Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-020-00535-4

ORIGINAL RESEARCH

Enhancing anomaly detection through restricted Boltzmann machine features projection

Gustavo H. de Rosa[1] · Mateus Roder[1] · Daniel F. S. Santos[1] · Kelton A. P. Costa[1]

Received: 11 April 2020 / Accepted: 30 September 2020

Abstract Technology has been nurturing a wide range of applications in the past decades, assisting humans in automating some of their daily tasks. Nevertheless, more advanced technology systems also expose some potential flaws, which encourage malicious users to explore and break their security. Researchers attempted to overcome such problems by fostering intrusion detection systems, which are security layers that try to detect mischievous attempts. Apart from that, increasing demand for machine learning also enabled the possibility of combining such approaches in order to provide more robust detection systems. In this context, we introduce a novel approach to deal with anomaly detection, where instead of using the problem's raw features, we project them through a restricted Boltzmann machine. The intended approach was assessed under a well-known literature anomaly detection dataset and achieved suitable results, better than some state-of-the-art approaches.

Keywords Intrusion detection system · Anomaly detection · Machine learning · Restricted Boltzmann machine

✉ Gustavo H. de Rosa
gustavo.rosa@unesp.br

Mateus Roder
mateus.roder@unesp.br

Daniel F. S. Santos
danielfssantos1@gmail.com

Kelton A. P. Costa
kelton.costa@gmail.com

[1] Department of Computing, São Paulo State University, Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, Bauru, SP 17033-360, Brazil

1 Introduction

The development of technology fostered the capacity of autonomous systems solving particular tasks. Nowadays, it is common to observe digital agents being applied everywhere, ranging from daily bank tasks to world wide web browsers. However, increasing connectivity among users also fosters mischievous intents, i.e., users who attempt to break digital security rules and gain access to unauthorized devices or confidential information [5].

An attempt to overcome such a problem is to employ digital barriers, commonly known as anomaly-based intrusion detection systems (IDS) [1]. Essentially, they are security layers that enable any application to detect a malicious attempt and revoke their access instantly [4]. Moreover, they can analyze previous data and decide whether an incoming attack is possible or not [19], helping systems to protect themselves against security attacks.

Even though IDSs seems to be the ideal tool to cope with such attempts, some flaws need to be addressed [22]. Every day new attacks and security breaches are being explored, inflicting the necessity of updating the IDS knowledge. Furthermore, depending on the type of application the IDS is protecting, there is no possibility of raising false positives or false negatives. In the past decade, researchers are combining concepts of machine learning techniques and IDS in an attempt to produce more intelligent and self-adaptable anomaly detection systems [12]. The overall idea is to use past data, i.e., previous learning, in order to update the IDS knowledge and prepare it for future attacks.

Nevertheless, machine learning is not capable of addressing such problems without raising new problems. One can perceive that every machine learning algorithm is susceptible to the data that it is feed with, being only

Published online: 13 October 2020

reasonable when the dataset is accurately modeled and represent the real possibilities of the problem [13]. An additional problem regards when the dataset is not balanced, providing more samples that pertain to one class than another. Such a problem may vitiate the learning process and prompt the technique to better learn the most common class instead of adequately learning the whole dataset.

A recent technique that has attracted considerable spotlight is the restricted Boltzmann machine [7], mainly due to its simple architecture and vast learning capacity. The restricted Boltzmann machine is a stochastic network that deals with probabilities and physical concepts, such as entropy and energy. Moreover, it is capable of learning real data distributions and reconstructing the original data in different latent spaces [15], i.e., acting as an auto encoder–decoder and extracting new features from the data distribution.

In this work, we propose to address such a problem by enhancing the raw features classification through a feature extraction approach. In other words, we will use the restricted Boltzmann machine as an auto encoder–decoder to project the raw features into new features spaces, and further classify them using state-of-the-art classifiers. In order to validate such a proposal, we will validate it under a well-known literature anomaly detection dataset, known as NSL-KDD. Additionally, we will provide several setups, i.e., the number of projected features, to assess whether our approach has been able to model the real data distribution and sample new features correctly. Therefore, the main contributions of this paper are twofold: (1) to introduce restricted Boltzmann machines to the context of features projection in anomaly detection systems, and (2) to fill the lack of research regarding Restricted Boltzmann Machines as auto encoder–decoder frameworks.

The remainder of this paper is organized as follows. Sections 2 and 3 present the theoretical background related to Intrusion Detection Systems and Restricted Boltzmann Machines, respectively. Section 4 discusses the methodology adopted in this work, while Sect. 5 presents the experimental results. Finally, Sect. 6 state conclusions and future works.

2 Intrusion detection system

When dealing with the network's security, it is common to find users that attempt to gain unauthorized access to resources, commonly known as intruders, and classified into two types: external and internal. External intrusions are unauthorized users that attempt to enter a particular machine, while internal intruders are users who can access a particular machine without root or administrator privileges [23]. Although intruders are only classified in a binary way, their possible attacks are more distinct, such as viruses, worms, trojans, denial of services (DoS), users to root (U2R), remote to local (R2L) and probing [3, 14].

Therefore, an attempt to overcome such issues has arisen in the form of intrusion detection systems, which are security layers useful to analyze users, systems, and the network's activity. They are also capable of reporting possible vulnerabilities, evaluating the system's file integrity, recognizing a pattern in attacks, and analyzing unusual activities, i.e., intrusions [2]. Additionally, they are divided into two types: host-based IDS and network-based IDS, and based on the specific strategies, such as misuse [9] and anomaly detection [3]. As a misuse detection strategy is based on pre-known attack signatures, it is not useful when dealing with new attacks. On the other hand, the anomaly detection strategy is fitted to identify whether the network presents malicious or common behaviors.

2.1 Anomaly-based IDS

It is vital to highlight that in the past decades, anomalies have been a prominent research field, mainly due to the increasing amount of unknown attacks [6, 18]. One can perceive that anomaly-based IDSs are capable of distinguishing whether the network's traffic is different from its typical pattern by using machine learning classification techniques, being extremely effective when identifying new attacks that have not been discovered yet [17]. Nevertheless, such an approach can still be manipulated while training the classification algorithms. Essentially, malicious users send anomalous packages in order to difficult the model's learning and occasionally misclassify likely attacks.

3 Restricted Boltzmann machines

Restricted Boltzmann machines are stochastic neural networks based on energy principles guided by physical laws and characterized by energy, entropy, and temperature factors. Most of the time, these networks learn in an unsupervised fashion and are applicable to a wide variety of problems, that range from image reconstruction, collaborative filtering, and feature extraction to pre-training deeper networks.

In terms of architecture, RBM contain a visible layer v with m units and a hidden layer h with n units. Additionally, a real-valued matrix $W_{m\times n}$ models the weights between the visible and hidden neurons, where w_{ij} represents the connection between the visible unit v_i and the

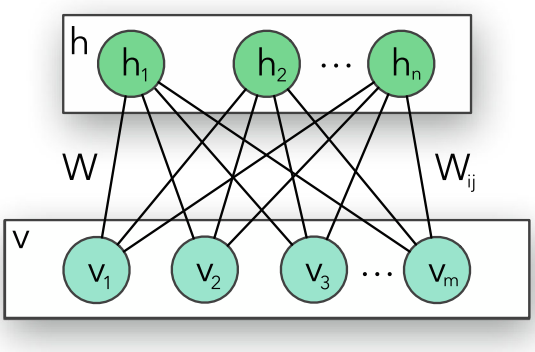

Fig. 1 The vanilla RBM architecture

hidden unit h_j. Figure 1 describes the well-know vanilla RBM architecture.

Mainly, the visible layer receives the data for processing, while the hidden layer learns its pattern and probabilistic distribution. Additionally, assume all units from layers v and h are binary and derived from a Bernoulli distribution [8], i.e., $v \in \{0,1\}^m$ and $h \in \{0,1\}^n$. Equation 1 models the energy function of an RBM:

$$E(v,h) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i=1}^{m}\sum_{j=1}^{n} v_i h_j w_{ij}, \tag{1}$$

where a and b represent the biases of visible and hidden units, respectively.

Furthermore, Eq. 2 models the joint probability of a given configuration (v, h):

$$P(v,h) = \frac{e^{-E(v,h)}}{Z}, \tag{2}$$

where Z is the partition function, which normalizes the probability over all possible configurations, considering visible and hidden units. Moreover, Eq. 3 represents the marginal probability of an input (visible units) vector:

$$P(v) = \frac{\sum_h e^{-E(v,h)}}{Z}. \tag{3}$$

Note that the RBM is a bipartite graph, allowing the information to flow in a non-directional manner, from visible to hidden neurons and vice versa. With this property, it is possible to formulate mutually independent activations for both units. Equations 4 and 5 describe their conditional probabilities:

$$P(v|h) = \prod_{i=1}^{m} P(v_i|h), \tag{4}$$

and

$$P(h|v) = \prod_{j=1}^{n} P(h_j|v), \tag{5}$$

where $P(v|h)$ and $P(h|v)$ stand for the probability of the visible layer given the hidden states and probability of the hidden layer given the visible states, respectively.

From Eqs. 4 and 5, it is possible to obtain the probability of activating a single visible neuron i given the hidden states, and the probability of activating a single hidden neuron j given the visible states. Equations 6 and 7 describe these activations:

$$P(v_i = 1|h) = \sigma\left(\sum_{j=1}^{n} w_{ij} h_j + a_i\right), \tag{6}$$

and

$$P(h_j = 1|v) = \sigma\left(\sum_{i=1}^{m} w_{ij} v_i + b_j\right), \tag{7}$$

where $\sigma(\cdot)$ stands for the logistic-sigmoid function.

Fundamentally, an RBM needs to learn a set of parameters $\theta = (W, a, b)$ through a training algorithm. One can observe this approach as an optimization problem, which aims to maximize the product of data probabilities for all the training set $\mathscr{V}$, as stated below:

$$\arg\max_{\Theta} \prod_{v \in \mathscr{V}} P(v). \tag{8}$$

An interesting approach to model this problem is applying the negative of the logarithm function represented by the negative log-likelihood (NLL), which represents the distribution approximation of the reconstructed data over the original data. Therefore, one can employ the partial derivatives of W, a and b at iteration t to cope with this problem. Equations 9–11 describe the parameters update rules:

$$W^{t+1} = W^t + \eta(vP(h|v) - \tilde{v}P(\tilde{h}|\tilde{v})), \tag{9}$$

$$a^{t+1} = a^t + (v - \tilde{v}), \tag{10}$$

and

$$b^{t+1} = b^t + (P(h|v) - P(\tilde{h}|\tilde{v})), \tag{11}$$

where η stands for the learning rate, $\tilde{v}$ stands for the reconstruction of the visible layer given h, and $\tilde{h}$ denotes an estimation of the hidden vector h given $\tilde{v}$. Such optimization technique is also known as contrastive divergence (CD) [7].

One interesting proposition by Tieleman [21] to improve the optimization results of the CD method, also known as the persistent contrastive divergence (PCD), is to not reset the Markov chain between parameter updates. Such a train strategy proven to be more efficient in

approximating the RBM reconstructed data distribution to the real data underlying distribution.

4 Methodology

In this section, we present the proposed approach to perform the projection of the raw features in a new latent space concerning restricted Boltzmann machines, as well as describe the employed dataset and the experimental setup.

4.1 Features projection

An interesting way to enhance the problem of classifying features is to apply a transformation and load the features up in a new latent space, commonly known as projection. In this work, we propose to use restricted Boltzmann machines as the projection algorithm, i.e., each RBM is responsible for learning its corresponding class data distribution and further reconstructing the samples from that particular class in a new latent space, with more or fewer features than the original ones.

4.2 Dataset

The KDD'99 Cup dataset[1] is amongst one of the most researched datasets in the context of anomaly detection. Stolfo et al. [16] have developed it based on the data collected by DARPA's 98 IDS evaluation program [11]. Furthermore, the KDD'99 training dataset has roughly 4,900,000 records, holding 41 features and labeled as either normal or attack. Additionally, some versions describe the type of attacks.

Nevertheless, the KDD'99 dataset provides a vast number of redundant records, with a rough estimation of 78% and 75% of the records in training and testing sets [20], respectively. This massive amount of redundant records can bias the classifiers or harm the evaluation results. Some researches attempt to model this drawback using statistical analysis to show that it affects the intrusion detection systems [10, 20].

In order to solve these issues, a refined dataset, the NSL-KDD,[2] was proposed by Tavallaee et al. [20], focusing on overcoming the previously mentioned problems by manually selecting records from the complete KDD'99 Cup dataset. One can observe that there is no redundancy in the NSL-KDD dataset, where its attack classes are divided into four categories:

- Denial of service attack (DoS): It is an attack category in which the victim's resources are attacked, making the resource too busy and unable to handle legitimate requests or even denying legitimate users access to a machine;
- Probing attack: Is comparable to an inspection in which the objective is to gather information about a network of computers to bypass its security controls;
- User to root attack (U2R): Is a type of exploit in which the attacker starts with access to an average user account on the system and attempts to gain root access to the system by exploiting some vulnerability;
- Remote to local attack (R2L): Occurs when an attacker sends packets to a machine over a network without authorized access, exploiting vulnerabilities to gain local access as a user of that machine.

Table 1 RBMs hyperparameters configuration

Acronym	Class	Hyperparameters
RBM_n	Normal	$e = 300, b = 100, k = 1, \alpha = 0.01$
RBM_d	DoS	$e = 300, b = 100, k = 1, \alpha = 0.001$
RBM_r	R2L	$e = 300, b = 100, k = 1, \alpha = 0.005$
RBM_u	U2R	$e = 300, b = 100, k = 1, \alpha = 0.005$
RBM_p	Probe	$e = 300, b = 100, k = 1, \alpha = 0.005$

4.3 Modeling RBM-based features projection

The NSL-KDD dataset is composed of five classes, such as normal, dos, r2l, u2r, and probe. Therefore, for each class, we employ a particular RBM in order to learn that class data distribution. Afterward, all RBMs weight matrices are concatenated in a unique matrix in order to perform the projection. For example, if we are working with 100 hidden units RBMs and a 39 features dataset, our projection matrix will have size equal to 38×500. Additionally, Table 1 exhibits the hyperparameters employed for each RBM, where e stands for epochs, b for batch size, k for contrastive divergence steps, and α for learning rate. Finally, each RBM had experiments with distinct number of hidden neurons, such as 10, 20, 38, 80 and 100.

After the feature projection step, we use state-of-the-art classifiers in order to validate the proposed approach, such as support vector machine (SVM) with radial basis function (RBF) kernels, decision trees (DT), and random forests (RF). Note that, before training the SVMs, we used a fourfold cross-validation in order to find the most appropriate cost and standard deviation hyperparameters. Also, before training the DTs and RFs, we used a random search algorithm in order to find their best hyperparameters. Finally, in order to assess such methodology, we use the

[1] https://www.kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[2] https://www.unb.ca/cic/datasets/nsl.html.

following classification metrics: accuracy, precision, recall, f1-score, as well as the ROC curve.

5 Experimental results

This section aims at presenting the experimental results concerning RBM's feature projection.[3]

5.1 Support vector machine with radial basis function

Tables 2 and 3 describe the experimental results over KDDTest+ concerning SVM-RBF trained with KDDTrain+ 20% and KDDTrain+, respectively. One can perceive that the 38-f experiment, i.e., 38 projected features, obtained the best metrics considering all possibilities of projected features, followed by the 100-f and 20-f approaches.

Such results are corroborated in Figs. 2 and 3, which illustrates the ROC curves over KDDTrain+ 20% and KDDTrain+ datasets, respectively. In these figures, it is possible to observe that the 38-f approach obtained of the best curves, complementing the results described by the already mentioned tables.

5.2 Decision tree

Regarding the decision tree classifier, Tables 4 and 5 describe the experimental results over KDDTest+ trained with KDDTrain+ 20% and KDDTrain+, respectively. Regarding the former dataset, it is important to highlight that the 20-f approach obtained the best metrics, followed by the 38-f experiment. Additionally, one can observe that in the latter dataset that the 20-f experiment obtained the best precision while the 38-f achieved the best remaining metrics.

Figures 4 and 5 portray the ROC curves over KDDTrain+ 20% and KDDTrain+ datasets using the DT classifier, respectively. As one can observe, the ROC curves were worst than the ones achieved by the SVM-RBF classifier. Nevertheless, it is crucial to highlight that it is possible to corroborate the results obtained by Tables 4 and 5, where the 20-f and 38-f experiments achieved the best metrics.

5.3 Random forest

Finally, Tables 6 and 7 describe the experimental results over KDDTest+ concerning RF trained with KDDTrain+ 20% and KDDTrain+, respectively. Even though in the former dataset, the 38-f achieved the best metrics, one can perceive that in the latter dataset, the 10-f was the one to achieve the best metrics.

Furthermore, Figs. 6 and 7 illustrate the ROC curves over KDDTrain+ 20% and KDDTrain+ datasets using the DT classifier, respectively. In both figures, it is possible to observe that there was not much distinction between all the employed projections. However, analyzing the ROC curve,

[3] The experiment's source code is available at https://github.com/danielfssantos/anomaly_detection.

Table 2 Experimental results over KDDTest+ concerning SVM-RBF trained with KDDTrain+ 20%

	Standard	10-f	20-f	38-f	80-f	100-f
Accuracy	0.7792	0.7794	0.8086	**0.8305**	0.7999	0.8259
Precision	0.8155	0.8035	0.8340	**0.8482**	0.8222	0.8459
Recall	0.8021	0.7982	0.8281	**0.8473**	0.8182	0.8436
F1-score	0.7786	0.7792	0.8084	**0.8305**	0.7998	0.8258

Table 3 Experimental results over KDDTest+ concerning SVM-RBF trained with KDDTrain+

	Standard	10-f	20-f	38-f	80-f	100-f
Accuracy	0.7681	0.8018	0.7931	**0.8078**	0.7900	0.8067
Precision	0.8103	0.8303	0.8244	**0.8340**	0.8228	0.8339
Recall	0.7927	0.8223	0.8145	**0.8276**	0.8119	0.8268
F1-score	0.7670	0.8016	0.7927	**0.8076**	0.7896	0.8065

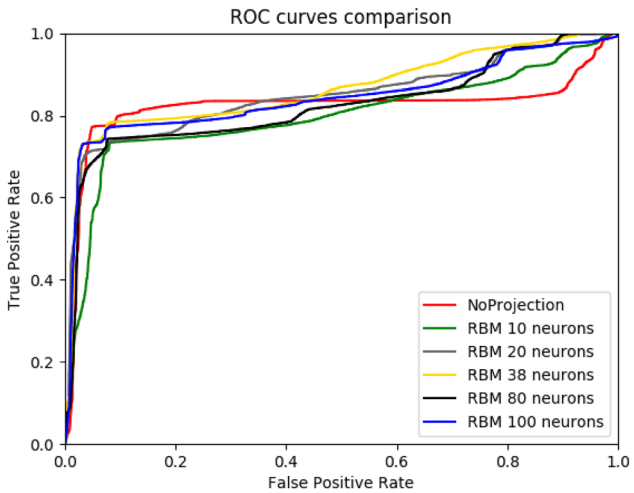

Fig. 2 ROC curves comparison of KDDTest+ trained with KDDTrain+ 20% using SVM-RBF classifier

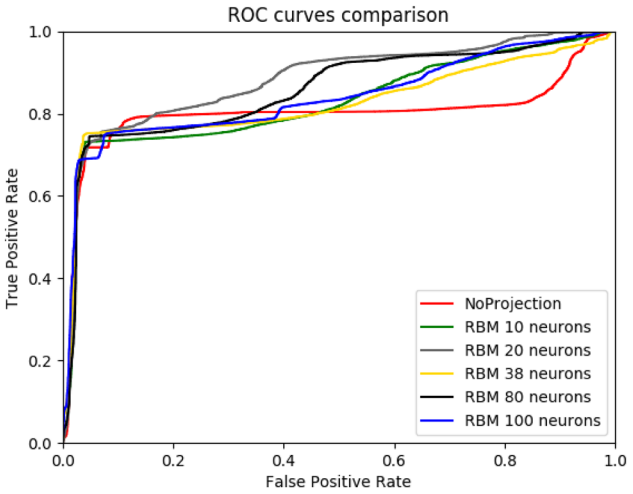

Fig. 3 ROC curves comparison of KDDTest+ trained with KDDTrain+ using SVM-RBF classifier

Table 4 Experimental results over KDDTest+ concerning DT trained with KDDTrain+ 20%

	Standard	10-f	20-f	38-f	80-f	100-f
Accuracy	0.7809	0.8120	**0.8189**	0.8112	0.7812	0.8009
Precision	0.8190	0.8351	**0.8399**	0.8350	0.8070	0.8288
Recall	0.8043	0.8307	**0.8369**	0.8302	0.8007	0.8212
F1-score	0.7801	0.8119	**0.8189**	0.8111	0.7810	0.8007

it is possible to conclude that the RF classifier obtained the worst results considering all classifiers.

5.4 Overall discussion

Figure 8 portrays a principal component analysis (PCA) using the first two components over the KDDTrain+ 20% dataset without any RBM's feature projection. It is possible to observe that the space is tangled between all classes, being unfeasible to depict individual classes clusters.

Furthermore, when employing the RBM's feature projection, as the ones depicted in Figs. 9, 10, 11, 12 and 13, it is possible to observe that an increasing number of projections causes a better separability between the classes, enabling the identification of particular clusters. Moreover, one can also perceive that using an extensive amount of features, as the one depicted by Fig. 13, it is not ideal, mainly due to the attempt of overindulging the space's separability. Therefore, considering the figures mentioned above, we can corroborate the results obtained in the previous subsections, attesting that almost all best results were encountered within 20–80 projected features.

Table 5 Experimental results over KDDTest+ concerning DT trained with KDDTrain+

	Standard	10-f	20-f	38-f	80-f	100-f
Accuracy	0.7686	0.7754	0.7942	**0.7997**	0.7742	0.7808
Precision	0.8122	0.8006	**0.8258**	0.8172	0.7989	0.8175
Recall	0.7935	0.7947	0.8157	**0.8162**	0.7933	0.8038
F1-score	0.7674	0.7752	0.7939	**0.7997**	0.7740	0.7802

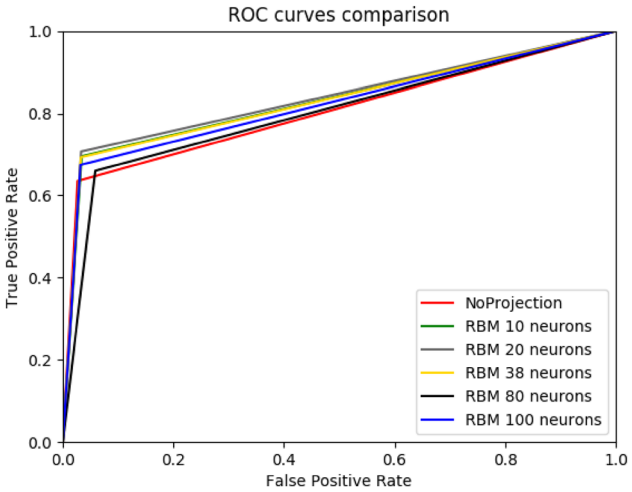

Fig. 4 ROC curves comparison of KDDTest+ trained with KDDTrain+ 20% using DT classifier

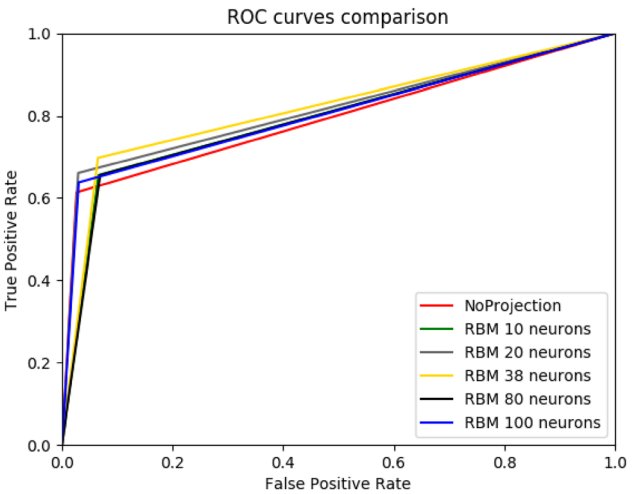

Fig. 5 ROC curves comparison of KDDTest+ trained with KDDTrain+ using DT classifier

Table 6 Experimental results over KDDTest+ concerning RF trained with KDDTrain+ 20%

	Standard	10-f	20-f	38-f	80-f	100-f
Accuracy	0.7757	0.7979	0.7982	**0.7987**	0.7966	0.7958
Precision	0.8157	0.8278	0.8279	**0.8285**	0.8271	0.8263
Recall	0.7996	0.8189	0.8191	**0.8196**	0.8178	0.8169
F1-score	0.7748	0.7976	0.7979	**0.7984**	0.7963	0.7955

Table 7 Experimental results over KDDTest+ concerning RF trained with KDDTrain+

	Standard	10-f	20-f	38-f	80-f	100-f
Accuracy	0.7708	**0.7905**	0.7711	0.7765	0.7725	0.7765
Precision	0.8133	**0.8246**	0.8136	0.8165	0.8142	0.8162
Recall	0.7954	**0.8127**	0.7957	0.8004	0.7969	0.8004
F1-score	0.7697	**0.7900**	0.7700	0.7756	0.7715	0.7756

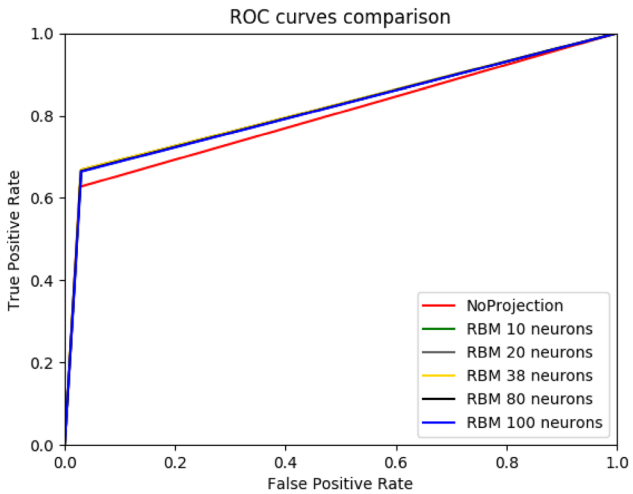

Fig. 6 ROC curves comparison of KDDTest+ trained with KDDTrain+ 20% using RF classifier

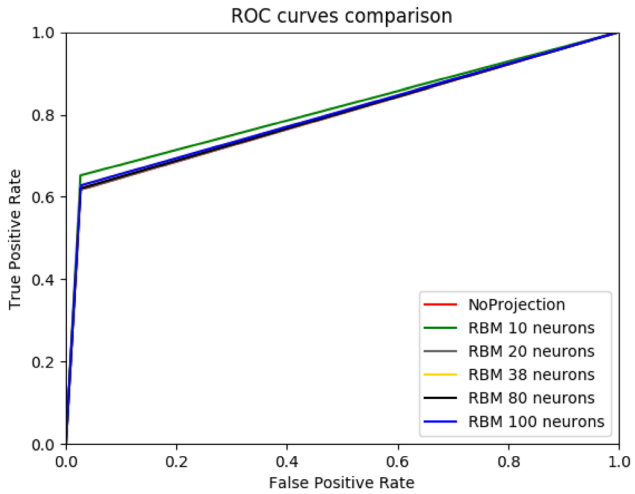

Fig. 7 ROC curves comparison of KDDTest+ trained with KDDTrain+ using RF classifier

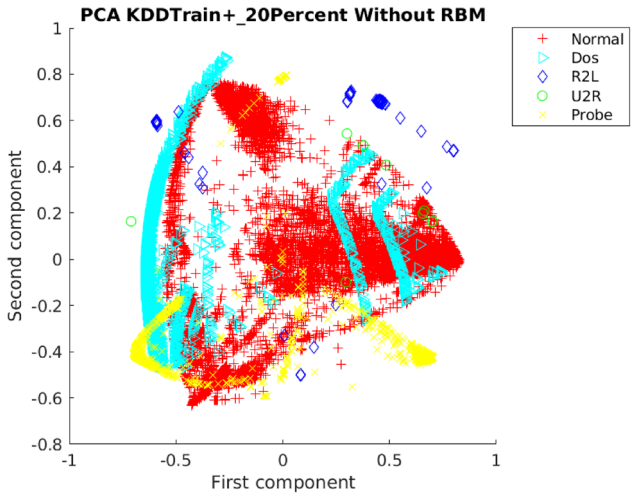

Fig. 8 Projection of KDDTrain+ 20% dataset (standard) using PCA

6 Conclusion

This paper addressed the problem of enhancing raw anomaly detection features through a Restricted Boltzmann Machine feature extraction task. A well-known literature dataset, NSL-KDD, was employed in order to validate such an approach. Additionally, we provide a comparison amongst some of the state-of-the-art classifiers and a distinct number of projected features in order to provide a more robust experimental setup.

For every classifier, the proposed approach was able to outperform the baseline approach (raw features classification). Moreover, it is possible to highlight that for each classifier a different number of projected features achieved the best results. Furthermore, it is crucial to observe that the RBMs were capable of learning the real data distribution, enabling them to sample and project the data into the new space accurately.

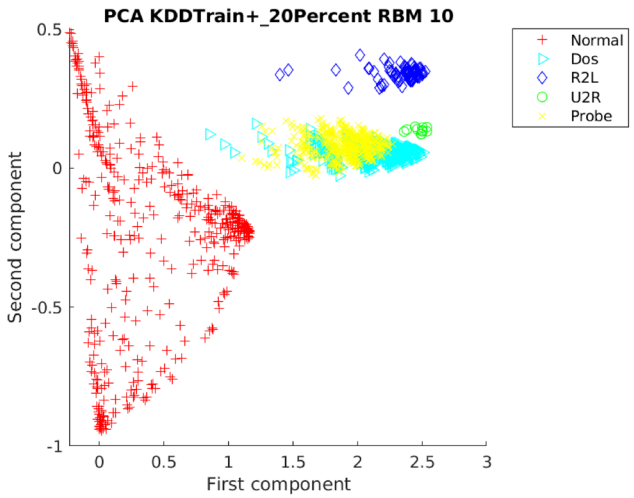

Fig. 9 Projection of KDDTrain+ 20% dataset (10 extracted features) using PCA

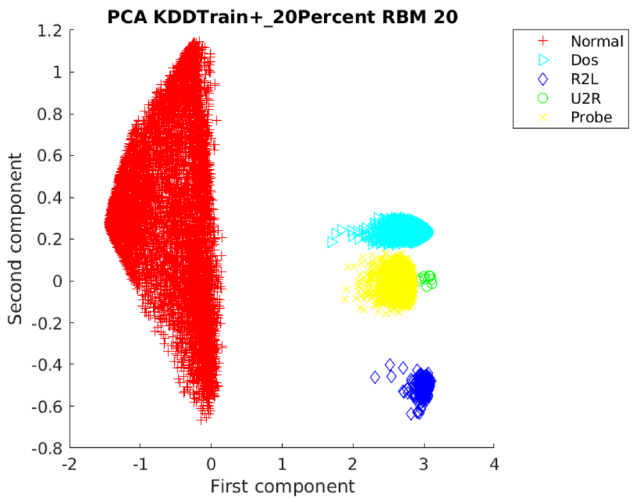

Fig. 10 Projection of KDDTrain+ 20% dataset (20 extracted features) using PCA

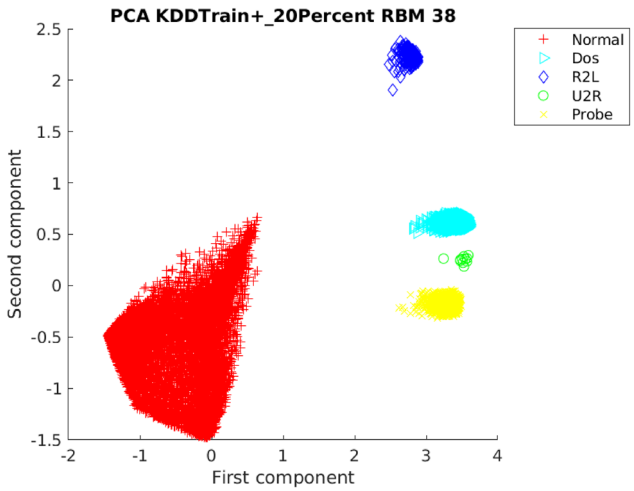

Fig. 11 Projection of KDDTrain+ 20% dataset (38 extracted features) using PCA

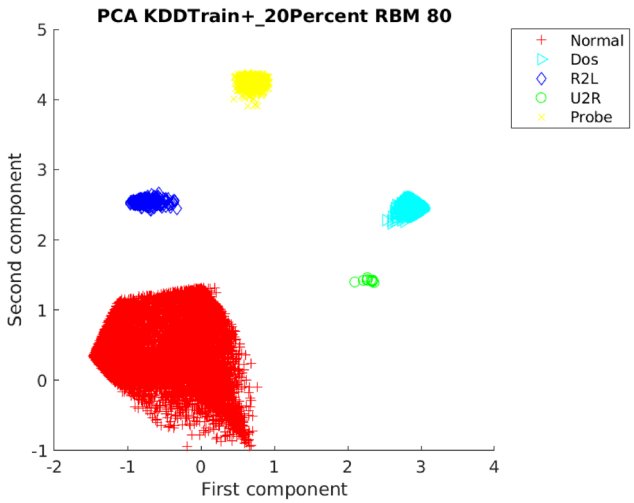

Fig. 12 Projection of KDDTrain+ 20% dataset (80 extracted features) using PCA

Regarding future works, we aim to explore within more depth the possibility of employing Restricted Boltzmann Machines to generate artificial data. It is common to witness that several literature datasets lack anomaly data when dealing with real-life network security attacks, allowing further research to be conducted in an attempt to fill these blanks.

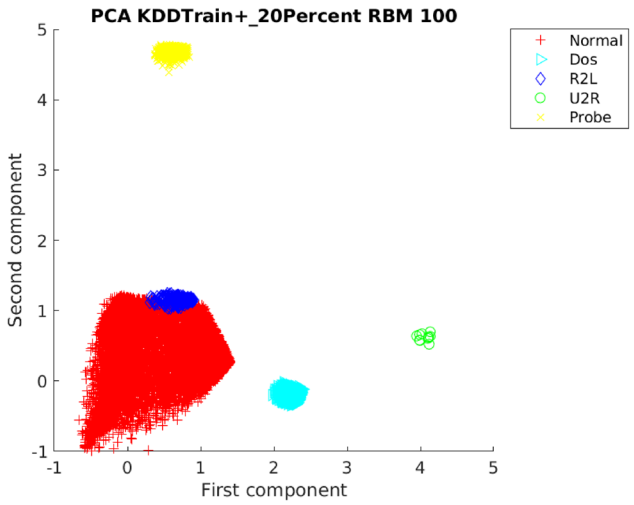

Fig. 13 Projection of KDDTrain+ 20% dataset (100 extracted features) using PCA

Acknowledgements The authors appreciate São Paulo Research Foundation (FAPESP) Grants #2019/02205-5 and #2019/07825-1, as well as Petrobras Grant #2017/00285-6.

References

1. Aldweesh A, Derhab A, Emam AZ (2020) Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowl Based Syst 189:105124
2. Bhuyan M, Bhattacharyya D, Kalita J (2014) Network anomaly detection: methods, systems and tools. IEEE Commun Surv Tutor 16(1):303–336
3. Bijone M (2016) A survey on secure network: intrusion detection and prevention approaches. Am J Inf Syst 4(3):69–88
4. Chalapathy R, Chawla, S (2019) Deep learning for anomaly detection: a survey. arXiv preprint arXiv:1901.03407
5. Cisco (2013) The 2018 Cisco annual security report. Cisco Systems
6. Gan XS, Duanmu JS, Wang JF, Cong W (2013) Anomaly intrusion detection based on PLS feature extraction and core vector machine. Knowl Based Syst 40:1–6
7. Hinton G (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14(8):1771–1800
8. Hinton G (2012) A practical guide to training restricted Boltzmann machines. Neural networks: tricks of the trade. Lecture notes in computer science, vol 7700. Springer, Berlin, pp 599–619
9. Hodo, E, Bellekens, X, Hamilton, A, Tachtatzis, CAtkinson, R (2017) Shallow and deep networks intrusion detection system: a taxonomy and survey. arXiv preprint arXiv:1701.02145
10. Kaushik SS, Deshmukh P (2011) Detection of attacks in an intrusion detection system. Int J Comput Sci Inf Technol (IJCSIT) 2(3):982–986
11. Lippmann R, Fried D, Graf I, Haines J, Kendall K, McClung D, Weber D, Webster S, Wyschogrod D, Cunningham R, Zissman M (2000) Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation. Proceedings DARPA Information Survivability Conference and Exposition, vol 2. Hilton Head, pp 12–26. https://doi.org/10.1109/DISCEX.2000.821506
12. Moustafa N, Hu J, Slay J (2019) A holistic review of network anomaly detection systems: a comprehensive survey. J Netw Comput Appl 128:33–55
13. Schmidt L, Santurkar S, Tsipras D, Talwar K, Madry A (2018) Adversarially robust generalization requires more data. In: Advances in neural information processing systems, pp 5014–5026
14. Shahbaz MB, Wang X, Behnad A, Samarabandu J (2016) On efficiency enhancement of the correlation-based feature selection for intrusion detection systems. In: 2016 IEEE 7th annual information technology, electronics and mobile communication conference (IEMCON). IEEE, pp 1–7
15. Srivastava N, Salakhutdinov, RR (2016) Multimodal learning with deep Boltzmann machines. In: Advances in neural information processing systems, pp 2222–2230 (2016)
16. Stolfo S, Fan W, Lee W, Prodromidis A, Chan P (2000) Cost-based modeling for fraud and intrusion detection: results from the jam project, vol 2, pp 130–144. https://doi.org/10.1109/DISCEX.2000.821515
17. Tama B, Comuzzi M, Rhee K (2019) Tse-ids: a two-stage classifier ensemble for intelligent anomaly-based intrusion detection system. IEEE Access 7:94497–94507
18. Tama B, Rhee K (2016) Performance analysis of multiple classifier system in dos attack detection. Revised Selected Papers of the 16th International Workshop on Information Security Applications, vol 9503. Springer, Berlin, pp 339–347. https://dl.acm.org/doi/10.1007/978-3-319-31875-2_28
19. Tama BA, Patil AS, Rhee K (2017) An improved model of anomaly detection using two-level classifier ensemble. In: 12th Asia joint conference on information security (AsiaJCIS), Seoul, pp 1–4. https://doi.org/10.1109/AsiaJCIS.2017.9
20. Tavallaee M, Bagheri E, Lu W, Ghorbani A (2009) A detailed analysis of the KDD cup 99 data set. In: Proceedings of the second IEEE international conference on computational intelligence for security and defense applications, CISDA'09. IEEE Press, Piscataway, NJ, USA, pp 53–58
21. Tieleman T (2008) Training restricted Boltzmann machines using approximations to the likelihood gradient. In: Proceedings of the 25th international conference on Machine learning. ACM, pp 1064–1071
22. van Oorschot PC (2020) Intrusion detection and network-based attacks. In: Computer security and the internet. Springer, pp 309–338
23. Wang J, Shan Z, Gupta M, Rao HR (2019) A longitudinal study of unauthorized access attempts on information systems: the role of opportunity contexts. MIS Q 43(2):601–622