An Anomaly Detection Framework Based on Autoencoder and Nearest Neighbor

Jia Guo, Guannan Liu, Yuan Zuo, Junjie Wu
School of Economics and Management, Beihang University, Beijing, China
corresponding author: liugn@buaa.edu.cn

***Abstract*—In recent years, anomaly detection has become a focal point of data mining, and numerous efforts have been made to conduct extensive researches on the theories and techniques for detecting abnormal data points. Although the amount of anomaly data is relatively small, they can potentially bring huge losses to social economy, public resources and individual properties. Thus, we propose an unsupervised anomaly detection framework named AEKNN, which aims to incorporate the advantages of automatically learnt representation by deep neural network to boost anomaly detection performance. The framework combines the training of an autoencoder and a k-th nearest neighbor based outlier detection method. We further validate the performance of our proposed model with an extensive experimental study on three UCI datasets. The parameter sensitivity results demonstrate that the proposed algorithm can scale well with respect to both dataset size, data feature dimensionality and anomaly class proportion.**

***Index Terms*—Autoencoder, Anomaly Detection, Representation Learning, Nearest Neighbor**

I. Introduction

With the widespread application of voluminous data in various industrial scenarios, extracting valuable information from massive trivial data has become a huge challenge. Sometimes, there exist a few abnormal points in the dataset, which deviate much from the normal data and are very crucial for management decisions. For example, in the financial area, an abnormal transaction can cause millions of dollars loss if it isn't spotted out timely. Thus, anomaly detection is a very important and meaningful research field of data mining and has received a lot of attention in both academic and industrial domains. In practical applications, the abnormal data is often hidden in a large amount of daily normal data, such as financial transaction records, telecommunication calls, etc.

For anomaly detection, it's of great importance to decide which data features to be used when representing data points. Effective characteristic can dramatically improve the performance of anomaly detection. Otherwise, it will cause a waste of labour and time. Representing learning, also known as feature learning, is an important research issue in the field of machine learning. Its goal is to automatically learn a transformation from the original input data to the new feature representation. The learnt feature representations should be able to apply to various tasks effectively. By conducting this learning method, people can be liberated from the tedious feature engineering works.

Recently, deep neural network based representation learning has become a research hotspot, which aims to better explore the implicit feature information from the original data. By utilizing the deep network structure to extract non-linear and high-level characteristics from the raw data features effectively, it is able to form a more differentiated and compact data representation. Furthermore, it has been shown that the learnt representation is well adapted to various data mining tasks such as natural language processing, computer vision and anomaly detection. In order to enhance the efficiency of feature analysis and mining from large-scale high-dimensional data and realize its practical application in the field of anomaly detection, an unsupervised framework for anomaly detection named AEKNN is proposed in this paper. The framework AEKNN integrates an autoencoder model and a k-th nearest neighbor based outlier detection method to automatically learn suitable deep data representations for discriminating anomalies from the normal class.

The rest of this paper is organized as follows. Section II describes the details of our proposed framework. Experimental setup is presented in Section III. Section IV explains our experiments and results. Section V reviews the widely applied anomaly detection methods and the variants of autoencoder model. Section VI concludes this paper.

II. The Proposed Model

In this paper, we propose an unsupervised anomaly detection framework named AEKNN, which integrates an autoencoder model and a k-th nearest neighbor based outlier detection method [1]. The main idea of AEKNN is to utilize the deep feature representation ability of autoencoder, and then improve the detection performance of the traditional outlier detection method. Our framework contains two stages, the training stage and testing stage. In the training stage, an autoencoder model will be trained on normal data, and then the compressed hidden layer vector which is regarded as the deep feature representation of the original data will be used to train a k-NN based anomaly detection method. In the testing stage, we validate the detection performance of our method on a contaminated dataset which contains both normal data and abnormal data. We introduce the details of our proposed framework as follows.

978-1-5386-5178-0/18/$31.00 ©2018 IEEE

A. Learning Sparse Representations

Autoencoder is a multi-layer forward neural network which plays an important role in the field of deep learning. The neural network is learnt in an end-to-end way through a large number of high-dimensional data points. As an unsupervised learning method, autoencoder designs the encoding and decoding processes to minimize the reconstruction error between the input vector and output vector. During the training process, the model can project the high-dimensional data to a more compact vector which would be automatically learnt and effectively represents the latent characteristic of input data.

The motivation of adopting this method is that autoencoder is data-dependent, which means that we train an autoencoder model on a normal dataset, and the trained model can only compress data that is similar with the training data, i.e., the normal class. If part of data is unusual or the testing dataset is mixed with contaminated data, the reconstruction error will rise up, which indicates that there are anomalies, and the corresponding changes will also reflect in the compressed hidden representation vector.

As Fig. 1 shows, from left to right, the first layer of autoencoder is to receive the original data as the initial network input, and the last layer of network is to reconstruct the original data vector. The middle layers are several continuous shallow hidden layers, which could extract highly nonlinear feature representation vector of the original data. The autoencoder generally involves two main phases, which are the encoding phase and the decoding phase.

In the encoding phase, the original data are continuously compressed by several hidden layers to a more compact vector compared with the initial input vector.

$$x_1 = E_1(W_{e1}x_0 + b_{e1}) \tag{1}$$

$$x_2 = E_2(W_{e2}x_1 + b_{e2}) \tag{2}$$

$$\vdots$$

$$g(x_0) = E_n(W_{en}x_{n-1} + b_{en}) \tag{3}$$

where x_0 represents the original data input vector, and E_i represents the encoding activation function of each layer. In this paper, we adopt the sigmoidal nonlinearities function as the activation function. $x_i (i \neq 0)$ denotes the data vector activated by each network layer. $g(x_0)$ denotes the deepest hidden representation vector. W_{ei}, b_{ei} are weight parameters and the corresponding biases of each layer respectively.

In the decoding phase, the deepest hidden representation vector is gradually decoded by several network layers, and in the last layer of network, the model outputs a reconstruction vector of the initial input.

$$\hat{x}_1 = D_1(W_{d1}g(x_0) + b_{d1}) \tag{4}$$

$$\hat{x}_2 = D_2(W_{d2}\hat{x}_1 + b_{d2}) \tag{5}$$

$$\vdots$$

$$\hat{x_n} = D_n(W_{dn}\hat{x}_{n-1} + b_{dn}) \tag{6}$$

where $\hat{x}_j$ represents the decoding vector of the j-th decoding layer, and D_j represents the activation function of each decoding layer. We also adopt the sigmoidal nonlinearities function as the activation function in the decoding phase. W_{dj}, b_{dj} are weight parameters and the corresponding biases of each decoding layer respectively. $\hat{x}_n$ is the final output vector of the network, which is also the reconstruction vector of x_0.

In general, the reconstruction error (RE) of autoencoder is commonly measured by mean square error (MSE) in training neural network. However, in our application scenario, the autoencoder is used to automatically learn features from unlabeled data and give better characterizations than raw data for anomaly detection. So, to enhance the feature learning ability of autoencoder model, we impose a weighted L1 regularization to the loss function, which can generate a sparse hidden vector and further optimize the feature representations.

$$J(\theta; x_0) = \frac{1}{m} \|x_0 - \hat{x}_n\|^2 + \lambda \|g(x_0)\|_1 \tag{7}$$

where λ denotes the regularization scale, and m is the size of training sample. The loss function is composed of two parts, i.e. the reconstruction error and the scaled L1 regularization. The parameters of the network $\theta = \{W_{ei}, b_{ei}, W_{dj}, b_{dj}\}$ are all optimized by minimizing the loss function $J(\theta; x_0)$.

B. k-NN Based Anomaly Detection

After training of autoencoder model, we can obtain the hidden representation vector in the middlemost network layer which represents the deep non-linear feature characteristic of the original data. To better manifest the distinctiveness between the normal data and anomalies, we use the hidden representation vector to train an anomaly detection model instead of the original data, and we adopt the k-th nearest neighbor based outlier detection model in our proposed framework. The k-th nearest neighbor based outlier detection model [1] adopted a novel formulation by utilizing the distance of a data point from its k-th nearest neighbor, and assert the top n points in the ranking list as the outliers. Ramaswamy et. al. [1] have also developed a highly efficient partition-based algorithm for distinguishing outliers, and validated the experimental effectiveness on both real-life and synthetic data sets.

C. Training Algorithm

Algorithm 1 illustrates the training process of our algorithm. In this paper, we adopt the Adam [2] optimization algorithm to compute the training error and train our neural network. The Adam optimization algorithm is an extension of the stochastic gradient descent (SGD) algorithm. Recently, it is widely used in deep learning applications, such as computer vision and natural language processing.

III. Experimental Setup

A. Dataset Description

In our experiments, we adopt three UCI datasets of different size, which are named MNIST, cardiotocography (cardio) and

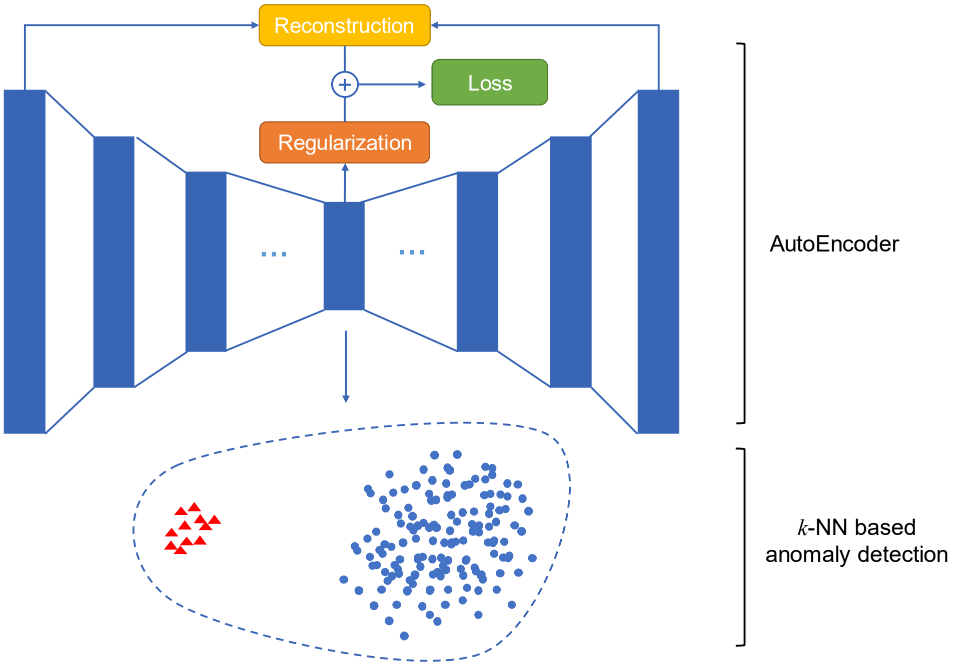


Fig. 1: Illustration of the proposed framework AEKNN

Algorithm 1 Training algorithm for AEKNN

```
Input: Training batch of dataset: $\mathcal{D}$; The parameters: $\theta = \{W_{ei}, b_{ei}, W_{dj}, b_{dj}\}$; The maximum number of epochs: $epoch$; The size of mini-batch: $batch$; The regularization scale $\lambda$; The learning rate $lr$.
Output: The trained model AEKNN
prepare the training batch of samples $\mathcal{D}$
initialize the parameters $\theta$
for $k \in \{1, 2, \cdots, epoch\}$ do
    for $q \in \{1, 2, \cdots, batch\}$ do
        for each $x \in \mathcal{D}$ do
            Compute the input vector and output vector for each layer by Eq. 1 to Eq. 6
            Compute the training error by Eq. 7
            Update the parameters $\theta$ by Adam
        end for
    end for
end for
Train $k$-NN based anomaly detection model using hidden representation vectors $\{g(x)\}$
return The trained model AEKNN
```

mammography separately. The datasets used in our experiments are downloaded from the website of Outlier Detection DataSets (ODDS) [3]. We choose one class from each dataset as normal class and other minor class is considered as the anomaly class. The detail information of datasets are illustrated as Table. I.

TABLE I: Description of experimental datasets

Datasets	#points	#dimensions	#outliers(%)
Cardio	1831	21	9.6%
MNIST	7603	100	9.2%
Mammography	11183	6	2.32%

B. Baseline Methods

- One-class support vector machines (OCSVM) [4]: It learns a close contour of the known normal data points by kernel function and scalar parameter. If observations lie outside the class contour, they would be regarded as anomalies. In our experiments, we choose the RBF kernel as kernel function.
- Robust covariance (RC) [5]: It assumes that normal data are generated from a known distribution, such as Gaussian distributed, and it discovers outliers if the data point deviates much from the fitted data distributions.
- Isolation forest (iForest) [6]: It is an ensemble-based and efficient anomaly detection method with linear time complexity and high precision. It finds out outliers by randomly splitting the high-dimensional data feature space with hyperplanes.
- Local Outlier Factor (LOF) [7]: It measures the local variation density of a given data point to the local densities of its neighbors. If the local density of a data point is significantly lower than its neighbors, then it would be considered as an anomaly.
- Histogram-based Outlier Score (HBOS) [8]: It first constructs an univariate histogram for each data feature

which are viewed as an estimation of feature density, and then all histograms are used to compute the anomaly score for each data point.

- k-NN based Outlier Detection (KNN) [1]: It adopts a novel formulation by utilizing the distance of a data point from its k-th nearest neighbor, and assert the top n points in the ranking list as the outliers.

C. *Parameter Settings*

In this experiment, we set the mini-batch size of training network as 256, 512 and 1024 for three different size datasets respectively. We adopt Adam [2] to compute the training error and optimize our algorithm. The learning rate of Adam algorithm is set as 0.003. The maximum number of training epochs is set as 300. The regularization scale is set as 0.003.

IV. Experimental Results

In this section, we conduct extensive experiments on different data settings to compare the anomaly detection performance of our proposed framework AEKNN with several state-of-art anomaly detection methods. All methods are trained on normal data, and are further tested on dataset which is composed of both normal data and anomalies. We also study the sensitivity of an important parameter named regularization scale λ, which is used to control the penalty term on the sparsity of hidden feature representation vector in our loss function.

A. *Comparison of Different Anomaly Detection Methods*

In this subsection, we choose three various datasets in terms of the size of dataset, the dimensions of data features and the proportion of outliers in the testing dataset to comprehensively validate the effectiveness of our proposed method. As the results of Fig. 3, Fig. 2 and Fig. 4 show, our proposed method outperforms all baseline methods on three data settings.

For small size dataset, i.e. the cardio dataset, we can see that the most methods achieve good performance on this data setting from Fig. 2, for that auc values are all above 0.85, except the HBOS algorithm, which fails to defeat other methods. The performances of LOF and OCSVM algorithm are very close, and both are better than the KNN based anomaly detection and RC algorithm. The detection performance of iForest algorithm achieves the best among all baseline methods, however, our proposed model AEKNN still outperforms it by almost 0.02 in terms of AUC, which well demonstrates effectiveness of the design of our framework.

For medium size and high-dimensional dataset, i.e. the MNIST dataset, the LOF algorithm performs best among all baseline methods, whereas our AEKNN model with the AUC of 0.97 still defeats all baseline methods. From Fig. 3, we can find that the AUCs of KNN method and OCSVM method are both scored as 0.5, which means that they fail to distinguish the anomalies from normal data. However, our method can observably improve the detection performance compared with the KNN method, which further testifies the effectiveness of intruducing autoencoder to learn the deep hidden representations of original data.

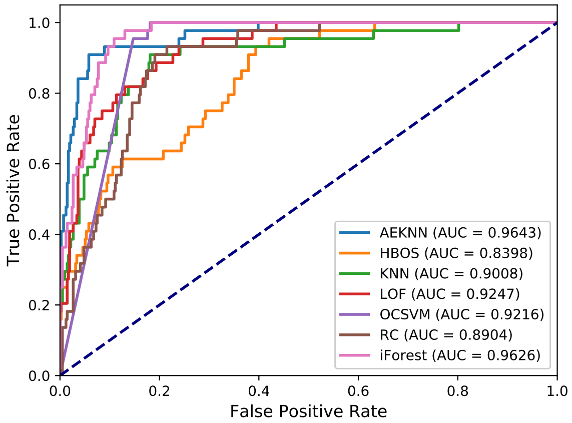


Fig. 2: The performance of anomaly detection on cardio dataset

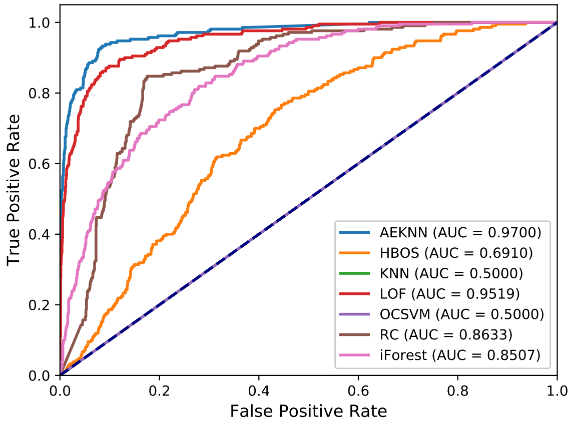
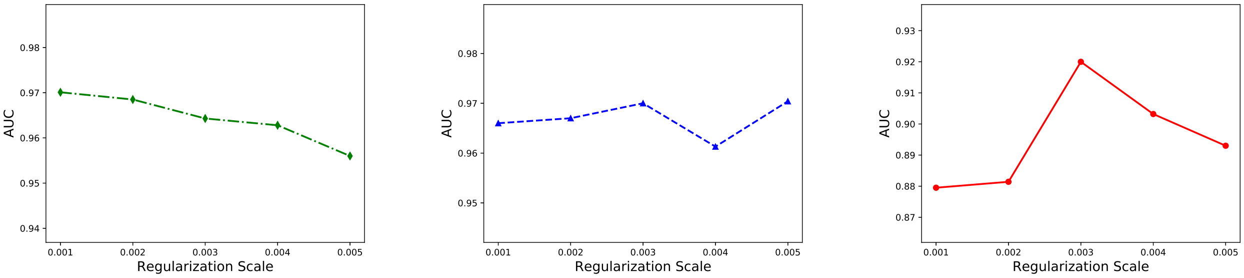


Fig. 3: The performance of anomaly detection on MNIST dataset

For large size and small anomaly propotion dataset, i.e. the mammography dataset, as Fig. 4 shows, we can see that the overall performance of baseline methods has declined compared with the examination on above two datasets. Nevertheless, our proposed model still achieves 0.92 in terms of AUC, and outperforms the best baseline method iForest by 0.03, which further validates the stable and superior property of our proposed method.

B. *Parameter Sensitivity*

In this subsection, we vary the regularization scale λ from 0.001 to 0.005 to test the sensitivity of this parameter. Specifically, as illustrated in Fig. 5, we report the AUC of our proposed method AEKNN. From the results of AEKNN, we can see that λ affects the AUC differently on three datasets. For example, on mammography and MNIST dataset, the AUC of AEKNN first increases along with the λ, and then begins



(a) cardio dataset

(b) MNIST dataset

(c) mammography dataset

Fig. 5: Paramenter sensity

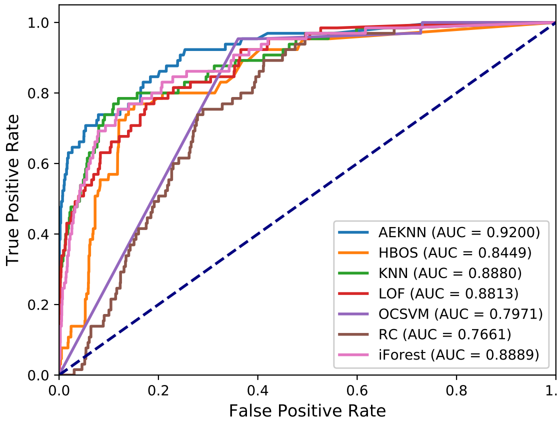


Fig. 4: The performance of anomaly detection on mammography dataset

to fall when λ equals to 0.003. In contrast, the AUC of AEKNN begins to decrease as the value of λ increases on cardio dataset. In general, our method is relatively stable and robust in different settings of λ for that the results fluctuate only in a small range.

V. Related Work

A. Anomaly Detection

Anomaly detection is an important branch of data mining, and numerous scholars in this field have conducted extensive research on both the theory and techniques of anomaly detection. In fact, researchers have studied various types of detection methods for different types of data and fraudulent behaviors. Chandola et al. [9] provided a more comprehensive review of this issue. According to the availability of data labels, anomaly detection problems can often be divided into three situations: supervised, unsupervised, and semi-supervised. We summarize several main research situations as follows.

Under supervised conditions, anomaly detection can be structured as a two-class problem based on whether the known data is abnormal or not. By extracting the characteristics of the data, various classification models are further adopted. Song et al. [10] proposed Robust Support Vector Machines (RSVM) which was robust to calculate the adaptive margin between each data point and the center of class, and improved the generalization performance significantly. Some rule-based techniques have been applied in the task of anomaly detection [11]. Tandon et al. [12] presented a rule-based classification algorithm named “LERAD” which aimed to learn a series of comprehensible rules for detecting anomalies in area of intrusion detection. By employing bayesian networks, Das and Schneider conducted anomaly detection in the multi-class setting [13].

Unsupervised anomaly detection generally believes that normal data can be clustered into different categories or meet a certain distribution, while abnormal points do not belong to any categories or deviate from normal data distribution. In unsupervised learning, it distinguishes anomalies by learning the unified latent regularities and patterns of data points without considering their category labels. Due to it doesn’t require the ground truth of training samples, the unsupervised methods are widely applicable in anomaly detection. Common unsupervised learning methods include cluster analysis [14], statistical inference [5], density-based models [7] and so on. Otey et al. [15] employed a clustering method for detecting anomalies which was based on a novel distance measure for a mix of categorical and continuous attributes in datasets. Liu et al. [6] proposed “iForest” (Isolation Forest) algorithm, which was an ensemble-based rapid anomaly detection method with linear time complexity and high precision. It uses a very efficient strategy to find out which points are easily isolated by gradually splitting data space with a random hyperplane.

In the setting of semi-supervised anomaly detection, the training dataset often contains a small part of correct category labels, while the labels of the rest of dataset are unknown. Blanchard et al. [16] provided a semi-supervised novelty detection (SSND) algorithm to solve the problem of determining whether two random instances come from the same distribution.

B. AutoEncoder

Autoencoder [17], as a multi-layer feed forward neural network, is widely applied in computer vision [18], natural language processing [19], recommendation systems [20] and it has also achieved great success in the field of anomaly detection [21], [22]. The autoencoder model generally include an encoder and a decoder. The encoding stage is used to produce the high-order feature representation of the original input, the decoding stage is used to generate neurons with the same size of the original input. The higher-order feature representation is obtained by continuously reducing the reconstruction error of the original data and the decoded data, and this feature will be further utilized in various learning tasks. There are a lot of variants of autoencoder. For instance, by imposing some restrictions on the hidden layer, sparse autoencoder [23] can make it learn better characteristics of instances in a harsh environment, and can effectively reduce the dimensions of samples. The restriction can be a sparsity regularization of hidden layer. In order to improve the robustness of the algorithm, noise is added to the input data to form a denoising autoencoder [24]. The main idea of the denoising autoencoder is to realize the reconstruction of real data by training samples adding with noise. One way to add noise is to randomly select several entries and then set them to zero. Rifai et al. [25] proposed a contractive Auto-encoder by adding a penalty term to the cost function, and results showed that the penalty was helpful to obtain a representation that can better capture the local variation of the input data.

VI. Conclusion

In this paper, we propose a novel anomaly detection framework named AEKNN, which combines the training of autoencoder and a k-th nearest neighbor based outlier detection method. The loss function of our framework is restricted with the penalty item of deep data representation vector, which aim to learn a more sparse and robust encoding vector. Extensive experiments on the three datasets demonstrate the effectiveness of our method. Our future work includes developing other network structures to explore higher detection performance and solving the application problem for large scale real-world datasets.

VII. Acknowledgement

This work is partially supported by the grants from the National Natural Science Foundation of China (NSFC) (71701007, 71531001).

References

[1] S. Ramaswamy, R. Rastogi, and K. Shim, “Efficient algorithms for mining outliers from large data sets,” in *Acm Sigmod International Conference on Management of Data*, 2000, pp. 427–438.

[2] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *Computer Science*, 2014.

[3] (2018) Outlier detection datasets (odds). [Online]. Available: http://odds.cs.stonybrook.edu/

[4] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[5] P. Rousseeuw and KatrienVanDriessen, “A fast algorithm for the minimum covariance determinant estimator,” *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

[6] F. T. Liu, M. T. Kai, and Z. H. Zhou, “Isolation forest,” in *Eighth IEEE International Conference on Data Mining*, 2009, pp. 413–422.

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: Identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 2000, pp. 93–104.

[8] M. Goldstein and A. Dengel, “Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm,” in *In KI-2012: Poster and Demo Track*, 2012, pp. 59–63.

[9] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, 2009.

[10] Q. Song, W. Hu, and W. Xie, “Robust support vector machine with bullet hole image classification,” *IEEE Transactions on Systems Man & Cybernetics Part C*, vol. 32, no. 4, pp. 440–448, 2002.

[11] Y. H. Tsai, C. H. Ko, and K. C. Lin, “Using commonkads method to build prototype system in medical insurance fraud detection,” *Journal of Networks*, vol. 9, no. 7, 2014.

[12] G. Tandon and P. K. Chan, “Weighting versus pruning in rule validation for detecting network and host anomalies,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 697–706.

[13] K. Das and J. Schneider, “Detecting anomalous records in categorical datasets,” in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 220–229.

[14] C. Böhm, C. Faloutsos, and C. Plant, “Outlier-robust clustering using independent components,” in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD ’08, New York, NY, USA, 2008, pp. 185–198.

[15] M. Otey, S. Parthasarathy, A. Ghoting, G. Li, S. Narravula, and D. Panda, “Towards nic-based intrusion detection,” in *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2003, pp. 723–728.

[16] Blanchard, Gilles, Lee, Gyemin, Scott, and Clayton, “Semi-supervised novelty detection.” *Journal of Machine Learning Research*, vol. 11, no. 11, pp. 2973–3009, 2010.

[17] G. E. Hinton, S. Osindero, and Y. W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[18] J. Masci, U. Meier, D. Cirean, and J. Schmidhuber, “Stacked convolutional auto-encoders for hierarchical feature extraction,” in *International Conference on Artificial Neural Networks*, 2011, pp. 52–59.

[19] S. C. A. P, S. Lauly, H. Larochelle, M. M. Khapra, B. Ravindran, V. Raykar, and A. Saha, “An autoencoder approach to learning bilingual word representations,” *Advances in Neural Information Processing Systems*, vol. 3, pp. 1853–1861, 2014.

[20] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang, “A hybrid collaborative filtering model with deep structure for recommender systems,” in *AAAI*. AAAI Press, 2017, pp. 1309–1315.

[21] S. Hawkins, H. He, G. J. Williams, and R. A. Baxter, “Outlier detection using replicator neural networks,” in *International Conference on Data Warehousing and Knowledge Discovery*, 2002, pp. 170–180.

[22] G. Williams, R. Baxter, H. He, S. Hawkins, and L. Gu, “A comparative study of rnn for outlier detection in data mining,” in *IEEE International Conference on Data Mining, 2002. ICDM 2003. Proceedings*, 2002, pp. 709–712.

[23] L. Meng, S. Ding, and Y. Xue, “Research on denoising sparse autoencoder,” *International Journal of Machine Learning & Cybernetics*, vol. 8, no. 5, pp. 1719–1729, 2017.

[24] Y. Bengio, L. Yao, G. Alain, and P. Vincent, “Generalized denoising auto-encoders as generative models,” *Advances in Neural Information Processing Systems*, pp. 899–907, 2013.

[25] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, “Contractive auto-encoders: Explicit invariance during feature extraction,” in *In International Conference on Machine Learning*, 2011.