Unsupervised anomaly detection in network traffic using Deep Autoencoding Gaussian Mixture model

Leonid Safonov

***Abstract*—Unsupervised anomaly detection in high-dimensional data is an important subject of research in theoretical machine learning and applied areas. One of important applications is anomaly detection in network traffic data, which can be useful for preventing network security violations.**

Unsupervised anomaly detection is based on density estimation, which is problematic in high-dimensional data. To deal with the issue dimensionality, reduction is performed first, and then the density is estimated in a space of smaller dimension.

Recently deep learning methods have been widely used in high-dimensional anomaly detection. One of such methods is the Deep Autoencoding Gaussian Mixture Model (DAGMM).

DAGMM is a combination of a deep autoencoder, which performs dimensionality reduction and reconstruction error estimation, and a Gaussian mixture model, which predicts if a data sample is anomalous.

We apply DAGMM to unsupervised anomaly detection in network traffic data. Testing anomaly detection system on network data presents a problem of lack of a generally accepted benchmark dataset, which would be recent, contain different types of attacks and have labels. We chose to use the UNSW-NB15 dataset, which satisfies these requirements and has been suggested as an up-to-date benchmark.

A correction to the algorithm, which improves anomaly detection accuracy is proposed.

***Keywords*—network intrusion detection, anomaly detection, deep learning, unsupervised learning, autoencoder, Gaussian mixture.**

I. Introduction

Unsupervised anomaly detection is an important area in machine learning, which has many applications in different fields, including intrusion detection systems (IDS). A survey of machine learning methods in IDS can be found in [1].

The central element of anomaly detection is density estimation. The probability density of the input data is estimated, and the data points in low-probability areas can be designated as anomalous. In case of high-dimensional data, the density estimation in the original feature space is difficult because of the "curse of dimensionality," when distances between any two data points are little different from each other, and any data point's probability can be low [2].

Manuscript received May 25, 2021
L. Safonov is with Infotecs, Moscow, Russia
(email: leonid.safonov@infotecs.ru)

Network Intrusion Detection Systems (NIDS) monitor the traffic of the entire network by analyzing protocol activity. They usually belong to one of two classes: signature based and anomaly detection based systems. Signature based systems are suitable for detection of known anomalies and utilize supervised learning methods. The dataset used to train such systems must be labelled.

Anomaly detection based systems can theoretically be made suitable for detection of novel attacks, but are prone to false-positive results, because is not known whether the learning dataset is clean or contains signatures of attacks. Another potential problem with anomaly detection IDS is difficulty in feature selection in the traffic dataset, especially if the data is high-dimensional.

Deep learning based approaches are expected to overcome difficulties caused by necessity to learn features from high-dimensional data. Surveys of deep learning methods in intrusion detection can be found in [3]-[5].

To deal with the "curse of dimensionality" two-step approaches have been proposed, where, first, the dimensionality is reduced, and then, in the space of reduced dimension, the density is estimated [6]. However, dimensionality reduction and density estimation are performed independently of each other, which can result in the loss of performance. On the other hand, simultaneous execution of both components is difficult in realization.

Unsupervised anomaly detection methods can be classified into three major groups:

- reconstruction-based methods, e.g. principal component analysis,
- cluster analysis methods, in which initially the dimensionality is reduced, and then clusterization is performed,
- one-class classification methods.

Deep Autoencoding Gaussian Mixture Model (DAGMM) was proposed in [7]. It improves the existing deep learning models for unsupervised anomaly detection by addressing the following issues:

- it preserves key information from the input inferred by the dimensionality reduction, and

stores it together with the reconstruction error in a low-dimensional space,
- it adds an estimation sub-network which takes a low-dimensional output from the GMM and outputs mixture membership prediction,
- both networks are trained together end-to-end, without pre-training of either network.

II. DAGMM MODEL DESCRIPTION

The DAGMM neural network consists of two components – a compression network and an estimation network. The compression network is a deep autoencoder, which performs dimensionality reduction, produces low-dimensional representation of features, and sends it together with the reconstruction error to the input of the estimation network. The estimation network predicts if the data item belongs to one of the clusters, or is anomalous. The scheme of the network is shown in Fig. 1.

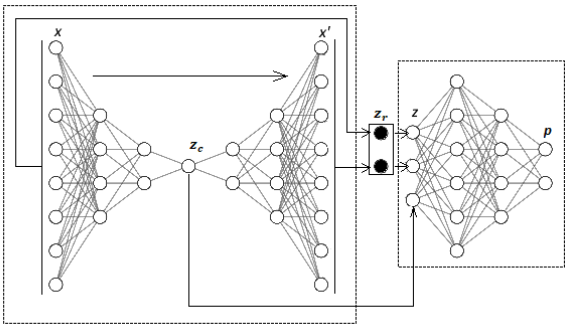

Figure 1. Scheme of DAGMM neural network.

The compression network receives the input data vector **x**, transformed to a standard form, and finds its low-dimensional representation

$$\mathbf{z}_c = h(\mathbf{x}, \theta_e),$$

where $h()$ is the encoding function, θ_e is a vector of parameters. From the low-dimensional representation the output vector

$$\mathbf{x}' = g(\mathbf{z}_c, \theta_d)$$

with the encoding function $g()$ and parameters θ_d is reconstructed. From the input and output vectors we find the vector of features calculated from the reconstruction error

$$\mathbf{z}_r = f(\mathbf{x}, \mathbf{x}'),$$

where $f()$ is a multi-dimensional function. In this study two features are used:
- Euclidean distance,
- cosine similarity.

The estimation network receives as input the reconstruction error features and the low-dimensional representation of the input value, and performs density estimation basing on the Gaussian mixture model.

The output of the estimation network is the vector **p**, whose dimension K is equal to the number of components of the Gaussian mixture. From this vector we have the prediction of membership as

$$\boldsymbol{\gamma} = \text{softmax}(\mathbf{p}),$$

where γ is a K-dimensional vector.

Based on this estimation, we have the parameters of the Gaussian mixture

$$\varphi_k = \frac{1}{N}\sum_{i=1}^{N}\gamma_{ik},$$

$$\mu_k = \frac{\sum_{i=1}^{N}\gamma_{ik}\mathbf{z}_i}{\sum_{i=1}^{N}\gamma_{ik}},$$

$$\widehat{\Sigma}_k = \frac{\sum_{i=1}^{N}\gamma_{ik}(\mathbf{z}_i-\mu_k)(\mathbf{z}_i-\mu_k)^T}{\sum_{i=1}^{N}\gamma_{ik}},$$

i.e. the mixture probability, mean and covariance for the k-th component in GMM, where $1 \le k \le K$, and N is the number of samples.

Using these parameters, we can find sample energy according to the Gaussian mixture model

$$E(\mathbf{z}) = -\log\left(\sum_{k=1}^{K}\varphi_k\frac{\exp\left(-\frac{1}{2}(\mathbf{z}-\mu_k)^T\widehat{\Sigma}_k^{-1}(\mathbf{z}-\mu_k)\right)}{\sqrt{(2\pi)^D\cdot\det(\widehat{\Sigma}_k)}}\right)$$

where $\boldsymbol{z} = [\mathbf{z}_c, \mathbf{z}_r]$ and $D = \dim(\mathbf{z})$. In our case D = 3: one-dimensional representation and two reconstruction error features (Euclidean distance and cosine similarity).

Using these energy values we can identify anomalous samples.

III. PROPOSED IMPROVEMENT

Samples with highest energy can be considered anomalies. However, it is not obvious, which measure can be used to objectively determine which samples are anomalous. It is suggested in [7], that a pre-chosen threshold is set to determine high-energy samples. This threshold is set either arbitrarily or using pre-knowledge of the ratio of anomalous data in the training data.

We suggest the following approach. Since anomalous samples are expected to have the highest energy values, in order to find them then we sort the energy values in decreasing order and find the largest interval between the neighboring values. If there are anomalous values, we can expect the maximal interval to be between them and normal samples (Fig. 2).

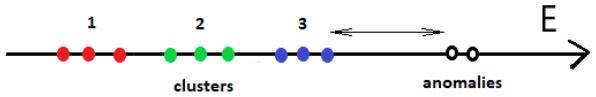

Figure 2. Sample energy distribution scheme showing distribution of samples between clusters of the Gaussian mixture. The anomalous samples are identified as outliers separated by the largest gap in energy values.

IV. OBJECTIVE FUNCTION

The objective function for the training of the model is

$$J(\theta_e, \theta_d, \theta_m) = \frac{1}{N}\sum_{i=1}^{N}L(\mathbf{x}_i, \mathbf{x}'_i) + \frac{\lambda_1}{N}\sum_{i=1}^{N}E(\mathbf{z}_i) + \lambda_2 P(\widehat{\Sigma}),$$

where

$$L(x_i, x'_i) = \|\mathbf{x}_i - \mathbf{x}'_i\|_2^2$$

is the L_2-norm,

$$P(\hat{\Sigma}) = \sum_{k=1}^{K} \sum_{j=1}^{d} \frac{1}{\hat{\Sigma}_{kjj}}$$

– is a term added to avoid singularity, when the diagonal elements of the covariance matrices are close to 0. λ_1 и λ_2 are free parameters.

V. Training dataset selection

When testing intrusion detection systems, it is necessary to compare results of different systems, which requires a benchmark dataset. An ideal dataset would

- be relatively recent,
- be labelled,
- contain pre-defined splits into training and testing datasets,
- be publicly available,
- contain real network traffic,
- include different types of attacks as well as normal traffic,
- cover a long time interval.

Such a dataset is impossible to create, and any available dataset would satisfy only some of these criteria. The main problem with creation of such a dataset is labelling, which for a large dataset would be very time-consuming.

For a long time the standard dataset for IDS system testing has been KDD CUP 99 [8]. However, this dataset has many flaws, the most serious of which is its age. Besides, it suffers from repeating patterns, excessive presence of some types of attacks in the testing subset, and non-stationarity, i.e. different ratios of some types of attacks in training and testing data [9]. In recent years several alternatives to KDD CUP 99 have been proposed, a detailed survey of which can be found in [10]. It is suggested there, that the most suitable alternatives are the CICIDS'17 [11], CIDDS-001 [12], UGR'16 [13] and UNSW-NB15 [14] datasets.

From these datasets we chose the UNSW-NB15 dataset as one of few, which have the following qualities:

- relative novelty,
- free accessibility,
- availability in the text form,
- presence of labels.

VI. UNSW-NB15 dataset

The UNSW-NB15 dataset was generated with the purpose of creating a benchmark dataset for NIDS testing. It has the following properties:

- Simulated using the IXIA Perfect Storm tool during a time interval of 31 hours. There are 45 unique IP addresses in 3 networks.
- Pcap-files were processed by Argus и Bro-IDS for feature extraction. In the final form the data is stored in CSV files and has 49 features (integer, float, timestamp, binary and nominal). The features fall into five groups: *Flow*, *Basic*, *Content*, *Time and Additionally Generated*.
- Incorporates 10 target classes – one normal and 9 anomalous: *Fuzzers*, *Analysis*, *Backdoors*, *DoS*, *Exploits*, *Generics*, *Reconnaissance*, *Shell Code* and *Worms*.
- Contains 175,341 data points and the test set 82,332 data points.
- The set is stationary, both train and test sets have the same distribution of normal and anomalous data.

VII. Implementation details

The data is preprocessed – numerical features are normalized to the interval [0,1], nominal features are transformed to binary vectors using one-hot encoding. After the preprocessing the data has dimension 128 with each feature being a number within the [0,1] interval.

The neural network is implemented with the following parameters. In the autoencoder the size of the input layer is equal to the number of features in the pre-processed input data. The size of each successive layer N_l equals $\lfloor N_{l-1}/2 \rfloor$, where N_{l-1} is the size of the previous layer, until $N_l = 1$. As the activation function tanh() is used.

In the estimation network the sizes of layers are 3, 20, 10, *K*.

VIII. Results

The data is divided into training and testing sets. The training set is created by random sampling of the selected portion of the total dataset. The remaining data is left for the testing set. In the first experiment only normal data is used for training, for further experiments a small percentage of attack data is added to the training set. This percentage is gradually increased from 1% to 20%. Initially the ratio between the train and test sets sizes is 3:1. Subsequently, the experiment with normal data is repeated for the ratio of 9:1.

Table 1 presents accuracy, precision, recall and F1-score of the DAGMM execution on the test dataset with different ratio of attack data added to the training data. The results indicate that the performance declines insignificantly when a small number of attacks is added to the training data.

Table 1. Results of testing depending on the ratio of attacks in the training data with the anomalous sample energy threshold determined by the largest gap in energy values (Fig. 2). The ratio between the training set and testing set lengths is 3:1.

Attacks ratio	Accuracy	Precision	Recall	F1
0%	0.8946	0.8950	0.9994	0.9443
1%	0.8917	0.8918	0.9998	0.9427
5%	0.8771	0.8790	0.9975	0.9345
10%	0.8616	0.8632	0.9977	0.9254
20%	0.8311	0.8319	0.9989	0.9077

Table 2 shows the results of experiments with the energy threshold value defined by an arbitrarily set

percentile cut-off. The training set contains only clean data. Several values of the threshold are tested, and all measures are lower, than those from the first experiment (row for 0% attack rate in Table 1). These results show that it is impossible to pre-select the threshold to outperform the algorithm with the sample energy threshold selected by the largest gap in the energy values.

Table 2. Results of testing with arbitrary energy threshold. The training set is clean from attacks, the ratio between the training set and testing set lengths is 3:1.

Threshold	Accuracy	Precision	Recall	F1
1%	0.8868	0.8961	0.9880	0.9398
5%	0.8611	0.9047	0.9443	0.9241
10%	0.8524	0.9097	0.9272	0.9184
20%	0.8378	0.9170	0.9003	0.9086

Table 3 presents the same results as in Table 1 with the ratio between the training set and test set lengths being 9:1.

Table 3. Results of testing depending on the ratio of attacks in the training data with the anomalous sample energy threshold determined by the largest gap in energy values (Fig. 1). The ratio between the training set and testing set lengths is 9:1.

Attacks ratio	Accuracy	Precision	Recall	F1
0%	0.9547	0.9551	0.9997	0.9768
1%	0.9487	0.9509	0.9976	0.9737
5%	0.9298	0.9347	0.9944	0.9636
10%	0.8988	0.9166	0.9783	0.9465
20%	0.8620	0.8726	0.9861	0.9259

From these tables we can conclude that setting the sample energy threshold using the largest gap between the neighboring points is more reliable than using a pre-selected percentile cut-off.

IX. Conclusion

We studied the application of the Deep Autoencoding Gaussian Mixture Model unsupervised learning algorithm to anomaly detection in the UNSW-NB15 dataset. It is found that the algorithm shows good results, which deteriorate insignificantly when the training data contain a small amount of data with attacks.

References

[1] S. Gulghane, V. Shingate, S. Bondgulwar, G. Awari, and P. Sagar, “A Survey on Intrusion Detection System Using Machine Learning Algorithms,” in: *Innovative Data Communication Technologies and Application. ICIDCA 2019*. Lecture Notes on Data Engineering and Communications Technologies, Springer, Cham, 2020, vol. 46, pp. 670-675.

[2] V. Chandola, A. Banerjee, and V. Kumar, Anomaly Detection: A Survey, *ACM Comput. Surv.* 41. 10.1145/1541880.1541882, 2009.

[3] Liu, H., Lang, B. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Appl. Sci.* 2019, vol. *9*, p. 4396.

[4] A. Aldweesh, A. Derhab, A. Z. Emam, “Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues,“ Knowledge-Based Systems, vol. 189, 105124, 2020.

[5] E. Hodo, X.J. Bellekens, A.W. Hamilton, C. Tachtatzis, and R.C. Atkinson, Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey. ArXiv, abs/1701.02145, 2017.

[6] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?,” *JACM*, vol. 58 (3), article 11, pp. 1 – 37.

[7] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, “Deep autoencoding Gaussian mixture model for unsupervised anomaly detection,” in *6th International Conference on Learning Representations*, 2018.

[8] “KDD Cup 1999 Data,” Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[9] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, “Benchmarking datasets for anomaly-based network intrusion detection: KDD CUP 99 alternatives.” *IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, 2018.

[10] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, “A survey of network-based intrusion detection data sets,” *ArXiv* abs/1903.02460, 2019.

[11] I. Sharasaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating s new intrusion detection dataset and intrusion traffic characterization”, *International Conference on Information Systems Security and Privacy (ICICSP)*, 2018, pp. 108-116.

[12] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, “Flow-based benchmark datasets for intrusion detection”, *International Conference on Cybernetic Intelligent Systems (CIS)*, ACPI, 2017, pp. 361-369.

[13] G. Marciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, UGR’16: A new dataset for the evaluation of cyclostantionarity-based network IDSs, *Computers & Security* 2018, vol. 73, pp. 411-424.

[14] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, 2015, pp. 1-6.