Scrutinizing Shipment Records To Thwart Illegal Timber Trade

Debanjan Datta
Department of Computer Science,
Virginia Tech
Arlington, Virginia, USA

Sathappan Muthiah
Department of Computer Science,
Virginia Tech
Arlington, Virginia, USA

John Simeone
Simeone Consulting LLC
New Hampshire, USA

Amelia Meadows
World Wildlife Fund
Washington, DC, USA

Naren Ramakrishnan
Department of Computer Science,
Virginia Tech
Arlington, Virginia, USA

ABSTRACT

Timber and forest products made from wood, like furniture, are valuable commodities, and like the global trade of many highly-valued natural resources, face challenges of corruption, fraud, and illegal harvesting. These grey and black market activities in the wood and forest products sector are not limited to the countries where the wood was harvested, but extend throughout the global supply chain and have been tied to illicit financial flows, like trade-based money laundering, document fraud, species mislabeling, and other illegal activities. The task of finding such fraudulent activities using trade data, in the absence of ground truth, can be modelled as an unsupervised anomaly detection problem. However existing approaches suffer from certain shortcomings in their applicability towards large scale trade data. Trade data is heterogeneous, with both categorical and numerical attributes in a tabular format. The overall challenge lies in the complexity, volume and velocity of data, with large number of entities and lack of ground truth labels. To mitigate these, we propose a novel unsupervised anomaly detection — **C**ontrastive Learning based **H**eterogeneous **A**nomaly **D**etection (CHAD) that is generally applicable for large-scale heterogeneous tabular data. We demonstrate our model CHAD performs favorably against multiple comparable baselines for public benchmark datasets, and outperforms them in the case of trade data. More importantly we demonstrate our approach reduces assumptions and efforts required hyperparameter tuning, which is a key challenging aspect in an unsupervised training paradigm. Specifically, our overarching objective pertains to detecting suspicious timber shipments and patterns using Bill of Lading trade record data. In order to target records pertinent to timber and forest products, we utilize domain knowledge curated from multiple sources. Detecting anomalous transactions in shipment records can enable further investigation of supply chain constituents.

KEYWORDS

Tabular data, Anomaly detection, Deep Learning

KDD '21, August 15, 2021, Virtual Event, Singapore

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn


Office of the United States Trade Representative | TRADE AGREEMENTS | COUNTRIES & REGIONS

USTR Announces Enforcement Action to Block Illegal Timber Imports from Peru

07/26/2019

Washington, DC – United States Trade Representative Robert Lighthizer today directed the United States Customs and Border Protection (CBP) to block future timber imports from Inversiones WCA E.I.R.L. (WCA), a Peruvian exporter, based on illegally harvested timber found in its supply chain. This marks the second time that the Trump Administration has taken such an enforcement action under the United States – Peru Trade Promotion Agreement's (PTPA) Annex on Forest Sector Governance (Forest Annex), demonstrating its intensified efforts to keep illegal timber out of the United States.

Figure 1: Enforcement remains a challenge for the initiatives and legislation enacted by the US government agencies to thwart illegal timber imports.

1 INTRODUCTION

Detecting suspicious activities in international trade is a persistent challenge faced by law enforcement agencies. Such suspicious activities can include various types of fraud, illicit financial flows, labor rights violations, unlawful wood harvesting, and violations in trade regulations. The focus of our work is building a detection framework targeted towards identifying suspicious shipments in global trade. The United States (US) is the largest importer of timber and wood products globally, valued at $51 billion in 2017, which accounted for over 20% of the global wood and timber products trade. Illegal trade in threatened timber species that are at risk of extinction is detrimental to not just ever-diminishing biodiversity, but also adversely impacts developing economies and is linked to national security [23]. The mislabelling (unintentional) or fraudulent (intentional) declaration of species to US consumers has been investigated [41] and poses a significant problem, though it is unknown whether the false claims exhibited to the consumer were also present on US import declarations.

Finding suspicious transactions can allow agencies to target which shipments need additional scrutiny upon import into the US, and to determine patterns of suspicious trades involving specific companies and trade channels. To achieve a near real-time process that is actionable by enforcement agencies, however, requires an automated framework to detect potentially suspicious timber trades given the volume and velocity of trade data. While manual examination and partially algorithmic tools are utilized by agencies, an integrated framework to highlight actionable trades is not readily available. This can be attributed to factors such as hurdles in policy implementation, inter-agency communication and information-sharing agreements, handling how non-governmental

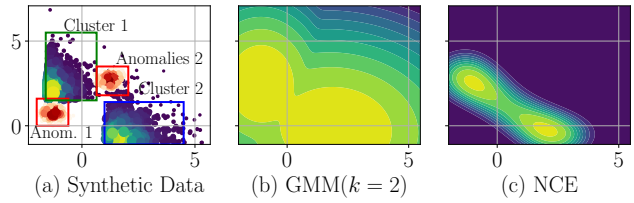

Figure 2: (a) shows an example scenario where data clusters are of arbitrary shape. (b) show the density as learned by GMM and (c) the density learned through contrast using Noise Contrastive Estimation. We can clearly see that the Gaussian assumption made by GMM does not generalize well to such data and thus can significantly affect anomaly detection performance

organizations and government agencies share sensitive information that might lead to an investigation and prosecution, and the lack of unified domain knowledge that can be incorporated into such systems and the absence of annotated data.

Following prior approaches towards the task of fraud detection, we formulate the task as unsupervised anomaly detection problem. Unsupervised anomaly detection in large scale heterogeneous tabular data is a non-trivial task, and existing approaches have certain drawbacks which we aim to address in this work. The challenge of the task is driven by the complexity of trade data, with multiple categorical attributes with high arity (cardinality) as well as numerical attributes which contain vital information and cannot be ignored. Secondly, model designs often have sensitive hyperparameters such as cluster numbers which are difficult to determine apriori in a true unsupervised scenario. Thirdly, there is often a reliance on various distributional assumptions — which may be not be satisfied by arbitrarily complex real world data. Thus it is imperative to overcome these impediments for our task and make the model free of such limiting assumptions, so as to create a automated suspicious shipment detection framework applicable to complex real-world trade data.

We propose a novel model, Contrastive Learning based Heterogeneous Anomaly Detector (CHAD) that attempts to alleviate these problems. The model uses an autoencoder based architecture to obtain a low-dimensional latent representation of data, with additional network features to handle potentially high arities in categorical variables. To estimate the density of data in this latent space we leverage the idea of Noise Contrastive Estimation [18]. To summarize, our contributions in this work are as follows:

(1) Anomaly detection approach based on direct likelihood estimation. Unlike prior approaches we do not make any distributional assumptions, but instead rely on contrastive learning for flexibility and generalizability. Additionally, it is robust to choice of hyperparameters.
(2) A novel and efficient negative sampling approach for heterogeneous data, that is used to train the model.
(3) Experimental evaluation demonstrating the improved performance of our model against competing approaches for trade data. Additional evaluation on multiple open source datasets show improved or competitive performance as well.

2 RELATED WORKS

Unsupervised anomaly detection methods for general data have been discussed in detail in surveys [7, 8, 15]. Application specific approaches like fraud detection [2] and intrusion detection [20] have been explored in prior works. There are prior works that focus on anomaly detection in specific types of data — such as graphs [3] and images [5]. We limit our discussion to some of the more popular techniques and organize similar approaches together.

Traditional approaches to anomaly detection include Kernel Density Estimation [22], Principal Component Analysis and Robust PCA [43]. Some other notable methods are Local Outlier Factor[6] and Isolation Forest [25]. A key challenge with such methods is that they are not scalable, and many of them are not effective for sparse high dimensional input features.

Clustering based and geometric anomaly detection approaches assume *normal* or expected data to posses proximity to each other along with some latent underlying structure, which has been can be utilized to detect outliers. Recent approaches like Deep Embedded Clustering (DEC) [42] and Deep Clustering Network (DCN) [44] are relevant here. Mixture models based approaches like DAGMM [46] which combines a deep autoencoder with a Gaussian mixture model to perform anomaly detection can also be considered under clustering methods. An interesting aspect of DAGMM is a very low dimensional latent representation is augmented with sample reconstruction errors.

Autoencoders utilize a reconstruction-based approach relying on the assumption that anomalous data cannot be represented and reconstructed accurately by a model trained on *normal* data. Autoencoders and its variants such as denoising autoencoders have been used for anomaly detection [34, 37, 45], along with ensemble based approaches [9].

One-class classification based approaches like OCSVM [35] separates the normal data from the anomalies using a hyper-plane of maximal distance from the origin. Support Vector Data Description [39] attempts to find the smallest hyper-sphere that contains all normal data. Both methods use a hyperparamter ν that helps define the boundary of the hyper-plane or hyper-sphere. Similarly, two recent works employ deep learning along with a one-class objective for anomaly detection. DeepSVDD [33] performs anomaly detection combining deep convolutional network as feature extractor and one-class classification based objective. Oneclass-NN [1] uses a similar formulation, albeit using a hyper-plane instead of a hyper-sphere, combined with an alternating minimization based training approach.

Methods for anomaly detection in purely multivariate categorical data include information theoretic methods such as *CompreX* [4], and embedding based methods such as *APE* [10] and *MEAD* [12]. Though *MEAD* uses a trade dataset, it considers only categorical features. We do not consider these methods for comparison since discretization of continuous variables require careful considerations and can lead to information loss [40]. Many of the above mentioned approaches make critical assumptions about the data such as cluster numbers or distributional family, etc. These are often specific to a particular kind of data and also it is difficult to obtain necessary domain knowledge to be able to make such choices. In the following section we present a motivating example demonstrating the

disadvantages of making such assumptions about data and propose an alternative non-parametric density estimation method.

3 DATA AND MOTIVATION

In this section we briefly outline the constraints, requirements and motivation for the data driven system and provide further background on the context of the application. We also present the conceptual motivation in proposing our anomaly detection model for the task, outlining the applicability of such an approach for real world data.

3.1 Trade Data

High resolution trade data contain heterogeneous features with high arity categorical features and numerical features. We specifically use US import Bill of Lading records for our framework from 2015 to 2017 obtained from Panjiva [29]. The raw data is preprocessed, since there are many attributes of which a subset are deemed irrelevant. Additionally many attributes have large percentages of missing data. With the help of domain experts we select the following attributes : *VolumeTEU*, *Quantity*, *WeightKg*, *NumberOfContainers*, *Carrier*, *HSCode*, *PortOfLading*, *PortOfUnlading*, *ShipmentDestination* and *ShipmentOrigin*. The first four are real valued attributes, and the remaining are categorical.

Goods and products involved in global trade are tracked using the standardized Harmonized Schedule (HS) code nomenclature and product classification system. The role of HS codes is important in the context of trade data. We obtain the ontology and data for HS codes from open source repositories containing text descriptions of products, which can include scientific names, family and common names of endangered and commercially viable timber species. We use natural language processing for parsing and lemmatizing, and use regular expression and n-gram based matching to obtain these from the HS code text descriptions after collating HS code data from multiple years. With the help of domain experts, we extract HS codes that can be associated with known high risk timber species. Additionally, we curate HS codes covered by legislation such as the Lacey Act's Plant Declaration Form and data on country-specific logging and export bans. HS codes for products containing solid wood (like furniture) are used to filter out trade records for our system. Although these curated HS codes may contain high risk species, they correspond to such a large number shipments that simple rule-set based matching is neither analyzable nor actionable by end users.

3.2 Conceptual Motivation

Geometric approaches anomaly detection are based on the assumption that nominal data points are clustered, in the original or a transformed space which is supported by the manifold hypothesis [16]. Assumptions are often placed on the shape of the cluster,such as hypersphere or ellipsoid for computational efficiency which the true data distributions may not conform to. Data in latent space can have complex non-linear or non-convex decision boundaries.

We demonstrate the limitations of such assumptions using a simple scenario with data $\in \mathbb{R}^2$ as shown in Figure 2. Two clusters (C_1 and C_2) of nominal data points points are generated using two independent bi-variate Gamma distributions, N_1 and N_2, which are almost triangular in shape. Another two sets of points, which are treated as anomalies are generated using Normal distributions denoted as a_1 and a_2. Let us model the data using Gaussian Mixture Model (GMM) with $k = 2$ components and also with K-Means with $k = 2$. We estimate the anomalousness of a point as the distance from assigned cluster center for the K-Means model and as the likelihood of a data point conditioned on the estimated model parameters for GMM (similar to DAGMM [46]). For the K-Means based model and the GMM based model, we find the average precision(auPR) to be < 0.5 and ≈ 0.8 respectively. Here k is chosen as 2 for the K-means based model and shows a reasonably good performance. However such knowledge is not readily available in complex real-world data. The average precision drops significantly to 0.2 when changing the number of clusters to $k = 1$. A single cluster assumption is not invalid, and made in cases like SVDD [39] and deepSVDD [33]. While methods have been proposed to estimate the optimal number of clusters for a given dataset such as X-means [30] or Bayesian GMM [32], they are known to suffer from scalability issues in complex (high dimensional) and big data models. While for a simple scenario like the one considered above, determining the appropriate number of clusters through tests such as silhouette analysis would be easy — but for sparse high dimensional data such analysis is prone to errors. We can depart from such limiting assumptions if we could approximate a *contrastive* distribution, whose density is high only in regions where the *normal* data is absent. Provided G_1 and G_2 is revealed to us, this can be easily done by sampling from an uniform distribution U over the valid data domain such that $P_{G_1}(u) < \epsilon$ and $P_{G_2}(u) < \epsilon$. Sampling from this distribution, and from the *normal* data, it is possible to create a discriminator function that can predict which region of space a point belongs, or in other words can estimate the probability of a point belonging to region of *normal* data following the concept of NCE [18]. Such a model achieves an average precision ≈ 0.9.

This shows a model that estimates the density of *normal* or expected data using a contrastive distribution can be effectively used for anomaly detection. The key challenge is that in real world data neither is analytical form of data distribution exactly known, nor is it a trivial task to estimate such a contrastive distribution. In the upcoming sections we detail how to overcome this problem and estimate the contrastive distribution through our model.

4 MODEL ARCHITECTURE

Problem Statement: Given a dataset $\mathcal{D}$ containing heterogeneous attributes which are assumed to be clean, learn a model $\mathcal{M}(\theta)$ that can predict $P(x|\theta)$, the likelihood of a test record being drawn from the underlying data distribution. The test records with likelihood below a user-defined threshold are deemed anomalous.

While embedding based architectures have been used in prior works, deep autoencoders preserve richer information in a reduced dimension compared to shallow or linear autoencoders [19]. Use of reconstruction error as anomaly score has the associated caveat that if the autoencoder generalizes too well - it can lead to anomalous samples having low reconstruction error [36]. Thus while we use an autoencoder to capture a reduced dimensional representation for the data, but build a density estimation network to directly estimate the likelihood of data instances

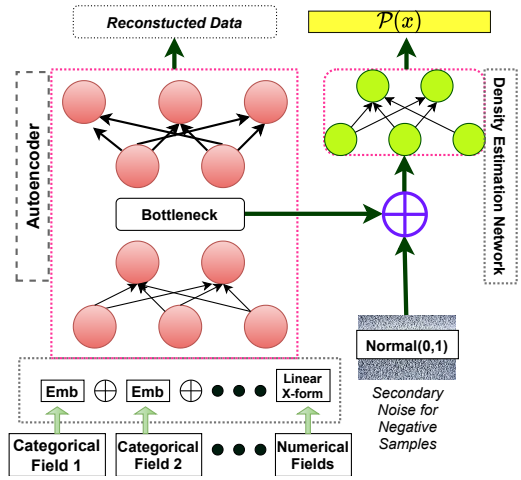

Figure 3: Proposed architecture of our model Contrastive Learning based Heterogeneous Anomaly Detector(CHAD)

4.1 Asymmetric Autoencoder

We propose the use of an asymmetric architecture that is aware of the fields in the tabular input data. The decoder is a fully connected dense neural network with dropout. The first fully connected layer in the encoder is concatenation of embedding(linear transform) for each categorical attribute(domain), and either linear or identity transform for the set of all continuous features. This decision to add the transformation depends on the dimensionality, and we find it helps to add a transform where dimensionality is over 32.

Formally, let $\mathbf{x} \in \mathcal{D}$ be an input. $\mathbf{x}$ is a multidimensional variable, with d features. Let $x_1, x_2 ... x_k$ be categorical features, and $x_{k+1} ... x_d$ be continuous features. $x_{k+1} ... x_d$ can be considered as a multivariate feature, denoted as $x_r \in R^{d-k}$. Let $f_i(.)$ denote the linear transform or the identity transform, for i^{th} categorical field. Let $g(.)$ denote transformation on x_r. Let x_t denote the transformed input, which is input to the fully connected layers of the encoder, as shown in Eq. 1. Both the encoder and decoder utilize dropout and a non-linear activation function(tanh), except for the final output where sigmoid is used.

$$x_t = f_1(x_1) \oplus f_2(x_2) ... \oplus f_k(x_k) \oplus g(x_r) \tag{1}$$

The autoencoder is designed to optimize reconstruction of the input vector, which is an auxiliary task in our case since we are interested in the latent representation. We choose Mean Squared Error for the reconstruction Loss, denoted as $\mathcal{L}_{\mathcal{R}}$ as shown in Eq. 2.

$$\mathcal{L}_{\mathcal{R}} = \frac{1}{N} \sum_i (\mathbf{x} - \hat{\mathbf{x}})^2 \tag{2}$$

While it is possible to have separate losses for categorical and numerical features, we do not find any significant performance difference in doing so. In the next section, we detail out how the latent representation is used in the discriminator module to identify data density.

4.2 Density Estimation Network

The central idea here is to learn the density of data through comparison or contrast with an artificially generated noise distribution, extending the concept of NCE [18]. Let the class labels for the normal data and noise be 1 and 0 respectively. We start with an assumption that $P(C = 1) = P(C = 0) = 0.5$ and denote the empirical distribution of noise as $p_n(.)$. We model the density of data as $p_d(.; \theta)$ where θ are the model parameters. That is, $P(x|C = 1, \theta) = p_d(x; \theta)$ and $P(x|C = 0) = p_n(x)$. Therefore,

$$P(C = 1|x; \theta) = \frac{p_d(x; \theta)}{p_d(x; \theta) + p_n(x)} = f(x; \theta) \tag{3}$$

Here $P(C = 1|x; \theta)$ is the posterior distribution of class label 1, and $f(x; \theta)$ is the approximate estimation function. A MLP can be utilized as functional approximator for $f(x; \theta)$ without loss of generality. Our objective is to find the set of parameters $\hat{\theta}$ for the model that maximize the likelihood of the training data.

$$\begin{aligned} \mathcal{L}(\hat{\theta}) &= \arg\max_{\theta} \sum_i C_i ln(P(C_i = 1|x_i; \theta) \\ &\quad + (1 - C_i) ln(P(C_i = 0|x_i; \theta) \\ &\approx \sum_i ln(f(x_i; \theta)) + ln\big(1 - \frac{1}{|k|} \sum_k (f(z_k; \theta))\big) \end{aligned} \tag{4}$$

In the second line of Eq. 4, z_k refers to the kth negative sample drawn for each observed instance x_i. This is because we do not have access to actual samples from class 0.

Now, we can replace x_i with x_e in Eq. 4, where $x_e = \mathbf{f_{enc}}(x_i)$ and $x_i \in \mathcal{D}$. Here $\mathbf{f_{enc}}$ is the transformation function of the encoder in the autoencoder. Specifically, let $\mathbf{f_{enc}}() : R^d \mapsto R^p$, such that p is dimensionality of latent vector. Also, for each x_i, we draw K negative samples from a noise distribution $\hat{\mathcal{D}}$, denoted as $z_i \in R^d$. Then, $z_e^i = \mathbf{f_{enc}}(z_i)$ with $z_e \in R^p$. We inject secondary noise n to latent representation z_e of each negative sample drawn from a multivariate Normal distribution, $n \sim N(0, \mathbf{I}^p)$, where I^p is a identity covariance matrix. The noise injection is done to increase variation in negative samples and improve model performance. Simplifying the above, we have the following.

$$\mathcal{L}_{est}(\hat{\theta}) = -\gamma \sum_{x_i \in |\mathcal{D}|} ln(f(x_e)) - ln\big(1 - \frac{1}{|k|} \sum_k (f(z_e^i + n_k))\big) \tag{5}$$

Minimizing this loss we can learn the parameters to capture the distribution of normal data. Here γ is a penalty if the model assigns low likelihood to normal(nominal) data instances. It is initially set to 1 and slowly increased to a maximum value in the third training phase. Empirically we find that results are not sensitive to γ.

We utilize a simple two layered MLP with dropout to estimate $f(; \theta)$. It is found to work well in practice, though a more sophisticated estimator network can be designed. The overall loss function to optimize, from Eq. 2 and 5 is as follows.

$$\textbf{Loss} = \lambda \mathbb{1}(\mathbf{t_r}) \mathcal{L}_{\mathcal{R}} + \mathbb{1}(\mathbf{t_e}) \mathcal{L}_{est}(\hat{\theta}) \tag{6}$$

Here $\mathbb{1}()$ is an indicator variable $\in \{0, 1\}$. We explain in Section 4.3 how $\mathbb{1}(\mathbf{t_r})$ and $\mathbb{1}(\mathbf{t_e})$ are set depending on the training phase. The training hyperparameter λ modulates the importance of $\mathcal{L}_{\mathcal{R}}$, and how it is varied is discussed in the training procedure. The overall architecture of our model, **C**ontrastive Learning based **H**eterogeneous **A**nomaly **D**etetctor is shown in Figure 3.

4.3 Model Training

We make use of a three phase training procedure to learn our model. The first phase is termed as burn-in phrase, is training the autoencoder only using reconstruction loss. In this phase the estimator network is not modified. This is essential because we want to first obtain the appropriate latent representation for the data, such that normal(nominal) data can be correctly reconstructed. Referring to Equation 6, $\mathbb{1}(\mathbf{t}_r) = 1$ and $\mathbb{1}(\mathbf{t}_e) = 0$ in this phase. The second phase trains both the autoencoder and the estimator jointly. We notice that the scale of the two losses can be different, we adopt an approach similar to some GANs [17], including the estimator loss only for alternate mini-batches. That is $\mathbb{1}(\mathbf{t}_e) = 1$ for alternate batches, $\mathbb{1}(\mathbf{t}_r) = 1$ being constant. This leads to a stable joint training of the two components, avoiding requirement for a sensitive user provided scaling hyperparameter. Referring to Eq. 6, the scaling hyperparameter λ is set to 1 in first phase.In the second phase it is decayed using $exp(-t)$ where t is the epoch number of the second phase. The third and final phase keeps the encoder parameters fixed. The density estimation network is trained based on the estimator loss, with $\mathbb{1}(\mathbf{t}_e) = 1$ and $\mathbb{1}(\mathbf{t}_r) = 0$.

5 GENERATING NEGATIVE SAMPLES

Generating negative samples is key in contrastive estimation of the data distribution. In the case of heterogeneous data, what can be construed as negative samples is not immediately evident as in other cases such as text or networks. Firstly, negative samples should have adequate variation and not be clustered such that they provide a contrastive background to estimate the data distribution. Secondly, they should not be entirely comprised of noise or be divergent from target distribution, but have some similarity with the normal data – so that negative samples are located within the domain of the nominal data in normal space.

While unexpected combination of entities or values might be considered as negative samples, simply perturbing only the categorical features or only the continuous features might not guarantee the resulting(perturbed) record satisfying the aforementioned criteria. This is because we are unaware of the importance of and interactions among individual features that defines the behavior of data in the latent space. Further, we found merely adding isotropic noise to the latent representation of data does not provide good negative samples. Thus we propose the random subspace based approach outlined in Algorithm 1 for obtaining negative samples.

In this approach at most half of categorical features are selected, and each entity is replaced by another instance belonging to same category. Let a_w be the arity of the w^{th} category. The probability of a category to be selected for perturbation is chosen following a multinomial distribution. The probability of w being selected is the sum normalized value of q_w, where $q_w = (a_w / \sum_{w'} a_{w'})^{0.75}$. The dampening factor 0.75 is added following [26], so that not only the high arity features are perturbed.

Continuous values are assumed to be normalized to be in the range $[0, 1]$. We add uniformly generated random noise to randomly selected $\lfloor r/2 \rfloor$ features, where r is the number of continuous features. The noise deviation parameter δ is used to shift the mean of $U(0, 1)$. Intuitively using $\delta = 0.5$ makes sense, and works well in our case. The range of the noise is set up to be beyond the expected range of the features. This per-feature perturbation forces the feature values for negative samples to spread out over and beyond the range of possible values.

Algorithm 1: Negative sample generation

```
Data: Training records 𝒟; noise deviation δ
Result: Negative samples for each training record
for each training record x ∈ 𝒟 do
    for i = 1:m negative samples do
        Select i ∈ [1, ⌊k/2⌋) categorical features;
        for each feature in i do
            Replace entity with a random entity;
        Select |j1| = ⌊r/4⌋ real features randomly;
        for each feature u ∈ j1 do
            n ~ Uniform(0,1) + δ, u = u + n
        Select |j2| = ⌊r/4⌋ real features randomly;
        for each feature v ∈ j2 do
            n ~ Uniform(0,1) - δ, v = v + n
```

6 MODEL EVALUATION

6.1 Baselines for Comparison

The metric used for evaluation performance of our model against competing baselines is chosen as *average precision*, which is the area under the precision-recall curve following prior works [4, 13] The following baselines are chosen for comparing model performance.

OCSVM [35] is a classification based approach, that determines a decision boundary for normal data. An exponential kernel is used and hyperparameter ν is set to a standard value.

Deep SVDD [33] proposed deep learning based feature extraction combined with a one class classification [39], [35], with two variations of the objective — one-class and soft-boundary. We replace the CNN based feature extractor presented in the original work with a three layered DNN with dropout for general data.

DAGMM [46] combines a deep autoencoder with Gaussian mixture model to perform anomaly detection. It augments the latent data representation with reconstruction losses, and clusters the points into mixture of multivariate Gaussian distributions. The estimated likelihood is used as anomaly score for the samples.

DCN [44] performs dimensionality reduction through a deep autoencoder and K-means clustering jointly. We use greedy layer-wise pretraining for the autoencoder to improve performance, prior to performing joint optimization. The distance of a point from its assigned cluster center is used as the anomaly score.

Deep Auto Encoder [9]: A deep autoencoder with dropout with greedy layer-wise pretraining is used, and reconstruction loss is used as anomaly score. Previous works [42] point out that successive layers of dropout achieve a denoising effect.

FAE-r: We use field aware auto-encoder from our CHAD with reconstruction loss as anomaly score, without any modification such as greedy layer-wise pretraining.

6.2 Experimental Results on Intrusion Detection Data

To better understand the performance of our model on we choose open source intrusion detection datasets, which are as follows:

Data Set	OCSVM	FAE-r	DAE	DCN	DAGMM	dSVDD	CHAD
KDDCup99	**0.99039** (± 0.0012)	0.96113 (± 0.0371)	0.76297 (± 0.1076)	0.89324 (± 0.0447)	0.79432 (± 0.2015)	0.68164 (±0.1407)	0.97332 (± 0.0158)
KDDCup99-N	0.99999 (± 0.0001)	0.58391 (± 0.0002)	0.99957 (± 0.0004)	0.94701 (± 0.0312)	0.99695 (± 0.0025)	0.97843 (± 0.0046)	0.99647 (± 0.0035)
UNSW-NB15	0.38445 (± 0.0055)	0.65070 (± 0.0379)	0.26121 (± 0.0498)	0.66786 (± 0.0785)	0.34278 (± 0.1980)	0.74479 (± 0.1144)	**0.78873** (± 0.0032)
NSL-KDD	0.84869 (± 0.0059)	**0.88895** (± 0.0316)	0.81286 (± 0.0329)	0.53152 (± 0.1574)	0.36002 (± 0.1440)	0.60989 (± 0.0729)	0.75512 (± 0.0517)
Gure-KDD	0.68652 (± 0.0063)	0.67999 (± 0.0229)	0.68553 (± 0.0433)	0.37014 (± 0.0981)	0.35806 (± 0.0834)	0.63064 (± 0.0719)	**0.75680** (± 0.03918)
Average Score	0.78201	0.75294	0.70443	0.68195	0.56960	0.72908	**0.85409**

Table 1: Performance comparison w.r.t to average precision (auPR). On average CHAD outperforms all models with an improvement of over 9.2% over second best model. Though CHAD is not the best in KDDCup99 and NSL-KDD datasets it does provide competitive results. For KDDCup99-N, we find there is no single definitive best performing model. Experimentally we find that unlike CHAD the other models are highly sensitive to choice of hyper-parameters (refer Fig. 4) and in absence of labelled data /domain knowledge making a good choice for hyper-parameters will be difficult.

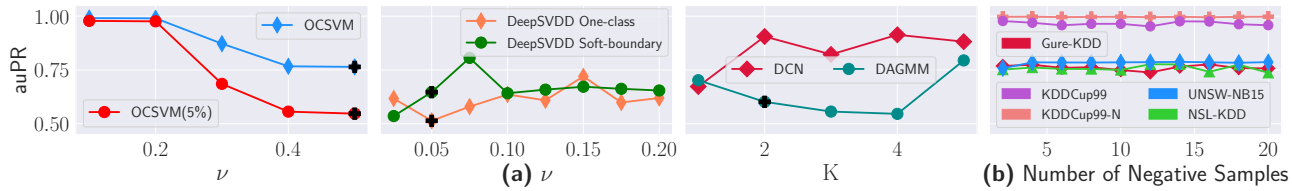

Figure 4: (a) Performance variation of competing methods to changing key hyper-parameters, for anomaly detection on KDDCup99 data. The standard or recommended hyperparameter settings are highlighted. (b) Effect of varying number of negative samples per instance on model performance for CHAD for all datasets.

KDDCup99 [14]: The KDDCup99 10 percent dataset has been a benchmark dataset for anomaly detection tasks. We choose the 'Normal' class label as normal data and attack classes are treated as anomalies.

KDDCup99-N [14]: We consider the instances KDDCup99 10 percent data with attack class 'Neptune' to be normal data, and data points with label 'Normal' are considered as be anomalies.

UNSW-NB15 [27]: We select data with label *Normal* as normal data, and classes *Backdoor*, *Analysis*, *Shellcode*, *Worms* as anomalies.

NSL-KDD [38]: The combined train and test partitions of the original dataset is used, with 'Normal' class chosen as normal data and rest as anomalies.

GureKDD [31]: The 'Normal' class label is chosen as normal data, and rest as anomalies.

For all models, we experiment with various choices for the respective hyper parameters and report the best score. We refer the reader to Section 6.5 for more details. The experimental results are shown in Table 1. We follow the same experimental protocols followed for the trade data. For our model CHAD, we use 10 negative samples for training the density estimation network. We use Adam [21] for optimization for all methods that require it, with learning rate to 5×10^{-3} and batch size 256 or 512 across all datasets. The number of epochs for the three training phases are set to 50,10 and 25 respectively. We use a generic 3-layered pyramid-like architecture for the autoencoder, with a dropout of 0.2.

In judging average performance we find our model CHAD performs better than the baselines. OCSVM provides comparable performance in some cases. However, it does not perform well for UNSW-NB15 dataset which can be attributed to either nature of the data and higher dimensionality caused by comparatively larger entity count. DAE and DCN also performs well in most cases. DAGMM however does not perform well despite its expected good performance on general data. We observe a high variance, and it can be attributed to the fact that loss metrics augmenting the latent space does not necessarily improve model expressiveness. FAE-r is shown to perform well in many cases, which demonstrates that our autoencoder captures the latent representation well. However, it performs poorly on KDDCup-N, which can be attributed to over generalization issue mentioned earlier. We find our model performs comparably or favorably in majority of cases. Our model does not perform the best for NSL-KDD, however in case of UNSW-NB15 and Gure-KDD it shows a strong advantage. However it is to be noted for baselines such OCSVM, DAGMM and DCN we experiment with different hyper-parameters using a holdout of test samples, which

Trade Dataset	Total cardinality	Numeric features	Train size	Test size
Dataset-1	809	4	90910	57772
Dataset-2	853	4	106858	57213
Dataset-3	850	4	96545	51189
Dataset-4	859	4	108841	56298
Dataset-5	844	4	109525	50481
Dataset-6	877	4	113692	56432

Table 2: Details of the trade datasets used for experimental evaluation. Total cardinality here refers to the sum of the cardinality of the categorical attributes.

Data Set	OCSVM	FAE-r	dSVDD	CHAD
Dataset-1	0.931139 (± 0.0003)	0.562055 (± 0.0169)	0.630166 (± 0.0964)	**0.982815** (± 0.0053)
Dataset-2	0.170002 (± 0.0400)	0.577034 (± 0.0149)	0.536559 (± 0.0652)	**0.996048** (± 0.0010)
Dataset-3	0.943908 (± 0.0001)	0.597657 (± 0.0159)	0.681765 (± 0.0839)	**0.992471** (± 0.0019)
Dataset-4	0.997622 (± 0.0001)	0.989836 (± 0.0031)	0.755632 (± 0.0209)	**0.999369** (± 0.0002)
Dataset-5	0.319723 (± 0.0010)	0.939353 (± 0.0119)	0.907916 (± 0.0001)	**0.993936** (± 0.0013)
Dataset-6	0.933144 (± 0.0002)	0.576750 (± 0.0111)	0.712136 (± 0.0420)	**0.986193** (± 0.0026)

Table 3: Performance (average precision) comparison of CHAD and baselines on trade datasets. CHAD shows a superior and consistent performance across all datasets.

is not possible in real-world scenarios. In Section 6.5 we further examine and demonstrate how competing models are sensitive to bad choices of hyper-parameters.

6.3 Trade Data and Synthetic Anomalies

For experimental evaluation of our model, we use a subset of the real world trade data that is part of our proposed framework. Six subsets are selected, where the training set is two months of data and the subsequent month is treated as the test set. Pre-processing steps includes removal of records with missing data and omitting entities which are rare using a count threshold, following prior works [11]. The details of the subsets are shown in Table 2

While generally evaluation of anomaly detection is performed using labelled data, considering one or more minority classes as anomalies, we do not have labels for trade data and thus generate synthetic anomalies following prior works [10]. For trade data, synthetic anomalous records are generated from the test set. Specifically, one categorical and one numeric attribute chosen at random are perturbed. For the categorical attribute, it is replaced with a randomly value(entity) belonging to the same field(domain). For the continuous valued attribute, a perturbation value (p) is added. If the current value is less than 0.5, p is chosen from Uniform(0.25,0.75) and if the current value is greater than 0.5 p is chosen from Uniform(-0.25,-0.75). This approach is adopted to ensure that the records have unexpected patterns, characterized by the pair of perturbed attributes. Additionally anomalies generated should be not be trivial and easy to detect, for a proper evaluation of the model. This approach however is distinct from the way we generate negative samples, thus resulting in a fair comparison.

6.4 Experimental Results on Trade Data

We compare our model's performance against top-3 best performing models obtained from the evaluation above. The model architecture is kept same, with similar values of dropout and a three layered autoencoder. For all the autoencoder based approaches, same network architecture is used, specifically 3 layered fully connected network with dropout to obtain the latent representation. The same hyperparameter settings for a particular model are used. The percentage of anomalies is set to 20% instead of a balanced distribution, we use 5 sets of randomly sampled anomalies for each instance of model evaluation. The results are shown in Table 3. Our model is shown to outperform the baselines consistently. This is significant for our purposes, since our usage scenario lacks any labelled data to perform hyperparameter tuning.

6.5 Comparing the Effect of Hyperparameters

Sensitive hyperparameters, which cannot be determined apriori without a validation set [28], have a significant effect on model performance. We empirically show that competing models are highly sensitive to hyper-parameter values. The experiments are performed on the KDDCup-99 dataset where all models perform well. Case in point, for OCSVM, although default value of ν is 0.5 is used in popular implementations and in the original work, we find setting it to 0.1 obtains best results. Further with a smaller number of anomalies to be detected, the hyperparameter setting affects performance more drastically as shown in Figure 4(a). In case of deepSVDD authors recommend ν between 0.01 and 0.1, but the model performance varies significantly between those bounds, and beyond that. Moreover while the one-class objective is shown to consistently perform better in the original work, this is not always true as shown in Figure 4(a). In case of trade data soft-boundary objective works best for seepSVDD. Similarly for DCN and DAGMM, the number of clusters has a significant effect on performance. In training our model with negative samples, the count of negative samples per instance is the most significant hyper-parameter. We train our model with varying number of negative samples per instance, and perform evaluation for all the data sets as shown shown in Figure 4(b). For the latter three datasets model performance is found to increase slightly with number of negative samples, though lower number of samples does not adversely affect the model performance. Thus our model is robust to choice of hyperparameters.

6.6 Varying Anomaly Content

Given anomalies can be rare, it is important to understand how our model performs with varying percentage of anomalies. For each benchmark data set, we vary the percentages of anomalies in the combined test set from 2% to 10%, to understand how our model performs in each scenario. The results are reported in Table 4. As

Data Set	2%	4%	6%	8%	10%
KDDCup99	0.9665	0.9705	0.9740	0.9739	0.9757
KDD-N	0.9941	0.9958	0.9964	0.9967	0.9965
NB15	0.7472	0.7630	0.7747	0.7834	0.7887
NSL-KDD	0.6498	0.7042	0.7254	0.7401	0.7499
Gure-KDD	0.6905	0.7232	0.7357	0.7481	0.7560

Table 4: Average Precision score of CHAD for detecting anomalies, with different percentages of anomalies in the test set. CHAD is able to effectively detect even a small percentage(ratio) of anomalies.

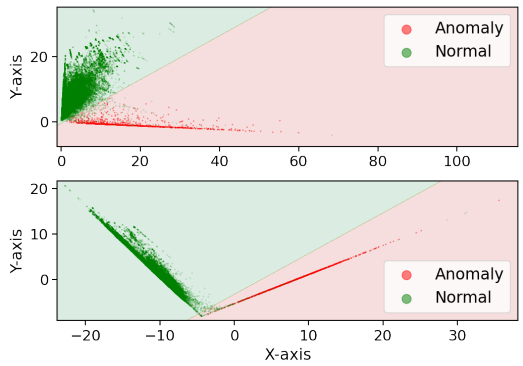

Figure 5: Visual representation of latent features for anomalies and nominal data points at different layers of the model. The top figure shows latent features obtained from the bottleneck layer of the asymmetric autoencoder, while the lower one shows features from penultimate layer of the density estimation network, both projected in a 2D space. With successive layers, we see the nominal and anomalous points are more distinctly separated.

expected, it is a more challenging task to detect anomalies when they are rarer; and our model performance improves slightly as the percentage of anomalies increase. However it is also evident that CHAD is effective in cases with very low anomaly percentages.

7 MODEL ANALYSIS

7.1 Visualizing Anomalies

The utility of autoencoders in anomaly detection model architectures is to perform feature extraction and embed the data in a low dimensional space, where nominal points are clustered or are separable from anomalies. To ascertain this, we perform visual analysis on feature vectors obtained from the lowest dimensional output of the autoencoder and the penultimate layer of the density estimation network as shown in Figure 5. We sample points from both the nominal data and anomalies, and perform dimensionality reduction to R^2 using SVD. For both the cases, we find that the nominal data is separable from the anomalies using a linear decision boundary in the transformed space. This fortifies our hypothesis, and the model is able to separate nominal data instances from anomalies in the decision space.

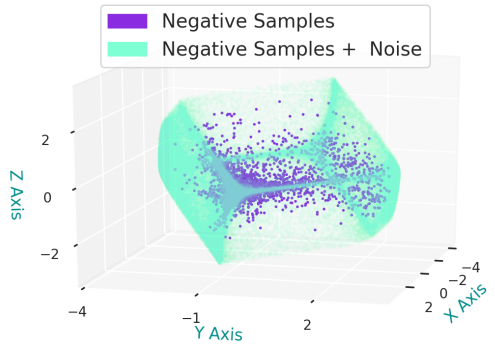

Figure 6: Visualization of latent vectors of negative samples for KDDCup99 dataset, with and without injected noise. The addition of isotropic Gaussian noise results in a greater variation(spread) of negative samples across the latent space.

Secondary noise	KDD	KDD-N	NB15	NSL-K	Gure-K
Yes	**0.9723**	0.9973	0.7874	**0.7682**	**0.7662**
No	0.9325	0.9974	0.7887	0.6983	0.6943

Table 5: Comparison of average precision scores when CHAD is trained with and without the secondary instance noise for negative samples. The secondary noise leads to significant overall improvement in model performance.

7.2 Importance of Secondary Noise

We hypothesise that adding a secondary instance noise to negative samples at the latent space of the autoencoder helps in building a better model through a better approximation of the contrastive latent space, and our experiments confirms this. This isotropic Gaussian noise helps prevent over-fitting while training the model, which is crucial without validation sets in this unsupervised training scenario. This is evidenced by the fact that omitting the instance noise in some training runs, although the loss converges to a lower value the test time performance is lower. To better understand this, we visualize the negative samples generated for the dataset KDD-Cup99 with and without this noise, as shown in Figure 6. It can be observed adding noise spreads out the latent representation of the negative samples over data space more evenly and across a larger region in the domain space of the nominal data. This effectively helps better estimate the space where nominal data does not occur. The model thus gets trained with negative samples with a significantly greater variation and is more robust. Table 5 shows the improvement in average precision provided by the addition of this secondary noise, which is more pronounced when only few negative samples per instance are used.

8 FUTURE DIRECTIONS AND CONCLUSION

The anomaly detection method proposed in this work is part of a larger effort to build a framework for the real world task of detecting suspicious timber shipments. Although prior methods have been shown to be effective on benchmark datasets, there remain several

shortcomings that impede their application in real world systems with heterogeneous tabular data. We show our model CHAD can generalize to different data distributions and is robust to changes in hyper-parameters, and performs favorably against competing baselines. Our model has the potential to be applicable beyond timber shipment records, and could be applied to shipment records for any globally traded commodity, as well as be applicable to any heterogeneous tabular data occurring in other domains, as demonstrated by our comprehensive evaluation.

There are however further research questions we would like to address. First relates to the issue of relevancy of anomalies, since only a subset of anomalies are relevant with respect to the application scenario. Improving relevancy of anomalies through active learning or a human-in-the-loop is a logical research direction that follows. The second research question is to find ways of presenting the user with possible explanations as to why a record is judged anomalous. Explainability [24] can be a powerful tool in garnering deeper understanding and trust of end users. While we can visualize anomalies in latent space as shown in Section 7.1, it does not provide a succinct reasoning. These challenges are applicable to our work in building the framework to detect suspicious timber shipments. Further research questions pertaining to temporal aspects of trade data and understanding feature importance are also of consequence for our application. Therefore, there are multiple research directions as part of building our framework as well more general problems that stem from this work. These questions motivate multiple continuing future research directions for us.

Acknowledgments This paper is based on work supported by the NSF (DGE-1545362).

REFERENCES

[1] 2018. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360* (2018).

[2] Aisha Abdallah, Mohd Aizaini Maarof, Anazida Zainal, et al. 2016. Fraud detection system: A survey. *Journal of Network and Computer Applications* (2016), 90–113.

[3] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery* 29, 3 (2015), 626–688.

[4] Leman Akoglu, Hanghang Tong, Jilles Vreeken, and Christos Faloutsos. 2012. Fast and reliable anomaly detection in categorical data. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 415–424.

[5] Amanda Berg, Jörgen Ahlberg, and Michael Felsberg. 2019. Unsupervised learning of anomaly detection from contaminated image data using simultaneous encoder training. *arXiv preprint arXiv:1905.11034* (2019).

[6] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 93–104.

[7] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).

[8] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys* 41, 3 (2009).

[9] Jinghui Chen, Saket Sathe, Charu Aggarwal, and Deepak Turaga. 2017. Outlier detection with autoencoder ensembles. In *SDM*. 90–98.

[10] Ting Chen, Lu-An Tang, Yizhou Sun, Zhengzhang Chen, and Kai Zhang. 2016. Entity Embedding-based Anomaly Detection for Heterogeneous Categorical Events. In *IJCAI*. 1396–1403.

[11] Kaustav Das and Jeff Schneider. 2007. Detecting anomalous records in categorical datasets. In *KDD*.

[12] Debanjan Datta et al. 2020. Detecting Suspicious Timber Trades.. In *Thirty Second IAAI Conference*.

[13] Jesse Davis and Mark Goadrich. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*. 233–240.

[14] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[15] M. Goldstein and S. Uchida. 2016. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one* 11, 4 (2016).

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems*. 5767–5777.

[18] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*.

[19] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* 313, 5786 (2006), 504–507.

[20] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity* 2, 1 (2019), 20.

[21] Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. (2015).

[22] Longin Jan Latecki, Aleksandar Lazarevic, and Dragoljub Pokrajac. 2007. Outlier detection with kernel density functions. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*. Springer, 61–75.

[23] Sam Lawson and Larry MacFaul. 2010. Illegal logging and related trade. (2010).

[24] Zachary C Lipton. 2018. The Mythos of Model Interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue* 16, 3 (2018), 31–57.

[25] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. 2008. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 413–422.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[27] N. Moustafa and J. Slay. 2015. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*. 1–6.

[28] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in neural information processing systems*. 3235–3246.

[29] Panjiva. 2019. Panjiva Trade Data. https://panjiva.com.

[30] Dan Pelleg, Andrew W Moore, et al. 2000. X-means: Extending k-means with efficient estimation of the number of clusters.. In *Icml*, Vol. 1. 727–734.

[31] Iñigo Perona et al. 2008. Service-Independent Payload Analysis to Improve Intrusion Detection in Network Traffic. In *Proceedings of the 7th Australasian Data Mining Conference*. Australian Computer Society, Inc., AUS.

[32] Carl Edward Rasmussen. 2000. The infinite Gaussian mixture model. In *Advances in neural information processing systems*. 554–560.

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

[34] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. Association for Computing Machinery, New York, NY, USA, 4–11. https://doi.org/10.1145/2689746.2689747

[35] Bernhard Schölkopf et al. 2000. Support vector method for novelty detection. In *Advances in neural information processing systems*. 582–588.

[36] Giacomo Spigler. 2019. Denoising autoencoders for overgeneralization in neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 4 (2019), 998–1004.

[37] Takaaki Tagawa, Yukihiro Tadokoro, Takehisa Yairi, et al. 2015. Structured denoising autoencoder for fault detection and analysis. In *Asian Conference on Machine Learning*. 96–111.

[38] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A Ghorbani. 2009. A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*. IEEE, 1–6.

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

[40] Yogesh Virkar, Aaron Clauset, et al. 2014. Power-law distributions in binned empirical data. *Annals of Applied Statistics* 8, 1 (2014), 89–119.

[41] A. C. Wiedenhoeft et al. 2019. Fraud and misrepresentation in retail forest products exceeds US forensic wood science capacity. *PloS one* 14, 7 (2019).

[42] Junyuan Xie, Ross Girshick, Ali Farhadi, et al. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*. 478–487.

[43] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. 2010. Robust PCA via outlier pursuit. In *Advances in neural information processing systems*. 2496–2504.

[44] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. 2017. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International Conference on Machine Learning*. 3861–3870.

[45] Chong Zhou and Randy C Paffenroth. 2017. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 665–674.

[46] Bo Zong et al. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*.