Net-GAN: Recurrent Generative Adversarial Networks for Network Anomaly Detection in Multivariate Time-Series

Gastón García González†, Pedro Casas†, Alicia Fernández*, Gabriel Gomez*
*Facultad de Ingeniería, Universidad de la República, Uruguay – †AIT Austrian Institute of Technology, Austria

Abstract—We introduce Net-GAN, a novel approach to network anomaly detection in time-series, using recurrent neural networks (RNNs) and generative adversarial learning (GAN). Different from the state of the art, which traditionally focuses on univariate measurements, Net-GAN detects anomalies in multivariate time-series, exploiting temporal dependencies through RNNs. Net-GAN discovers the underlying distribution of the baseline, offering a powerful approach to detect anomalies in complex, difficult to model network monitoring data.

I. NET-GAN: ARCHITECTURE AND APPROACH

Network monitoring data generally consists of hundreds or thousands of counters periodically collected in the form of time-series, resulting in a complex-to-analyze multivariate time-series process (MTS). In particular, detecting anomalies in such multivariate, temporal data is challenging. Without loss of generality, we refer to the MTS as a set of \( n \), non-iid time series sampled at the same rate, referred to as \( x_t = \{x_t(1), x_t(2), \ldots, x_t(n)\} \in \mathbb{R}^n \). Current approaches to anomaly detection tackle this challenge by either focusing on univariate time-series analysis – running an independent detector for each time-series \( x_t(i) \), or by considering multi-dimensional input data \( x \in \mathbb{R}^n \) at each time \( t \), neglecting the temporal aspects of the MTS. To improve the state of affairs we propose Net-GAN, a novel unsupervised approach to anomaly detection in MTS data, based on Recurrent Neural Networks (RNNs), trained through a Generative Adversarial Networks framework (GAN) [1]. To capture the temporal correlations characterizing an MTS, we adapt the original model proposed in [1], replacing the multilayer perceptrons by recursive, LSTM networks for both generator and discriminator models. The input data is therefore sequences of multi-dimensional measurements, of length \( T: \{x_{t-T}, ..., x_t\} \). Net-GAN is inspired by previous work on GANs for time-series synthesizing and anomaly detection [2]–[4].

Fig. 1 depicts the Net-GAN architecture and both the training and anomaly detection procedures. In the training phase (left), the generator \( G \) draws synthetic sample sequences \( G(z) \) from Gaussian noise – the latent space \( Z \), with the objective of deceiving the discriminator \( D \), which in turn learns to determine whether training samples are real or derived from the generative distribution. The classification result proposed by \( D \) is additionally fed back to \( G \), serving as a reinforcement loop to guide the generation process. As both \( G \) and \( D \) compete to achieve their adversarial tasks, synthetic samples become more and more “realistic”, and the discriminator becomes robust to noise, improving the detection of non-conforming (i.e., out of the baseline) samples. In the application phase (right), the trained discriminator \( D^* \) acts naturally as an anomaly detector, detecting deviations from the baseline, through a discrimination loss function. The trained generator \( G^* \) is also used to improve detection performance, serving as baseline generation; by doing an inverse search in the latent space, we find the sample \( z \in Z \) which generates the closest sample to the tested one, producing a residual loss. Finally, both the discrimination and the residual loss functions are combined into an anomaly score, which is compared to a calibrated threshold to take the final decision.

II. NET-GAN: PRELIMINARY EVALUATION RESULTS

Fig. 2 reports the performance achieved by Net-GAN in the detection of synthetic attacks to industrial control systems, using a labeled, publicly available dataset of MTS sensor network measurements [4] – which includes 51 monitored time-series. For simplicity, in this example we only use \( D^* \) as detector. While results are preliminary, Net-GAN detects 56% of the attacks with a FPR below 1%. Being purely data-driven, we are currently working with better and bigger network measurement datasets to improve Net-GAN.

REFERENCES