DRL for Intrusion Detection using VAE Reduced State Space

Hanish Prashant Dhanwalkar
Mechanical Engineering,
IIT Bombay
Mumbai, India
210100060@iitb.ac.in /
hanishdhanwalkar.iitb@gmail.com

***Abstract*—The application of new techniques to increase the performance of intrusion detection systems is crucial in modern data networks with a growing threat of cyber-attacks. These attacks impose a greater risk on network services that are increasingly important from a social end economical point of view. In this work we present a novel application of several deep reinforcement learning (DRL) algorithms to intrusion detection using a labeled dataset. We present how to perform supervised learning based on a DRL framework. Also, Intrusion detection frameworks are based on a supervised learning paradigm that uses a training dataset composed of network features and associated intrusion labels. In this work, we integrate this paradigm with a reinforcement learning algorithm that is normally based on interaction with a live environment (not a pre-recorded dataset). To perform the integration, the live environment is replaced by a simulated one. The principle of this approach is to provide the simulated environment with an intelligent behavior by, first, generating new samples by randomly extracting them from the training dataset, generating rewards that depend on the goodness of the classifier's predictions, and, second, by further adjusting this initial behavior with an adversarial objective in which the environment will actively try to increase the difficulty of the prediction made by the classifier. In this way, the simulated environment acts as a second agent in an adversarial configuration against the original agent (the classifier). We prove that this architecture increases the final performance of the classifier. This work presents the first application of adversarial reinforcement learning for intrusion detection, and provides a novel technique that incorporates the environment's behavior into the learning process of a modified reinforcement learning algorithm.**

***Index Terms*—Intrusion detection, Reinforcement learning, Adversarial learning, VAE, auto-encoders**

I. Introduction

Given the significance of contemporary security threats targeting sophisticated and high-demand networks, coupled with the economic importance of services operating on these networks and the escalating demands placed on data networks by these services, it becomes crucial to depend on automated systems proficient in promptly and reliably detecting intrusions. An example of such a system is an Intrusion Detection System (IDS), with its ultimate objective being the swift and precise analysis of network traffic to anticipate potential threats.

The Intrusion Detection System (IDS) is recognized as a key application in the realm of machine learning research for emerging data networks. IDS encounters significant challenges in prediction, given the necessity to manage extensive, noisy, and imbalanced datasets. Additionally, the features derived from network traffic are intricate, often accompanied by a noisy attribution of labels to their respective ground-truth states. This labeling complexity arises from the challenge of accurately determining the true value of the intrusion state, leading to the manual association between features and intrusion labels.

For this work, we have applied anomaly-based supervised machine learning (ML) models to two well-known, intrusion detection dataset: the NSL-KDD dataset (Tavallaee et al., 2009).

A Reinforcement Learning (RL) algorithm can provide very interesting properties for intrusion detection: (1) It is a very general framework with a flexible reward function, that does not require to be differentiable. (2) Once the training is done, the resulting policy function is usually a simple and fast neural network. (3) Thanks to the use of simple reward functions, the algorithm is suitable for online training, which allows a rapid response to changes in network conditions.

Considering the above points, we propose a new model that integrates a simulated environment, which provides samples of network traffic and rewards with an agent, which implements the classifier, and which is trying to predict the correct intrusion label based on the network samples given by the environment. The rewards generated by the environment will be positive/negative depending on the correct/incorrect prediction of the agent. The algorithm is trained with the objective of maximizing the total sum of rewards.

As a summary, the motivations/contributions of this work are:
– To propose a new classifier model for intrusion detection in networking.
– To present a classifier which is fast, flexible and with excellent performance metrics for prediction.
– To integrate the RL framework with a supervised classification problem.
– To explore the important field of research of multi-agent and adversarial RL, and its application to intrusion detection.
– To apply an adversarial RL to address the training bias associated with an unbalanced dataset.

II. Related works

There are many works in the literature presenting results for intrusion detection using different datasets.

Machine learning and intrusion detection: Intrusion detection can be addressed as anomaly detection [14] or as a classification problem, depending on the available data. In this analysis we focus on the classification perspective. Many works apply ML for intrusion detection with the NSL-KDD dataset: they report an accuracy of 79.9% for test data for the 5-labels prediction scenario, applying an MLP with three layers. Authors in the report, an F1 of 98% for the 5-labels scenario using AdaBoost with Naive Bayes as weak learner and a previous feature selection.

Reinforcement learning and intrusion detection: The work in Adversarial environment reinforcement learning algorithm for intrusion detection by Guillermo Caminero, Manuel Lopez-Martin, Belen Carro. The paper studys about application of various RL algorithms on IDS. The present work is very similar to the paper using Q-learning algorithm on latent space reduced using VAE (Variational auto-encoders).
The work in Towards Traffic Anomaly Detection via Reinforcement Learning and Data Flow by A. Servin presents an anomaly detector based on reinforcement learning with a simulated network environment, where anomalies are injected in a controlled manner and the reward is based on the correct detection of the anomalies. The experiment presents similarities with the present work: although the environment is physically simulated, the reward function is manually controlled and is not generated by the environment itself, and the real-time generation of sequences of actions, states and rewards could be assimilated to those registered in a dataset.

Variational Auto - Encoders: Numerous prior works have illuminated the efficacy of Variational Autoencoders (VAEs) in the realm of dimensional reduction across diverse domains. In the field of computer vision, Kingma and Welling (2013) introduced VAEs as a powerful generative model, demonstrating their ability to learn probabilistic encodings of images. Subsequent research by Rezende et al. (2014) extended VAEs by introducing the concept of normalizing flows, further improving their generative capabilities.

III. Work description

In this Section, (1) we describe the datasets chosen to compare the detection capacity of the different models, and (2) provide a comprehensive review of our proposed model (VAE-RL model).

The datasets employed are described in Section III.A. The proposed model is presented in detail in Section III.B.

A. Selected datasets

NSL-KDD dataset: The NSL-KDD dataset is a classic well-known IDS dataset. NSL-KDD dataset is an evolution of the KDD-99, solving the redundant records problems of the original dataset. The NSL-KDD dataset has 125973 training samples and 22544 test samples, containing 41 features: 38 continuous and 3 categorical (discrete valued). These features have been additionally transformed: scaling all continuous features to the range [0–1] and one-hot encoding all categorical features. This provides a final dataset with 122 features: 38 continuous and 84 with binary values (0, 1) associated to the three one-hot encoded categorical features. This is a very unbalanced dataset with a frequency of 43.1% and 1.7% for the most and least frequent labels.

Each training sample has a label output from 23 possible labels (normal plus 22 labels associated to different types of anomaly). The test data has the same number of features (41) and output labels from 38 possible values. That implies that the test data has anomalies not presented at training time. The 23 training and 38 testing labels have 21 labels in common; 2 labels only appear in training and 17 labels are unique to the test dataset. Up to 16,6% of the samples in the test dataset correspond to labels unique to the test dataset, and which were not present at training time. The existence of new labels at testing introduces an additional challenge to the learning methods.

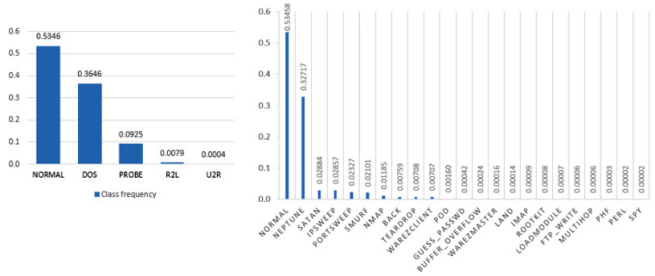

Fig. 1. Frequency distribution of intrusion classes (left) and types of attacks (right) for the NSL-KDD dataset.

B. Model description

- We propose a novel VAE-RL model that implementing classifier based on Q learning algorithm . In a RL framework, an environment informs an agent of the state of the environment and the agent responds with an action within the environment. This action eventually produces a change in the state of the environment. The environment responds to the action of the agent with a reward that is associated with how good/bad the action is in terms of some final goal. Considering this framework of an agent that interacts with a live environment, our proposed model provides the following modifications: – We provide a simulated environment whose states correspond to random samples taken from a dataset of labeled attacks (network intrusions).
  – The agent implements a classifier which tries to predict the intrusion label from the states provided by the simulated environment.

– The simulated environment produces rewards according to the correct/incorrect predictions of the agent.

The RL model:

The Q-learning algorithm is based on finding the best Q-function for the agent (the classifier in our case). A Q-function estimates a value for each state-action pair, this value corresponds with the sum of the rewards for the given state considering that we take a certain action and then move forward with the current policy. The Q-function can be calculated iteratively by:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma \max_A Q(S_{t+1}, A_{t+1}) - Q(S_t . A_t) \right]$$

where S_t is the current state, A_t is the current action, A_{t+1} is the next action, R_{t+1} is the next reward value, α is the learning rate and γ is the discount factor which in this case is set close to zero because the states are not correlated with each other (they are obtained by sampling the dataset, not in a sequential order), and therefore there is no need for the algorithm to remember previous states.

(Values of 1.0 and 0.001 for and have been used for α and γ, respectively.)

Both agents, the environment agent and the classifier (main) agent, are trained in parallel (both using DQN). The action (output) of the classifier agent will be the intrusion type prediction for a sample of the dataset (input), and the action of the environment agent will be the class of attack that will be used to generate new samples in the training process. To approximate the Q-function, a fully connected neural network (NN) is used, following. The input to this NN is the current state, which corresponds to the features extracted from the labeled dataset. The output of the NN represents the Q-function for the set of available actions.

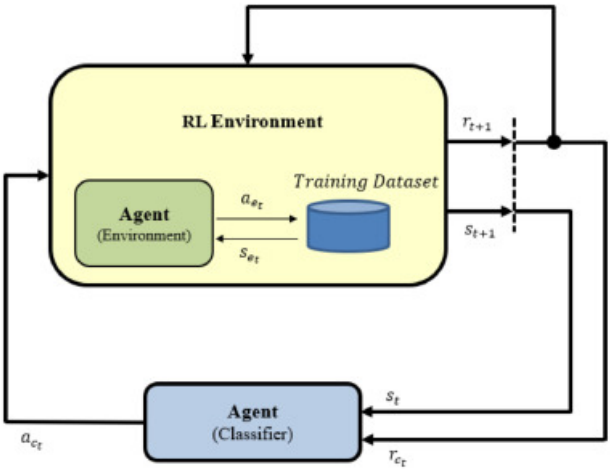

Fig: Reinforcement learning interaction between the agent and its intelligent environment.

The VAE model:

Variational Autoencoders (VAEs) have emerged as one of the most popular approaches to unsupervised learning of complicated distributions. VAEs are appealing because they are built on top of standard function approximators (neural networks), and can be trained with stochastic gradient descent. The generative model's (here, VAE's) job is to somehow capture the dependencies between features of the dataset and reduce dimensionality for ease of learning on these features afterwards down the system.

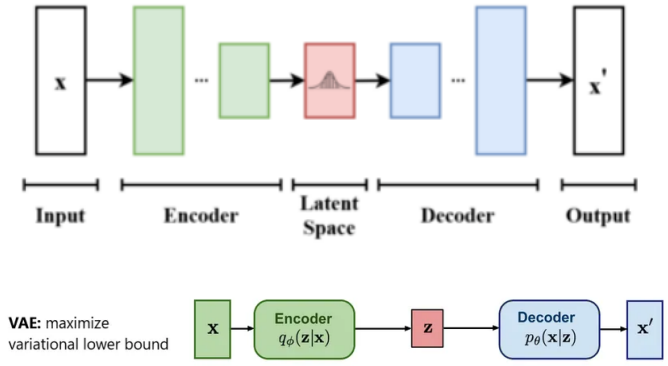

Fig. 2. model learns the parameters in order to model the distribution of the data points.

With this framework, the new model generates a more balanced data sampling by producing samples in which the classifier fails more frequently. This may be due to the low frequency of occurrence in the training set or to the difficulty of prediction for some states. All positive rewards for the main agent will be negative for the environment. In this way, the environment learns which are the classes in which the main agent fails most frequently and with a certain probability increases the frequency of these samples.

VAE model is used in dimensionality reduction which addresses the curse of dimensionality, enhancing model generalization by reducing sparsity in high-dimensional data. Computational efficiency is significantly improved, making models more scalable and reducing training and evaluation times.

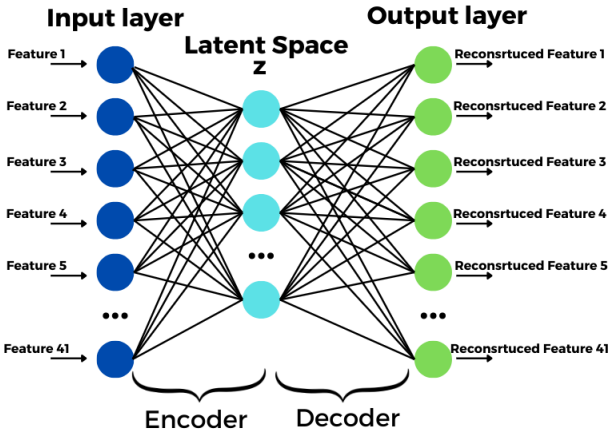

Fig. 3. VAE model, 41 input features, learns distribution among them to predict latent space of dimension of 8, Calculates Reconstruction loss from decoder.

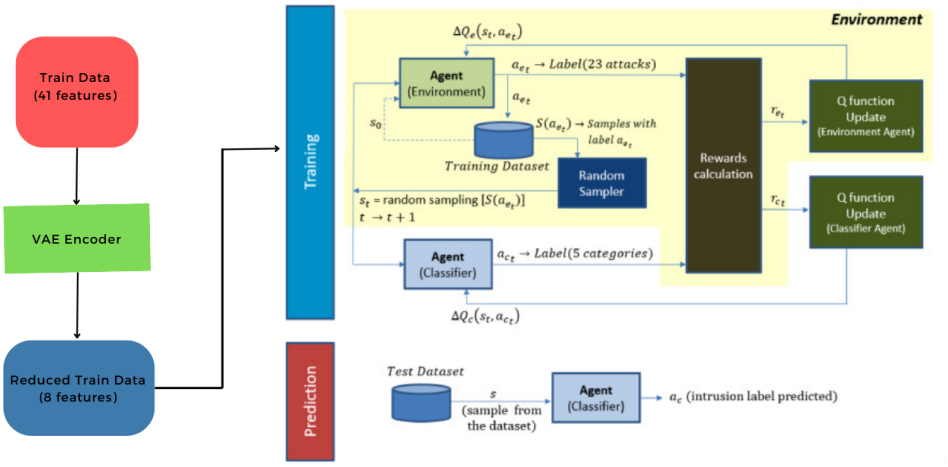

Fig. 4. Overall model, VAE reducing the dimension by learning the distribution in teh dataset. The learnt latent space used by RL algo to predict class of the IDS system

VAE model description:
Completely implemented in python using tensorflow anf keras.

1) Encoder Neural Network:
   - Input Layer: The input layer of the encoder receives features (41 features in the current work).
   - Hidden Layers: 22 hidden fully connected dense neural networks.
   - Latent Space: The final layer of the encoder is Sampling layer which produces the latent representation, for particular case of VAE, it samples mean and log variance of the distribution. Here a numpy array of length 8. This layer returns $z_m + \exp 0.5 * \log \sigma * \epsilon$, z being input to the encoder NN.
2) Decoder Neural Network:
   - Latent Space Input: The latent representation obtained from the encoder serves as the input to the decoder.
   - Hidden Layers: Like the encoder, the decoder comprising of 2 fully connected dense hidden layers that aim to reconstruct the original input from the latent representation.
   - Output Layer: The output layer of the decoder generates the reconstructed data, attempting to closely match the input data from the latent space.
3) Losses:
   - Kullback–Leibler divergence (KL loss):

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right),$$

   - Reconstruction loss:

$$= x \log p_{out} + (1 - x) \log(1 - p_{out}),$$

IV. Conclusion

IDS is a critical service for modern data networks. Network security attacks are complex, cannot be easily assigned to network patterns and are constantly changing. The special nature of network intrusion detection makes it necessary to investigate new models that can address some of the difficulties imposed by IDS on a machine learning algorithm, such as: noisy, unbalanced and complex datasets under changing conditions. This work tries to provide a new alternative, in the form of a model that brings the RL framework to the IDS problematic. The RL algorithms have been particularly successful in other areas (e.g. robotic, finance, videogames, business operations...) and we are able to prove that they may also be available for IDS.
The proposed new model (AE-RL) integrates the reinforcement and supervised frameworks, providing a simulated environment that follows the guidelines of an RL environment. The resulting environment is able to: 1) interact with a dataset of pre-recorded samples formed by network features and associated intrusion labels, and 2) select samples with an optimized policy to achieve the best possible classification results. The specific learning mechanism provided to the environment is based on an adversarial strategy.

References

Adversarial environment reinforcement learning algorithm for intrusion detection [1].
Network anomaly detection: methods, systems and tools [2].
Towards Traffic Anomaly Detection via Reinforcement Learning and Data Flow [3].

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [**?**].

References

[1] Guillermo Caminero, Manuel Lopez-Martin, Belen Carro,"Adversarial environment reinforcement learning algorithm for intrusion detection ".
[2] M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita, "Network anomaly detection: methods, systems and tools"
[3] Arturo Servin, "Towards Traffic Anomaly Detection via Reinforcement Learning and Data Flow"