%	78.06%	83.03%	88.06%	4.05%	4.97%	5.03%
IoT monitor device attacks	97.60%	96.76%	99.33%	99.50%	1.73%	2.57%	0.17%
IoT doorbell device attacks	96.00%	96.50%	98.95%	99.05%	2.95%	2.45%	0.10%
IoT thermostat device attacks	95.15%	95.30%	98.05%	98.40%	2.90%	2.75%	0.35%

[8] J. R. Campos, M. Vieira, and E. Costa. Exploratory study of machine learning techniques for supporting failure prediction. In *14th European Dependable Computing Conference, EDCC 2018, Iaşi, Romania, September 10-14, 2018*, pages 9–16. IEEE Computer Society, 2018.

[9] S. Cerf, R. Birke, and Lydia Y. Chen. Duo learning for classifications with noisy labels. In *Continual Learning Workshop, in conjunction with Neural Information Processing Systems (NIPS)*, 2018.

[10] Y. Fan, J. Li, and D. Zhang. A method for identifying critical elements of a cyber-physical system under data attack. *IEEE Access*, 6:16972–16984, 2018.

[11] Z. Fang, J. Tzeng, C. C. Chen, and T. Chou. A study of machine learning models in epidemic surveillance: Using the query logs of search engines. In *Pacific Asia Conference on Information Systems, PACIS 2010, Taipei, Taiwan, 9-12 July 2010*, page 137. AISeL, 2010.

[12] B. Frénay and M. Verleysen. Classification in the presence of label noise: A survey. *IEEE Trans. Neural Netw. Learning Syst.*, 25(5):845–869, 2014.

[13] G. Giantamidis and S. Tripakis. Learning moore machines from input-output traces. In J. S. Fitzgerald, C. L. Heitmeyer, S. Gnesi, and A. Philippou, editors, *FM 2016: Formal Methods - 21st International Symposium, Limassol, Cyprus, November 9-11, 2016, Proceedings*, volume 9995 of *Lecture Notes in Computer Science*, pages 291–309, 2016.

[14] Y. He, G. J. Mendis, and J. Wei. Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism. *IEEE Trans. Smart Grid*, 8(5):2505–2516, 2017.

[15] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.*

[16] J. Kang, I. Joo, and D. Choi. False data injection attacks on contingency analysis: Attack strategies and impact assessment. *IEEE Access*, 6:8841–8851, 2018.

[17] D. Karagiannis and A. Argyriou. Jamming attack detection in a pair of RF communicating vehicles using unsupervised machine learning. *Vehicular Communications*, 13:56–63, 2018.

[18] R. Kozik, M. Choras, M. Ficco, and F. Palmieri. A scalable distributed machine learning approach for attack detection in edge computing environments. *J. Parallel Distrib. Comput.*, 119:18–26, 2018.

[19] J. Larsen, L. Nonboe, M. Hintz-Madsen, and L. K. Hansen. Design of robust neural network classifiers. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 1205–1208. IEEE, 1998.

[20] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L.-J. Li. Learning from noisy labels with distillation. In *ICCV*, pages 1928–1936, 2017.

[21] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici. N-baiotnetwork-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, 2018.

[22] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013.

[23] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, pages 2233–2241, 2017.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[25] A. Pellegrini, P. di Sanzo, and D. R. Avresky. A machine learning-based framework for building application failure prediction models. In *2015 IEEE International Parallel and Distributed Processing Symposium Workshop, IPDPS 2015, Hyderabad, India, May 25-29, 2015*, pages 1072–1081. IEEE Computer Society, 2015.

[26] T. Pitakrat, A. van Hoorn, and L. Grunske. A comparison of machine learning algorithms for proactive hard disk drive failure detection. In P. Kruchten and S. Malek, editors, *Proceedings of the 4th international ACM Sigsoft symposium on Architecting critical systems, ISARCS 2013, Vancouver, BC, Canada, June 17-21, 2013*, pages 1–10. ACM, 2013.

[27] C. Reiss, J. Wilkes, and J. L. Hellerstein. Google cluster-usage traces: format+ schema. *Google Inc., White Paper*, pages 1–14, 2011.

[28] U. Reuter, A. Sultan, and D. S. Reischl. A comparative study of machine learning approaches for modeling concrete failure surfaces. *Advances in Engineering Software*, 116:67–79, 2018.

[29] A. Rosà, L. Y. Chen, and W. Binder. Failure analysis and prediction for big-data systems. *IEEE Trans. Services Computing*, 10(6):984–998, 2017.

[30] A. Rosà, Lydia Y. Chen, and W. Binder. Understanding the dark side of big data clusters: An analysis beyond failures. In *IEEE/IFIP DSN*, pages 207–218, 2015.

[31] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014.

[32] K. K. Thekumparampil, A. Khetan, Z. Lin, and S. Oh. Robustness of conditional gans to noisy labels. In *Advances in Neural Information Processing Systems*, pages 10291–10302, 2018.

[33] V. N. Vagin and M. V. Fomina. Problem of knowledge discovery in noisy databases. *Int. J. Machine Learning & Cybernetics*, 2(3):135–145, 2011.

[34] A. Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *NIPS*, pages 5601–5610, 2017.

[35] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. J. Belongie. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pages 6575–6583, 2017.

[36] D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Mach. Learn.*, 38(3):257–286, Mar. 2000.

[37] J. Xue, R. Birke, Lydia Y. Chen, and E. Smirni. Spatial-temporal prediction models for active ticket managing in data centers. *IEEE Trans. Network and Service Management*, 15(1):39–52, 2018.

[38] B. Zhou, J. Li, J. Wu, S. Guo, Y. Gu, and Z. Li. Machine-learning-based online distributed denial-of-service attack detection using spark streaming. In *2018 IEEE International Conference on Communications, ICC 2018, Kansas City, MO, USA, May 20-24, 2018*, pages 1–6. IEEE, 2018.