ELITE: Robust Deep Anomaly Detection with Meta Gradient

Huayi Zhang
WPI, Worcester, MA, USA
hzhang4@wpi.edu

Lei Cao*
MIT, Cambridge, MA, USA
lcao@csail.mit.edu

Peter VanNostrand
WPI, Worcester, MA, USA
pvannostrand@wpi.edu

Samuel Madden
MIT, Cambridge, MA, USA
madden@csail.mit.edu

Elke A Rundensteiner
WPI, Worcester, MA, USA
rundenst@cs.wpi.edu

ABSTRACT

Deep Learning techniques have been widely used in detecting anomalies from complex data. Most of these techniques are either unsupervised or semi-supervised because of a lack of a large number of labeled anomalies. However, they typically rely on a clean training data not polluted by anomalies to learn the distribution of the normal data. Otherwise, the learned distribution tends to be distorted and hence ineffective in distinguishing between normal and abnormal data. To solve this problem, we propose a novel approach called ELITE that uses a small number of labeled examples to infer the anomalies hidden in the training samples. It then turns these anomalies into useful signals that help to better detect anomalies from user data. Unlike the classical semi-supervised classification strategy which uses labeled examples as training data, ELITE uses them as validation set. It leverages the gradient of the validation loss to predict if one training sample is abnormal. The intuition is that correctly identifying the hidden anomalies could produce a better deep anomaly model with reduced validation loss. Our experiments on public benchmark datasets show that ELITE achieves up to 30% improvement in ROC AUC comparing to the state-of-the-art, yet robust to polluted training data.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

Anomaly Detection; Polluted Training Data; Validation Loss

ACM Reference Format:
Huayi Zhang, Lei Cao, Peter VanNostrand, Samuel Madden, and Elke A Rundensteiner. 2021. ELITE: Robust Deep Anomaly Detection with Meta Gradient. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467320

1 INTRODUCTION

Motivation. In recent years deep neural networks have been widely used to detect anomalies from complex data sources, such as imagery and time series. Because real applications typically do not have a large number of labeled anomalies available beforehand, most deep anomaly detection techniques are either unsupervised [11, 14, 20] that do not use any labels, or semi-supervised [11, 20, 31] that uses a small set of normal or abnormal examples to improve the accuracy of unsupervised deep anomaly techniques.

The Limitations of State-of-the-art. However, these deep anomaly methods, either unsupervised or semi-supervised, require that the unlabeled training data be clean – not contaminated by any anomalies, so that they can learn a data representation that captures the distribution of the normal data. Were the training data to be contaminated by anomalies, the representation learned by these deep models could encode information about anomalous samples as part of the distribution of normal data. In this case, there is no guarantee that these models can properly distinguish between normal and anomalous samples. However, in real applications such a clean training data set rarely exists. Although the semi-supervised deep anomaly methods improve the quality of unsupervised anomaly detection by leveraging the classical semi-supervised classification strategy, they still suffer from the polluted training data. As shown in our experiments (Sec. 5.2), their performance degrades quickly when the number of the anomalies in the training data increases.

Proposed Approach. In this work, we propose an approach, called ELITE that leverages the labeled examples to solve the problem caused by polluted training data.

Unlike the semi-supervised classification strategy that uses labeled examples as training data, ELITE uses them as *validation data.* The core methodology of ELITE is to infer the labels of the polluted training data samples as normal or anomalous according to their potential influence on the model's validation loss. ELITE is based on a basic hypothesis: the correct labels of the unlabeled training samples should reduce the validation loss on the labeled examples. Thus ELITE uses a strategy that continuously discovers the anomalies in the polluted training data and learns a better deep anomaly model based on the corrected labels.

Moreover, using a tailored loss function that copes with normal and anomalous samples differently, ELITE trains the model to *maximize* the anomalous score for unlabeled samples that are likely anomalies while *minimizing* this score for unlabeled samples that are likely normal. In this way, ELITE not only uses the information from labeled examples, but also effectively turns the anomalies in

[*] Corresponding author

KDD '21, August 14–18, 2021, Virtual Event, Singapore.

ACM ISBN 978-1-4503-8332-5/21/08.
https://doi.org/10.1145/3447548.3467320

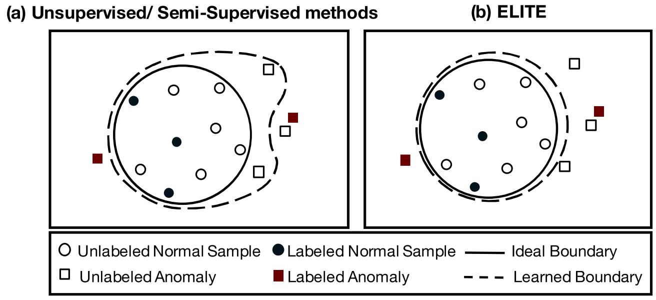


Figure 1: ELITE: Robust to Polluted Training Data. Leveraging the labeled examples, ELITE turns the hidden anomalies into useful signals that help to learn a better classification boundary.

the training data into *useful signals* that help to produce a data representation inherently anomaly-aware.

Clearly, the key of ELITE is how to efficiently identify the optimal labels for the unlabeled samples that minimize the model's validation loss. Finding optimal labels by repeatedly flipping the label of each sample and re-training the model to compute the validation loss will be too expensive. To solve this problem, ELITE proposes an efficient label inference method, called ALICE. ALICE introduces the concept of *meta-gradient* to directly estimate the potential change of the validation loss caused by altering the label of any training sample, without having to indeed re-train the model. ELITE then fuses ALICE into every iteration during the training process to dynamically adjust the labels of the training samples in a way that is guaranteed to *monotonically* reduce the validation loss.

ELITE is general in that different categories of unsupervised deep anomaly techniques can seamlessly plug their objective functions into ELITE and benefit from the labeled examples, such as Auto-Encoder-based methods [3, 6, 13, 22, 30] and Deep One Class Classification-based methods [19, 23], as discussed in Sec. 4.4 and confirmed by our experiments (Sec. 5).

Contributions Our key technical contributions include:

- We propose ELITE, an approach that uses a small set of labeled examples to solve the problem caused by polluted training data.
- Unlike existing semi-supervised classification techniques, ELITE adopts a new optimization paradigm that uses the labeled examples as validation set to infer the labels of the polluted training data.
- We propose ALICE that directly infers the labels of the training data based on the gradient of the validation loss, without having to re-training the deep learning model.
- Our experimental study on several benchmark datasets confirms that ELITE consistently outperforms the state-of-the-art semi-supervised deep anomaly methods and the unsupervised robust deep anomaly methods by 30% in ROC AUC score. Further, it is robust to polluted training data: the more anomalies in the data, the more it outperforms the alternatives.

2 RELATED WORK

Unsupervised Deep Anomaly Detection. Unsupervised deep anomaly techniques in general can be characterized into two categories. The first category learns a representation that better distinguishes anomalies from normal data. Some of these techniques [3, 6, 13, 22, 30] use the reconstruction errors of Auto-Encoder as the anomalous score to directly detect anomalies, assuming that Auto-Encoders incur larger reconstruction errors on anomalies than normal objects. Some other techniques use the same principle, but apply different deep learning techniques to learn the data representation, such as Generative Adversarial Networks [2, 17, 29], self-learning models [10] and Auto-regressive models [1]. One-class classification-based methods [8, 19–21, 23] instead learn a feature embedding that maps normal objects into a minimal volume hyper-sphere; then the objects out of the hyper-sphere are considered as anomalies. The second category of deep anomaly techniques [24, 25, 27, 33] use learned deep embedding to enhance the classical shallow anomaly detection methods. To learn a representation that is effective in separating anomalies, most of these methods require a clean training data set – a data set not containing any anomalies. However, such clean training data rarely exist in real applications.

Robust Deep Anomaly Detection. Robust deep anomaly detection [4, 5, 28, 32] targets this problem. Based on the assumption that anomalies in the training samples tend to incur large training loss in the training process, these techniques iteratively remove anomalies from the training set in each training epoch. However, they suffer from the chicken-egg problem. That is, identifying anomalies based on the training loss requires an accurate model, while training an accurate model needs a clean training set. Another strategy is to use the deep learning techniques that are robust to anomalies [8, 16] to learn the representation. However, to overcome the influence of anomalies these techniques often assume the distribution of the normal examples is known beforehand. This assumption usually does not hold in practice.

Semi-supervised Deep Anomaly Detection Semi-supervised deep anomaly detection [11, 14, 20] uses a small number of anomaly examples to improve the accuracy of unsupervised deep anomaly techniques. Similar to classical semi-supervised classification, their key idea is to use these anomaly examples as *labeled training data* that are modeled as *labeled loss* to supplement the loss function of the unsupervised deep learning method. However, these techniques still assume that the unlabeled training data is clean and essentially treat them as labeled normal examples. Therefore, they suffer from the performance degradation caused by the hidden anomalies in the unlabeled training data. Our ELITE approach instead uses a small set of anomaly examples as *validation set*. It effectively discovers the anomalies hidden in the polluted training data and turns these anomalies into useful signals that help to learn a data representation that better distinguishes between normal and abnormal samples.

3 PRELIMINARIES

3.1 Problem Definition

Given a set of unlabeled training samples $\mathbb{X}^U : \{x_1^u, \cdots, x_N^u\}$ that contains anomalies, and a small set of labeled samples $\mathbb{X}^L : \{(x_1^l, y_1^l), \cdots, (x_M^l, y_M^l)\} \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{Y} \in \{-1, 1\}$ with $y^l = 1$ denoting normal sample and $y^l = -1$ denoting anomalies, the goal is to train a neural network $\phi(x; \theta)$ that assigns small anomalous scores to normal data and large anomalous scores to anomalies:

$$\Omega(x)|_{y=-1} \geq \Omega(x)|_{y=1} + C \tag{1}$$

In Eq. 1, $\Omega(x)$ represents the anomalous score of x, while C is a hyper-parameter that controls the margin of anomalous score between normal data samples and anomalies.

3.2 Unsupervised and Semi-supervised Deep Anomaly Detection

To better present our proposed approach in Sec. 4, in this section we briefly introduce the key concepts of unsupervised and semi-supervised deep anomaly detection, using one-class classification-based methods [19, 23], deep Auto-Encoder-based methods [3, 6, 13, 22, 30], and semi-supervised DeepSAD [20] as examples.

3.2.1 Unsupervised Deep Anomaly Detection. Let $\phi(x;\theta)$ be a neural network parameterized by θ, and $\Omega(x)$ be the anomalous score function for a data sample x. The goal of deep one-class classification [19, 23] is to map the training samples into a compact hypersphere in the learned latent space, where $\Omega(x) = \|\phi(x,\theta) - o\|^2$ with o denoting the center of the learned hypersphere.

The Auto-Encoder-based methods train a dimension reduction model that reconstructs all training samples with small error. Naturally, it uses the reconstruction error as the anomalous score function, i.e. $\Omega(x) = \|\phi(x;\theta) - x\|$. The training objective is to minimize the average anomalous score of the training samples as shown in Eq. 2.

$$\arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \Omega(x) \tag{2}$$

These unsupervised deep anomaly methods work well when the training dataset contains no or only very few anomalies. However, this assumption does not hold in many real applications. Minimizing the anomalous score of all training samples thus causes performance degradation as discussed in Sec. 1.

3.2.2 Semi-Supervised Deep Anomaly Detection. As a semi-supervised deep anomaly method, DeepSAD [20] uses the training loss incurred on the labeled anomaly samples to compensate the loss function of the unsupervised Deep SVDD [19].

$$\arg\min_{\theta} \frac{1}{N+M} \sum_{i=1}^{N} \|\phi(x_i,\theta) - o\|^2 + \frac{1}{N+M} \sum_{j=1}^{M} (\|\phi(x_j,\theta) - o\|^2)^{y_j} \tag{3}$$

In Eq. 3, o represents a vector in the deep feature embedding. N and M are the size of the unlabeled and labeled set respectively. The first part of Eq. 3 is identical to the loss function of the unsupervised Deep SVDD [19]. We call it *unsupervised loss*. The second part corresponds to the *supervised loss*. As a penalization function, it pushes the labeled anomalies further away from the center.

4 PROPOSED METHOD: ELITE

4.1 Overview of ELITE

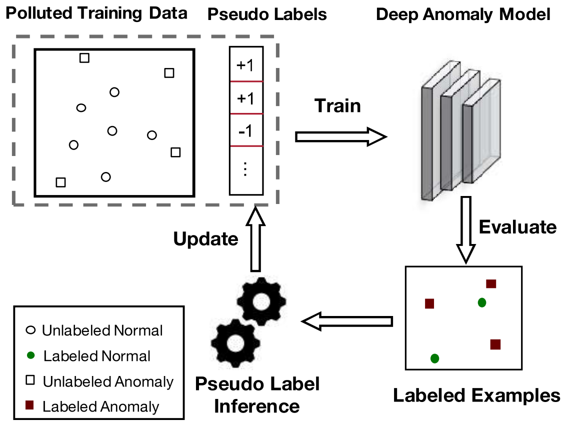


Figure 2: Overview of ELITE

Next, we introduce ELITE, a novel approach that effectively leverages a small number of labeled examples to solve the pollution problem of training samples. ELITE uses the labeled examples as validation set to evaluate the model trained on the unlabeled training samples. The key idea of ELITE is to infer the labels of the unlabeled training samples as normal or anomalous according to the potential influence on model's validation loss. It then learns from the corrected labels a better deep anomaly model. In this way, ELITE no longer relies on the availability of a clean training dataset.

Fig. 2 depicts the overall process of ELITE. Given a polluted training set $\mathbb{X}^U : \{x_1^u, \cdots, x_n^u\}$, ELITE starts with assigning a *pseudo label* to each sample in $\mathbb{X}^U$ and trains a deep learning model on these pseudo labels. Initially, we assume all samples are normal. It then uses the labeled examples to validate the effectiveness of the model. Next, ELITE uses a pseudo label inference method that leverages the gradient of the validation loss to correct the pseudo labels of the training samples in a way guaranteed to reduce the validation loss. ELITE then updates the deep anomaly model based on the corrected labels. It iterates the pseudo label inference and model update steps until updating the labels of the training samples no longer decreases the validation loss. ELITE deploys the final deep anomaly model to detect anomalies from user data.

In the rest of this section, we first introduce ELITE's objective functions including the training loss and validation loss in Sec. 4.2. Then in Sec. 4.3 we propose an effective strategy to update the pseudo labels and analyze its time complexity and convergence. Finally, we show how ELITE works seamlessly with the existing unsupervised anomaly methods using Deep SVDD [19] as example.

4.2 Objective Functions

4.2.1 Training loss. The objective of ELITE is to train a deep learning model $\phi(x;\theta)$ that assigns large anomalous score to anomalies and small anomalous score to normal data, e.g., $\Omega(x)|_{y=-1} \geq \Omega(x)|_{y=1} + C$. To achieve this goal, we design a tailored hinge loss function that copes with anomalous and normal samples differently. More specifically, given the pseudo label y of the training sample x, we define the loss function as:

$$l(x,y) = \begin{cases} \Omega(x), & y = 1 \\ \max\{C - \Omega(x), 0\}, & y = -1 \end{cases} \tag{4}$$

In Eq. 4, $\Omega(x)$ can be any anomalous score function used by existing unsupervised deep anomaly methods as discussed in Sec. 3.2.

Given the pseudo labels y and the loss function defined in Eq. 4, ELITE learns the optimal parameters $\theta^*(y)$ to minimize the average loss incurred by these pseudo labels. The objective function is defined as follows.

$$\theta^*(y) = \arg\min_{\theta} \frac{1}{N^u} \sum_{i=1}^{N^u} l(x_i, y_i) \tag{5}$$

Given the pseudo labels $\hat{y}$, it is straightforward to learn $\theta^*(y)$, using the existing training methods.

Note in Eq. 4 C is a hyper-parameters that controls the margin of anomalous score between normal samples and anomalies. The optimal parameters $\theta^*(y)$ will concurrently minimize the anomalous score of normal samples and grow the anomalous score of anomalies to a value no smaller than C.

An appropriate hyper-parameter C is critical to the performance of ELITE. A too large C tends to make the training process unstable, while a small C fails to separate anomalies from normal samples. We design an intuitive method to automatically determine C. Given an unsupervised counterpart of ELITE denoted as $\phi(x;\theta^0)$ where θ^0 represents its initial parameters, we simply set the hyper-parameter C as its training loss averaged on all samples. The intuition is that because the training process targets minimizing the training loss on the normal examples, the final model will produce a training loss on each normal example that in average is guaranteed to be much smaller than the initial average loss. Therefore, a hyper-parameter C set in this way tends to be effective in separating anomalies from normal samples.

4.2.2 Validation Loss. Given a set of labeled examples as validation set, ELITE defines the validation loss $\mathcal{L}^v$ as follows.

$$\mathcal{L}^v(\theta) = \frac{1}{N^l} \sum_{j=1}^{N^l} l(x_j^l, y_j^l; \theta) \tag{6}$$

In Eq. 6, N^l represents the number of labeled examples and $l(x_j^l, y_j^l; \theta)$ corresponds to the training loss function (Eq. 4).

ELITE aims to assign a pseudo label y to each unlabeled training sample so that the validation loss of the trained model is minimized.

$$y^* = \arg\min_{y} \mathcal{L}^v(\theta^*(y)) \tag{7}$$

Here $\theta^*(y)$ corresponds to the optimal parameters learned from the current pseudo labels as discussed in Sec. 4.2.1.

4.3 Pseudo Label Inference

The key of ELITE is to effectively identify the optimal pseudo labels that minimize the model's validation loss. Obviously, inferring such optimal pseudo labels by recursively flipping the label of each sample, re-training the deep anomaly model, and calculating the validation loss will be too expensive.

To solve this problem, ELITE proposes an efficient pseudo label inference method, called ALICE. The key idea is to use the gradient of the current model's validation loss to predict how altering the label of one training sample will change the validation loss.

4.3.1 Meta-gradient-based Pseudo Label Inference

Assume we have already trained a model using all training samples $\mathbb{X}^U$ and denote its learned parameters as θ^*. Given a training sample x_t in $\mathbb{X}^U$, if we flip its label, we could learn a new model parameterized by θ^*_-.

Let $L^v(\theta)$ denote the validation loss of a model parameterized by θ, that is, the model's loss on the validation set. If we are aware of the difference between the validation loss of the original model θ^* and that of the new model θ^*_-, namely, $L^v(\theta^*) - L^v(\theta^*_-)$, it will be straightforward to decide if we should flip the label of x_t. That is, assume x_t was normal. If $L^v(\theta^*) - L^v(\theta^*_-) > 0$, ELITE should flip x_t to be abnormal, and change its pseudo label as $\hat{y} = -1$, because this will reduce the validation loss. Otherwise, x_t remains normal.

Because we already have θ^* of the original model, computing its validation loss $L^v(\theta^*)$ is straightforward, that is, $L^v(\theta^*) = \frac{1}{M} \sum_i^M l(x_i^l, y_i^l; \theta^*)$. The goal of ALICE is to estimate $L^v(\theta^*) - L^v(\theta^*_-)$ without learning the new model θ^*_-.

By the objective function (Eq. 5), θ^* is learned as: $\arg\min_\theta L(\theta)$ where $L(\theta) = \frac{1}{N}[\sum_{i \neq t}^N l(x_i^u, y_i^u; \theta) + \Omega(x_t;\theta)]$. Here by the loss function (Eq. 4), $\Omega(x_t;\theta)$ represents the loss on x_t if considering x_t as normal.

Without loss of generality, we assume C in the loss function (Eq. 4) is large enough and therefore $\max\{C - \Omega(x;\theta), 0\} = C - \Omega(x;\theta)$ that corresponds to the loss of an anomaly x. Now if we change x_t to anomaly, the new model θ^*_- can be learned as follows:

$$\theta^*_- = \arg\min_{\theta} \{L(\theta) - \frac{2}{N}\Omega(x_t;\theta)\} \tag{8}$$

This is because altering the label of x_t from normal to abnormal is equivalent to first removing $\Omega(x_t;\theta)$ from $L(\theta)$, and then adding $C - \Omega(x;\theta^*)$ back.

Next, we use ϵ to represent - $\frac{2}{N}$ that weights the training loss of x_t. Now Eq. 8 changes to: $\theta^*_- = \arg\min_\theta \{L(\theta) + \epsilon\, \Omega(x_t;\theta)\}$. Similar to [7, 15, 18], we consider ϵ as a variable [7]. Now θ^*_- is a function of ϵ, denoted as $\theta(\epsilon)$. When N is sufficiently large, ϵ approaches 0.

ALICE then uses the gradient of $\theta(\epsilon)$ at $\epsilon = 0$ to approximate the change from $L^v(\theta^*)$ to $L^v(\theta^*_-)$.

$$L^v(\theta^*) - L^v(\theta^*_-) = \frac{dL^v(\theta^*(\epsilon))}{d\epsilon}\Big|_{\epsilon=0} \tag{9}$$

We call the gradient $\mathcal{M} = \frac{dL^v(\theta^*(\epsilon))}{d\epsilon}\big|_{\epsilon=0}$ as **meta-gradient**.

Once getting the meta-gradient, applying the update rule defined below is guaranteed to reduce the validation loss.

Definition 1. **Update Rule.**

$$\hat{y} = -\operatorname{sign}(L^v(\theta^*) - L^v(\theta^*_-)) = -\operatorname{sign}(\frac{dL^v(\theta^*(\epsilon))}{d\epsilon}\Big|_{\epsilon=0}) \tag{10}$$

The reason is that a positive value of $L^v(\theta^*_+) - L^v(\theta^*_-)$ means treating the new training sample as an anomaly will lead to a smaller validation loss than treating it as normal, and vice versa.

Note above we assume the training sample x_t was originally normal. However, the update rule equally works if x_t was abnormal.

4.3.2 Meta-gradient Estimation

To compute meta-gradient, the only thing missing here is $\theta^*(\epsilon)$. Similar to [18] ALICE approximates $\theta^*(\epsilon)$ by taking one step of gradient descent on the original model θ^*.

$$\hat{\theta}(\epsilon) = \theta^* - \eta_\theta \epsilon \nabla_{\theta^*} \Omega(x_t, \theta^*) \tag{11}$$

η_θ represents leaning rate, a hyper-parameter of deep learning.

Given $\hat{\theta}(\epsilon)$, ALICE now is ready to apply the update rule to approximate $\hat{y}_i$. More specifically,

$$\begin{aligned}\hat{y}_i &= -\operatorname{sign}\,(\frac{dL^v(\hat{\theta}(\epsilon))}{d\epsilon}\Big|_{\epsilon=0}) \\ &= -\operatorname{sign}\,(\frac{d}{d\epsilon}\frac{1}{M}\sum_{i=1}^{M} l(x_i^l, y_i^l; \hat{\theta}(\epsilon))|_{\epsilon=0})\end{aligned} \tag{12}$$

Intuitive Interpretation of ALICE. First, we unroll Equation 12 with the chain rule. Given a training sample x_i, we have $\hat{\theta}(\epsilon) = \theta^*$ when $\epsilon = 0$. Then we have:

$$\begin{aligned}\hat{y}_i &= -\operatorname{sign}\,(\frac{dL^v(\hat{\theta}(\epsilon))}{d\epsilon}) \\ &= \operatorname{sign}\,(\frac{L^v(\hat{\theta}(\epsilon))}{d\theta}\Big|_{\hat{\theta}(\epsilon)} \frac{d(\theta^* - \frac{1}{N}\eta_\theta \epsilon \nabla_\theta \Omega(x_i;\theta^*))}{d\epsilon}\Big|_{\epsilon}) \\ &= \operatorname{sign}\,(\frac{\eta_\theta}{N}\frac{dL^v(\theta^*)}{d\theta}\Big|_{\theta^*} \frac{d\Omega(x_i;\theta^*))}{d\theta}\Big|_{\theta^*})\end{aligned} \tag{13}$$

Eq. 13 shows that $\hat{y}_i$ corresponds to an inner product between the gradient of the training loss of the given training sample and the gradient of the validation loss. Given a training sample x_i initialized as normal, if its gradient is in the same direction to the gradient of the validation loss, then x_i will indeed be a normal object. This is because in this case minimizing its training loss by gradient descent – the typical practice of deep learning optimization, will also minimize the validation loss. Otherwise, x_i should be an anomaly.

4.3.3 Learning at Scale

The Learning process. Next, we introduce how ELITE infers the optimal pseudo labels for the entire unlabeled dataset. ELITE fuses ALICE into every iteration during the training process of the deep anomaly model and dynamically adjusts the labels of the training samples. ELITE starts with assuming that all unlabeled training samples are normal. Once one training iteration is done, ELITE estimates the meta-gradient for each sample x_i and applies the update rule to update its pseudo label. Thereafter, ELITE updates the parameters of the deep anomaly model using Eq. 14.

$$\theta_{t+1} = \theta_t - \eta_\theta[\frac{1}{N}\sum_{i=1}^{N}\alpha_i \nabla_\theta l(x_i, \hat{y}_i; \theta)] \tag{14}$$

In Eq. 14, θ_{t+1} represents the new parameters, while θ_t represents the parameters produced in last iteration. η_θ is the learning rate. Same to the traditional gradient descent optimization, Eq. 14 uses the gradient of the loss function $\nabla_\theta l(x_i, \hat{y}_i; \theta)$ to update θ_t. But ELITE weights the meta-gradient at each training sample x_i with $\alpha_i = \eta_M \cdot \|\mathcal{M}_i\|$, where $\|, \mathcal{M}_i\|$ represents the absolute value of the meta-gradient, and η_M is a hyper-parameter. The intuition is that, if the meta-gradient of a training sample x_i has a larger absolute value, x_i is more important. This is because by our ALICE method, potentially x_i will contribute more in reducing the validation loss.

Batch Optimization. Although ELITE effectively avoids recursively re-training the deep learning model, it still tends to be expensive when the unlabeled training dataset is large. Similar to [12, 18], ELITE employs a mini-batch based optimization strategy to address the efficiency concern. During each training iteration, ELITE randomly divides the unlabeled training samples into many mini-batches and then concurrently updates the labels with respect to each mini-batch. Each mini-batch contains only $n \ll N$ unlabeled objects. Therefore, it significantly speeds up the training process. As a standard deep learning training process, ELITE can run on any deep learning platform such as TensorFlow and Pytorch.

Time Complexity Analysis. Compared to unsupervised deep anomaly methods, ELITE requires an extra forward and backward pass to obtain the gradient of each training sample and an additional forward and backward pass to calculate $\hat{y}_i$. Thus, ELITE is approximately 3× slower than the unsupervised deep anomaly methods. We argue that the additional computing cost is worthwhile in practice because ELITE is robust to polluted training data and significantly improves the accuracy of anomaly detection.

Convergence Analysis.

Theorem 2. *Suppose the validation loss $L^v(x;\theta)$ is Lipschitz smooth with constant L, and the gradient of training data is bounded by σ. Then as long as the learning rate $\eta_y\eta_\theta \le \frac{2n}{L\sigma^2}$, the validation loss decreases monotonically,*

$$L^v(\theta_{t+1}) \le L^v(\theta_t) \tag{15}$$

Proof. Without loss of generality, we assume C in the loss function (Eq. 4) is large enough and therefore $\max\{C - \Omega(x;\theta), 0\} = C - \Omega(x;\theta)$ which corresponds to the loss of an anomaly x. Combining Equation 14 and Equation 13, we have,

$$\theta_{t+1} = \theta_t - \eta_y\eta_\theta\{\frac{1}{n}\sum_{i=i}^{n}[\nabla_\theta L^v(\theta_t)\nabla_\theta L_i(\theta_t)]\nabla_\theta L_i(\theta_t)\} \tag{16}$$

where $\nabla_\theta L^v(\theta_t) = \frac{\partial L^v(\theta_t)}{\partial\theta}\Big|_{\theta_t}$ and $\nabla_\theta L_i(\theta_t) = \frac{\partial\Omega(x_i;\theta_t))}{\partial\theta}\Big|_{\theta_t}$. For simplicity of expression, we denote $\nabla_\theta L^v(\theta_t)$ as ∇L^v and $\nabla_\theta L_i(\theta_t)$ as ∇L_i.

Since the validation loss $L^v(\theta)$ is Lipschitz smooth with constant L, from [9],

$$L^v(\theta_{t+1}) \le L^v(\theta_t) + (\nabla L^v)^T \triangle\theta + \frac{L}{2}\|\triangle\theta\|^2 \tag{17}$$

Plugging in Equation 16,

$$L^v(\theta_{t+1}) \le L^v(\theta_t) - I_1 + I_2, \tag{18}$$

where,

$$I_1 = \eta_y\eta_\theta\sum_{i=1}^{m}(\nabla L^v \nabla L_i)^2 \tag{19}$$

and,

$$\begin{aligned}I_2 &= \frac{L}{2}\|\frac{\eta_y\eta_\theta}{n}\sum_{i=1}^{m}(\nabla L^v\nabla L_i)\nabla L_i\|^2 \\ &\le \frac{L}{2}\frac{\eta_y^2\eta_\theta^2}{n^2}\sum_{i=1}^{m}\|(\nabla L^v\nabla L_i)\nabla L_i\|^2 \\ &= \frac{L}{2}\frac{\eta_y^2\eta_\theta^2}{n^2}\sum_{i=1}^{m}(\nabla L^v\nabla L_i)^2\|\nabla L_i\|^2 \\ &\le \frac{L}{2}\frac{\eta_y^2\eta_\theta^2}{n^2}\sum_{i=1}^{m}(\nabla L^v\nabla L_i)^2\sigma^2\end{aligned} \tag{20}$$

The first inequality comes from the triangle inequality, and the second inequality holds since the gradient of training data is bounded by σ. If we denote a value τ at iteration t, $\tau_t = \sum_{i=1}^{m}(\nabla L^v \nabla L_i)^2$, then we have,

$$L^v(\theta_{t+1}) \leq L^v(\theta_t) - \frac{\eta_y \eta_\theta}{n}\tau_t(1 - \frac{L\eta_y\eta_\theta\sigma^2}{2n}) \quad (21)$$

Note by definition τ_t is non-negative and $\eta_y\eta_\theta \leq \frac{2n}{L\sigma^2}$, we have,

$$L^v(\theta_{t+1}) \leq L^v(\theta_t) \quad (22)$$

Theorem 2 is proven. □

4.4 Example: Applying ELITE to Deep SVDD

In this section, we show that ELITE is able to easily adapt existing unsupervised deep anomaly methods to benefit from the anomaly examples at hand. More specially, to support one unsupervised deep anomaly method, the only change we need to make is to plug its anomalous score function $\omega(x)$ into the loss function of ELITE (Eq. 4 in Sec. 4.2). Next, we use Deep SVDD [19] as an example to showcase this. Deep SVDD is briefly reviewed in Sec. 3.2.

Algorithm 1 ELITE on Deep SVDD

```
Input:
  Unlabeled data: $X_U : \{x_1, \ldots, x_N\}$
  Validation examples: $X_V : (\{(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_M, \tilde{y}_M)\}$
  Hyperparameters: $\eta_M, \eta_\theta$, Hypersphere center, $o$, Margin, $C$
  Loss Function: $\Omega(x, o) = \|x - o\|$
Initialize:
    Neural network weights: $\theta$
for each epoch do
    for each mini-batch do
        Draw mini-batch $B_U$:$\{x_1, \ldots, x_n\}$ from $X_U$
        Draw mini-batch $B_V$:$\{(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_m, \tilde{y}_m)\}$ from $X_V$
        Initialize:
        $\hat{y}_i \leftarrow 0 \ \forall x_i \in B_U$
        $\hat{\theta}(\hat{y}) \leftarrow \theta - \eta_\theta[\frac{1}{n}\sum_{i=0}^{n} \hat{y}_i \nabla_\theta \Omega(x, \theta)]$
        Update:
        $\mathcal{M}_i \leftarrow \eta_y \frac{\partial}{\partial y_i} \frac{1}{m} \sum_{i=1}^{m} L^v(\tilde{x}_i, \hat{\theta}(\hat{y}))|_{\tilde{y}})$
        $\hat{y}_i = -\operatorname{sign}(\mathcal{M}_i)$
        $\alpha_i = \eta_M \cdot \|\mathcal{M}_i\|$
        $\theta \leftarrow \theta - \eta_\theta[\frac{1}{n}\sum_{i=0}^{n} \alpha_i \nabla_\theta \text{ł}(x_i, \hat{y}_i; \theta)]$
Output:    Trained Model: $\phi^\star(x, \theta^\star)$
```

As shown in Algorithm 1, ELITE starts with initializing the neural network's parameters θ and the hypersphere center o exactly as what Deep SVDD does. Then, in each epoch ELITE samples a mini-batch of unlabeled samples B_U and uses the labeled samples as validation set. Next, ELITE assigns an initial pseudo label $\hat{y}_i$ to each unlabeled sample in B_U. ELITE uses these pseudo labels to learn the parameters θ of the network. It then computes the validation loss using the loss function in Eq. 4, alters the pseudo labels according to the update rule in Def. 1, and updates the parameters by Eq. 14. These steps iterate until the validation loss is minimized or reaching the epoch limit.

5 EXPERIMENTS

We conduct an experimental study to evaluate the effectiveness of ELITE. Specifically, we focus on the following four questions:

1. **Robustness to Polluted Training Data**: How does ELITE compare with existing deep anomaly techniques in term of the robustness to the polluted training data?

2. **Performance with different number of labels**: How does ELITE perform in contrast to the existing deep anomaly methods when using different number of labels?

3. **Sensitivity Analysis**: Is ELITE sensitive to the selection of its hyper-parameters?

4. **Training Mechanism**: How is our training mechanism different from the standard semi-supervised learning?

5.1 Experiment Setup and Methodology

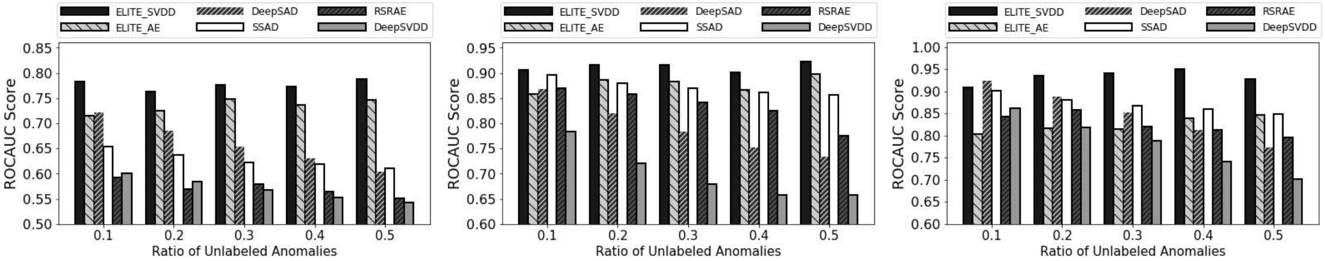
Experimental Setup. All experiments are conducted on Google Cloud with a virtual machine with 12 CPU cores and 4 P-100 GPUs. All code is developed with Python 3 on Pytorch 1.5.0.

Datasets. We evaluate ELITE using three benchmark datasets which are also frequently used in the experiments of the state-of-the-art deep anomaly works we compare against [19, 20].

- **MNIST:** The MNIST dataset consists of 28 × 28 pixel grayscale images of the handwritten digits 0-9. Each image contains only one digit centered in the frame and is given a class label corresponding to the digit it contains. Given the relatively simple and clear shape of the digits and the consistent black background, we consider it as the least complex dataset among the three datasets we use.
- **FMNIST:** The FMNIST or Fashion-MNIST dataset was created to be a more complex replacement for MNIST. FMNIST consists of 28x28 pixel grayscale images for ten types of clothing articles such as T-shirts, coats, and sneakers with corresponding labels.
- **CIFAR-10:** The CIFAR-10 dataset consists of 32x32 color images of ten distinct object classes. Four of the classes are types of vehicles – airplane, automobile, ship, truck – with the remaining six being varying types of animals. Images in this dataset were originally drawn from internet search engines and converted to the 32x32 resolution.

Alternative Methods. We compare ELITE against the state-of-the-art unsupervised (DeepSVDD [19]), semi-supervised (DeepSAD [20], SSAD [11], and robust (RSRAE [16]) deep anomaly methods. Moreover, to show ELITE is model agnostic, we implement ELITE on top of two types of unsupervised deep anomaly models, namely the one-class classification-based DeepSVDD [19] and Auto-Encoder.

- **DeepSVDD** [19] is the state-of-the-art unsupervised anomaly method, which detects anomalies by mapping the training data into a compact hyper-sphere, assuming the training data is clean.
- **DeepSAD** [20] extends Deep SVDD method to the semi-supervised setting and uses the labeled examples as training data to improve the accuracy of anomaly detection. We consider DeepSAD as the most related work to ELITE.
- **SSAD** [11] is a popular shallow semi-supervised anomaly method built on vanilla SVDD [26]. Similar to DeepSAD, it directly uses the labeled examples as training data and encourages the model to generate large anomalous score on the labeled anomalies.
- **RSRAE** is the state-of-the-art robust deep anomaly method, which combines a simple Auto-Encoder with robust deep learning techniques, more specifically Robust Subspace Recovery (RSR) layer.

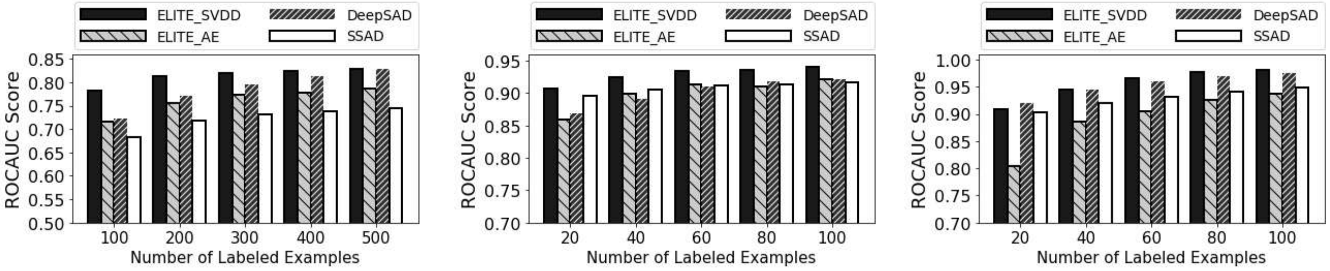


(a) **CIFAR-10**

(b) **FMNIST**

(c) **MNIST**

Figure 3: ROCAUC: Varying the Ratio of Anomalies in Training data

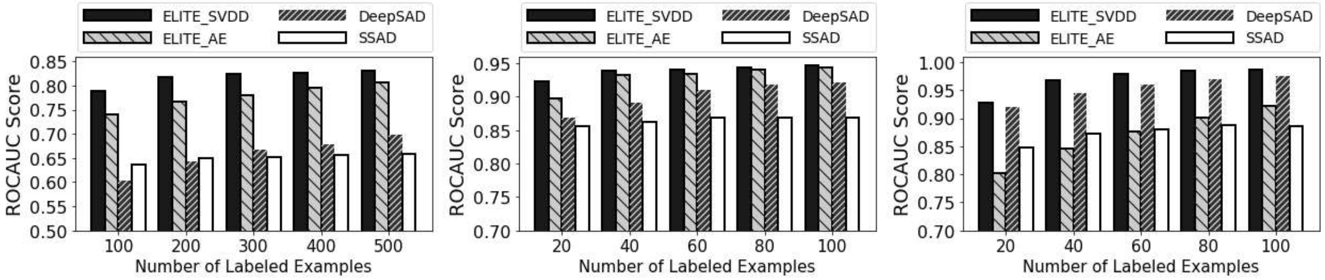


(a) **CIFAR-10** (**r_p** = 0.1)

(b) **FMNIST** (**r_p** = 0.1)

(c) **MNIST** (r_p = 0.1)

Figure 4: ROCAUC: Varying the Number of Labeled Examples (Lightly Polluted)



(a) **CIFAR-10**(r_p = 0.5)

(b) **FMNIST** (**r_p** = 0.5)

(c) **MNIST** (**r_p** = 0.5**)**

Figure 5: ROCAUC: Varying the Number of Labeled Examples (Heavily Polluted)

The RSR layer is used to learn a subspace within the latent space where normal and anomalous samples are well separated.

Methodology. Following the state-of-the-art [20], for each dataset we select one class as normal and consider other classes as abnormal. To ensure that results are not class dependent, we repeat each set of experiments with a different class selected as the normal class until all classes are exhausted. We then report the average of these results. For each experiment, we randomly select 5,000 objects to create a training dataset. This set contains samples from both normal and anomalous classes, with the ratio of anomalies controlled by the value r_p. In general r_p is selected to be small such that the majority of the training samples are drawn from the normal class, while the few anomalies are drawn from the remaining classes. From each dataset, we randomly sample an equal number of normal samples and anomalies to be used as the labeled training dataset and consider the remaining samples to be unlabeled. We vary the ratio of training points allocated to the labeled training set r_l and the ratio of pollution in the training dataset r_p to analyze the performance of ELITE in a wide variety of scenarios. Again, following [20], we use the Area Under Curve (AUC) score of the Receiver Operating Characteristic (ROC) curve as the metric to evaluate the accuracy of each method.

5.2 Varying the Ratio of Anomalies

In this experiment, we investigate the robustness of different deep anomaly detection methods to the increasing ratio of anomalies in the training set. To do this, we vary the ratio of anomalies in training set from 0.1 to 0.5. We fix the ratio of labeled examples r_l, and repeat the experiments on all ten classes and report the average results over all experiments on each dataset. For MNIST and FMNIST we use 20 labeled examples, while for CIFAR-10 we use 100 to account for its much higher complexity.

Figure 3 indicates that both of our ELITE-based methods, ELITE _AE and ELITE _SVDD, outperform all other methods by up to 30%, especially on the complex datasets such as CIFAR-10. Also, we find that the performance of ELITE never degrades with the increasing ratio of anomalies in training data. However, the performance of the state-of-the-art methods, including the robust deep anomaly method RSRAE, significantly decrease as the ratio of anomalies in the training data increases. Furthermore, on the *CIFAR-10* and *FMNIST* dataset, ELITE achieves even higher performance when the anomaly ratio is highest, i.e. $r_p = 0.5$. This is because ELITE not only identifies the anomalies in the training dataset, but also effectively uses them to learn an anomaly-aware data representation that improves the accuracy of anomaly detection. This confirms that ELITE not only outperforms the other methods but also is much more robust to anomalies in the training dataset. Furthermore, we find that the shallow *SSAD* method even outperforms its deep competitor, *DeepSAD*. We argue that this shows it is easier for deep anomaly detection models to overfit the anomalies in the training data due to their complex network structure using a large number of parameters.

5.3 Varying the Ratio of Labeled Examples

In this scenario, we compare the performance of different semi-supervised deep anomaly methods given a different number of labeled examples. For this experiment, we evaluate our method on both lightly polluted training data where $r_p = 0.1$, and heavily polluted training data where $r_p = 0.5$. For FMNIST and MNIST we vary the number of labeled samples from 20 to 100 in steps of 10 ($r_l = 0.004 - 0.02$), while for CIFAR-10 we test 100 to 500 labeled samples with intervals of 50 ($r_l = 0.02 - 0.1$). Again, we exhaustively use every class in each dataset as normal samples and report each dataset's average result.

Figure 4 and Figure 5 show the result on lightly polluted ($r_p = 0.1$) and heavily polluted datasets ($r_p = 0.5$) respectively. Both of our methods significantly outperform the other methods on all heavily polluted datasets by up to 25%. This again shows ELITE is significantly more robust to anomalies in the training data, because ELITE effectively leverages the labeled examples. Moreover, ELITE reaches very high accuracy with very few labeled examples. This is because ELITE uses the labeled examples as validation data, and it requires much fewer labels to evaluate the model performance than training the model. Therefore, although increasing the number of labels improves the performance of *DeepSAD*, it is consistently less accurate than our ELITE-based methods. Note that even when the dataset is lightly polluted, *DeepSAD* still requires 2 - 3 times more labeled examples to achieve comparable performance to ELITE on complex datasets like *CIFAR-10* and *Fashion-MNIST*.

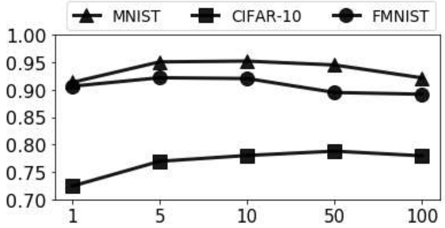


Figure 6: Sensitivity Analysis of η_M of ELITE

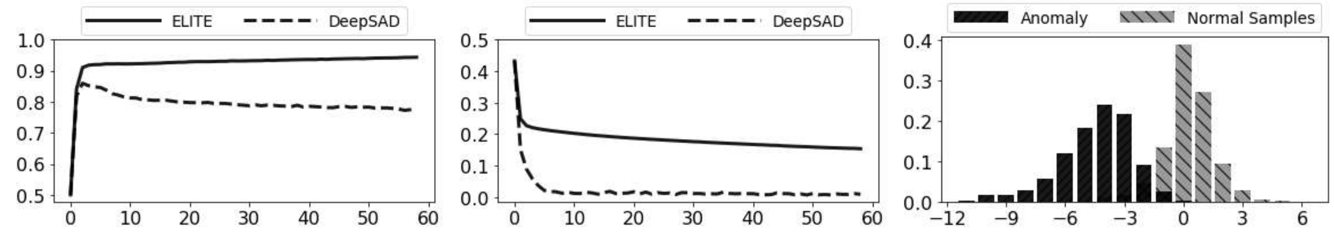
5.4 Sensitivity Analysis

Here we investigate how sensitive ELITE is to the value of hyper-parameter η_M which controls the factor that the validation loss plays in the learning process. We report the results on our ELITE _SVDD method, although ELITE _AE shows the similar trend. We set r_p to 0.1 and we use 20 labeled examples for both MNIST and FMNIST and 100 labeled examples for CIFAR-10. We vary η_M from 1 to 100, while keeping all other hyper-parameters fixed. Figure 6 show that the performance of ELITE is stable. This confirms that ELITE is not sensitive to the hyper-parameter η_y, and thus partially mitigates the hyper-parameter tuning problem. We also observe that *FMNIST* and *MNIST* prefer small η_M as the performance decreases with the increase of η_M. However, on CIFAR-10 ELITE achieves slightly better performance as η_M increases. Therefore, based on these results, we recommend to set a large η_M on complex datasets and set a small value if the data set is relatively simple.

5.5 Evaluating the Training Mechanism

5.5.1 Training Process. To better understand the training mechanism of ELITE, we compare ELITE with the semi-supervised *DeepSAD* which is based on the classical semi-supervised classification mechanism. To ensure a fair comparison, we apply the same loss function (Eq. 4) to both ELITE and *DeepSAD*. We report the results on the *FMNIST* dataset. Figure 7(a) and Figure 7(b) depict how ROCAUC score and labeled loss change over the training process. In *DeepSAD*, the loss on labeled examples quickly decreases to 0, while it reduces slowly in ELITE. Meanwhile, the ROCAUC score of *DeepSAD* decreases after reaching the peak, potentially because the deep neural network starts overfitting the labeled examples. In contrast, the ROCAUC score of ELITE increases stably.

5.5.2 Distribution of Anomalous Scores. As discussed in Sec. 4.3, ALICE, ELITE's label inference method, uses meta-gradient to determine the anomalous score of the training data, because the meta-gradient of anomalies tends to show distinct patterns from that of normal samples. Here we verify its effectiveness by measuring the distribution of $\hat{y} \cdot \|\mathcal{M}\|$ which represents the anomalous score of each training sample. In this experiment, we run ELITE on *MNIST* with $r_p = 0.5$ and $r_l = 0.004$. We separately report the $\hat{y} \cdot \|\mathcal{M}\|$ of normal and anomalous samples averaged over the first 500 iterations. Figure 7(c) shows that the anomalous score effectively separates anomalous samples from normal ones. That is, ELITE assigns small scores (negative) to anomalous samples, while large scores (positive) to normal samples. Although ELITE still erroneously assigns negative score to some normal samples,



(a) **ROCAUC VS Training Epochs**

(b) **Labeled Loss VS Training Epochs**

(c) **Distribution of Anomalous Scores**

Figure 7: ROCAUC: Varying the Number of Labeled Examples

their scores still tend to be larger than those of the real anomalies. This confirms the effectiveness of our ALICE method.

6 CONCLUSION

In this work, we propose ELITE that addresses a fundamental problem in semi-supervised and unsupervised deep anomaly detection, namely requiring a clean training data not polluted by anomalies. ELITE solves above problems by proposing a novel optimization methodology. Unlike the classical semi-supervised classification methodology, ELITE uses labeled examples as validation set and continuously discovers the anomalies in the polluted training data and learns a better deep anomaly model based on the cleaned training data. Our experiments in rich variety of scenarios confirm ELITE's superiority to the state-of-the-art and its robustness to polluted training data.

7 ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation under grants IIS-1910880, CSSI-2103832, CNS-1852498 and by the U.S. Dept. of Education under grant P200A180088.

REFERENCES

[1] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. 2019. Latent space autoregression for novelty detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 481–490.

[2] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. 2018. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*. Springer, 622–637.

[3] Jerone TA Andrews, Edward J Morton, and Lewis D Griffin. 2016. Detecting anomalous data using auto-encoders. *International Journal of Machine Learning and Computing* 6, 1 (2016), 21.

[4] Laura Beggel, Michael Pfeiffer, and Bernd Bischl. 2019. Robust anomaly detection in images using adversarial autoencoders. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 206–222.

[5] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. 2017. Robust, deep and inductive anomaly detection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 36–51.

[6] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. 2017. Outlier detection with autoencoder ensembles. In *Proceedings of the 2017 SIAM international conference on data mining*. SIAM, 90–98.

[7] R Dennis Cook and Sanford Weisberg. 1982. *Residuals and influence in regression.* New York: Chapman and Hall.

[8] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. 2016. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition* 58 (2016), 121–134.

[9] Pontus Giselsson. 2014. Improved fast dual gradient methods for embedded model predictive control. *IFAC Proceedings Volumes* 47, 3 (2014), 2303–2309.

[10] Izhak Golan and Ran El-Yaniv. 2018. Deep anomaly detection using geometric transformations. In *NeurIPS*. 9758–9769.

[11] Nico Görnitz, Marius Kloft, Konrad Rieck, and Ulf Brefeld. 2013. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research* 46 (2013), 235–262.

[12] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. 2020. Safe Deep Semi-Supervised Learning for Unseen-Class Unlabeled Data. ICML.

[13] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. 2002. Outlier detection using replicator neural networks. In *International Conference on Data Warehousing and Knowledge Discovery*. Springer, 170–180.

[14] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2018. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606* (2018).

[15] Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. *arXiv preprint arXiv:1703.04730* (2017).

[16] Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. 2019. Robust Subspace Recovery Layer for Unsupervised Anomaly Detection. *arXiv preprint arXiv:1904.00152* (2019).

[17] Pramuditha Perera, Ramesh Nallapati, and Bing Xiang. 2019. Ocgan: One-class novelty detection using gans with constrained latent representations. In *CVPR*. 2898–2906.

[18] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050* (2018).

[19] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *ICML*. 4393–4402.

[20] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. 2020. Deep Semi-Supervised Anomaly Detection. In *International Conference on Learning Representations*.

[21] Mohammad Sabokrou, Mohammad Khalooei, Mahmood Fathy, and Ehsan Adeli. 2018. Adversarially learned one-class classifier for novelty detection. In *CVPR*. 3379–3388.

[22] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. 4–11.

[23] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation* 13, 7 (2001), 1443–1471.

[24] Ningxin Shi, Xiaohong Yuan, and William Nick. 2017. Semi-supervised Random Forest for Intrusion Detection Network.. In *MAICS*. 181–185.

[25] Hongchao Song, Zhuqing Jiang, Aidong Men, and Bo Yang. 2017. A hybrid semi-supervised anomaly detection model for high-dimensional data. *Computational intelligence and neuroscience* 2017 (2017).

[26] David MJ Tax and Robert PW Duin. 2004. Support vector data description. *Machine learning* 54, 1 (2004), 45–66.

[27] Ryan J Urbanowicz, Melissa Meeker, William La Cava, Randal S Olson, and Jason H Moore. 2018. Relief-based feature selection: Introduction and review. *Journal of biomedical informatics* 85 (2018), 189–203.

[28] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. 2015. Learning discriminative reconstructions for unsupervised outlier removal. In *Proceedings of the IEEE International Conference on Computer Vision*. 1511–1519.

[29] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. 2018. Efficient gan-based anomaly detection. *arXiv preprint arXiv:1802.06222* (2018).

[30] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. 2016. Deep structured energy based models for anomaly detection. *arXiv preprint arXiv:1605.07717* (2016).

[31] Huayi Zhang, Lei Cao, Yizhou Yan, Samuel Madden, and Elke A. Rundensteiner. 2020. Continuously Adaptive Similarity Search. In *SIGMOD*. 2601–2616.

[32] Chong Zhou and Randy C Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *SIGKDD*. 665–674.

[33] Bing Zhu, Wenchuan Yang, Huaxuan Wang, and Yuan Yuan. 2018. A hybrid deep learning model for consumer credit scoring. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*. IEEE, 205–208.