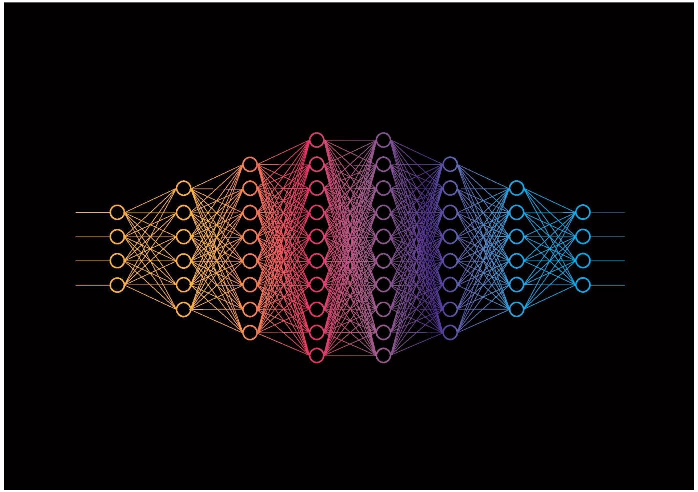
A Performance-Oriented Comparison of Neural Network Approaches for Anomaly-based Intrusion Detection

Presentation by Jesse Ables

Paper by Stefano Iannucci, Jesse Ables, William Anderson, Bhuvanesh Abburi, Valeria Cardellini, and Ioana Banicescu

Motivation



Anomaly Detection Techniques

- A. Aldweesh, et al. categorizes ANN approaches into: generative, discriminative, and hybrid
- Generative Approaches:
  - Autoencoders (AE)
  - Recurrent Neural Networks (RNN)
- Hybrid Approach:
  - Generative Adversarial Network (GAN)

[1,2]

Anomaly Detection Techniques

- https://paperswithcode.com
- Works published in or after 2018
- Results from highly ranked conferences or journals were considered
- Algorithms Selected:
    - REPresentations for a random nEarest Neighbor (REPEN) and DevNet for AE
    - OmniAnomaly for RNN
    - Multi-Objective Generative Adversarial Active Learning (MO-GAAL) for GAN

Anomaly Detection Techniques

- REPEN
  - Incorporates outlier detection into training process
  - Requires few labeled samples to improve accuracy
- DevNet
  - Utilizes a small number (~30) labeled anomalies to enforce "statistically significant deviations" with a prior and a neural deviation learner
  - Output scores are "highly interpretable" as they are directly applicable to z-score testing
  - Based on statistical pre-processing and an ANN.
- OmniAnomaly
  - Created to handle multivariate time series and the temporal dependence between data instances
  - Based on a combination of Variational Autoencoder and RNN
- MO-GAAL
  - Re-define the concept of anomaly with respect to the density of the sample space
  - Avoids computationally expensive calculation by generating synthetic data
  - GAN is used to generate outliers that occur near real data
  - Multiple generators are used to avoid mode collapse

[3,4,5,6]

Datasets

- NSL-KDD
  - Created in 1999
  - 150K Samples
  - Contamination rate of 46.5%
- CIC-IDS-2017
  - Created in 2017
  - 2.8M Samples
  - Contamination rate of 19.7%

Experiment Design

- 12 hour time limit
- 20%, 40%, 60%, 80%, and 100% subsample Datasets
  - Samples are randomly extracted from the full dataset
  - OmniAnomaly uses non-randomized extractions
- 4-fold cross validation is used
  - Except for OmniAnomaly where Scikit-Learn's TimeSeriesSplit function is used
- Effectiveness and Performance metrics were recorded on a 'per epoch' basis

Experiment Design

- Preprocessing for general datasets
  - One-hot encoding

Preprocessing for each algorithm was done based on their respective authors' choices

- Repen
  - Min-Max scaling
- DevNet
  - No additional preprocessing was done
- OmniAnomaly
  - Min-Max scaling
- Mo-GaaL
  - Flipped labels to match expected input

Experiment Design

Metrics Recorded:

- Max F1
- ROC AUC
- PR AUC
- CPU usage
- Virt, Res, Shr memory usage

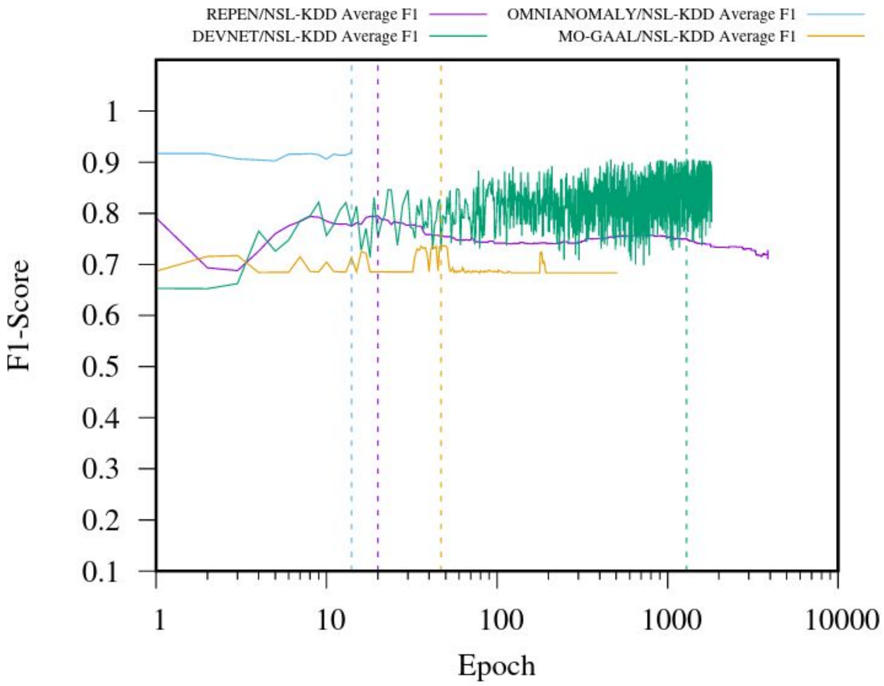


Fig. 1. Learning Curves vs Number of Epochs with NSL-KDD

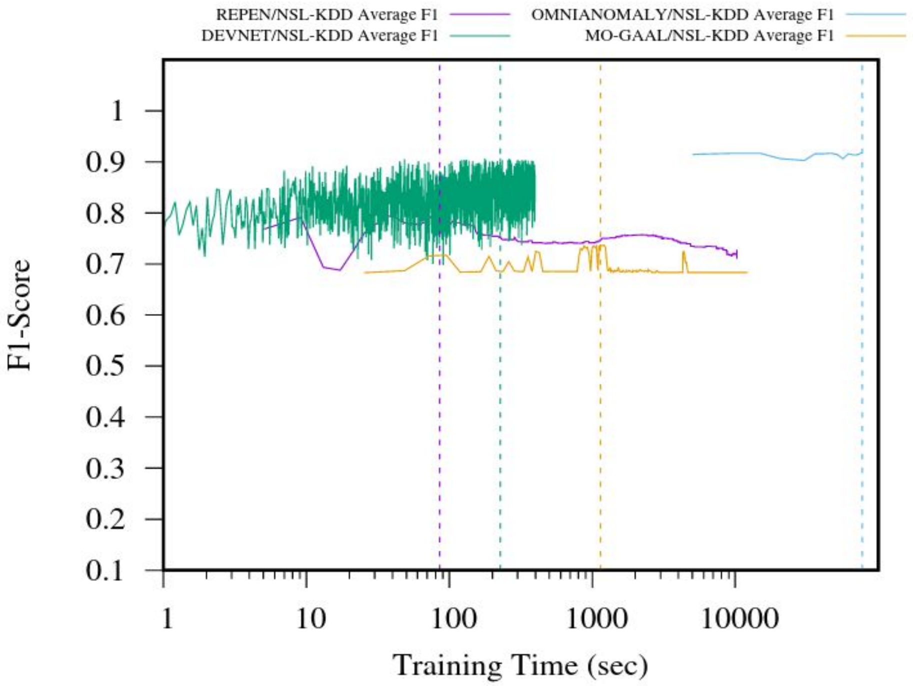


Fig. 2. Learning Curves vs vs Training Time with NSL-KDD

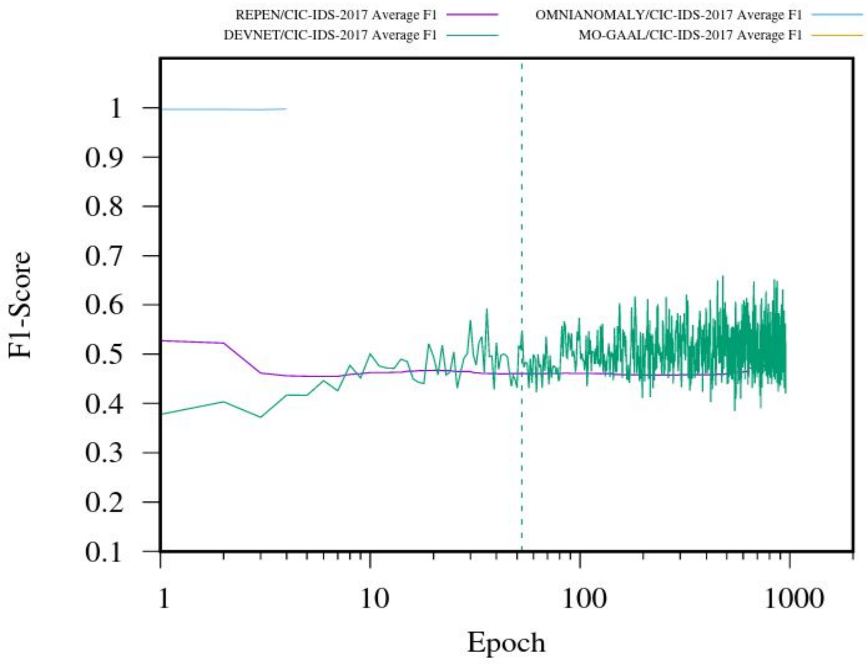


Fig. 3. Learning Curves vs Number of Epochs with CIC-IDS-2017

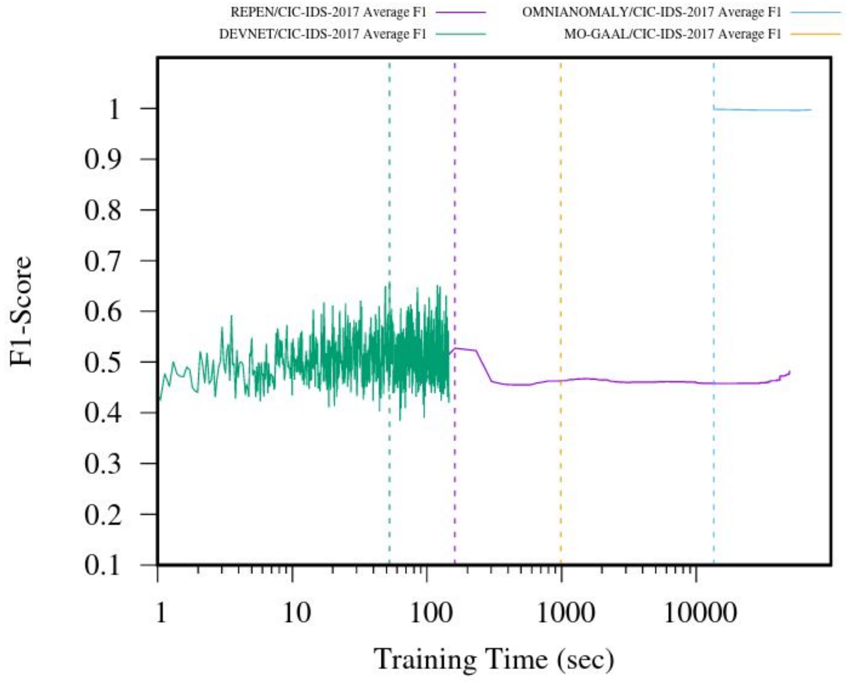


Fig. 4. Learning Curves vs Training Time with CIC-IDS-2017

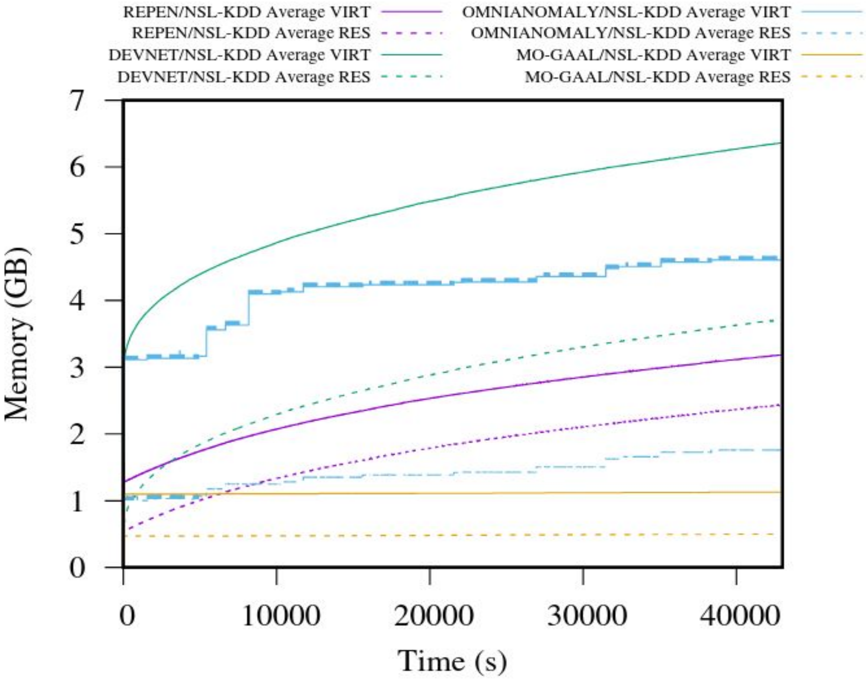


Fig. 5. Memory Usage Over Time with NSL-KDD

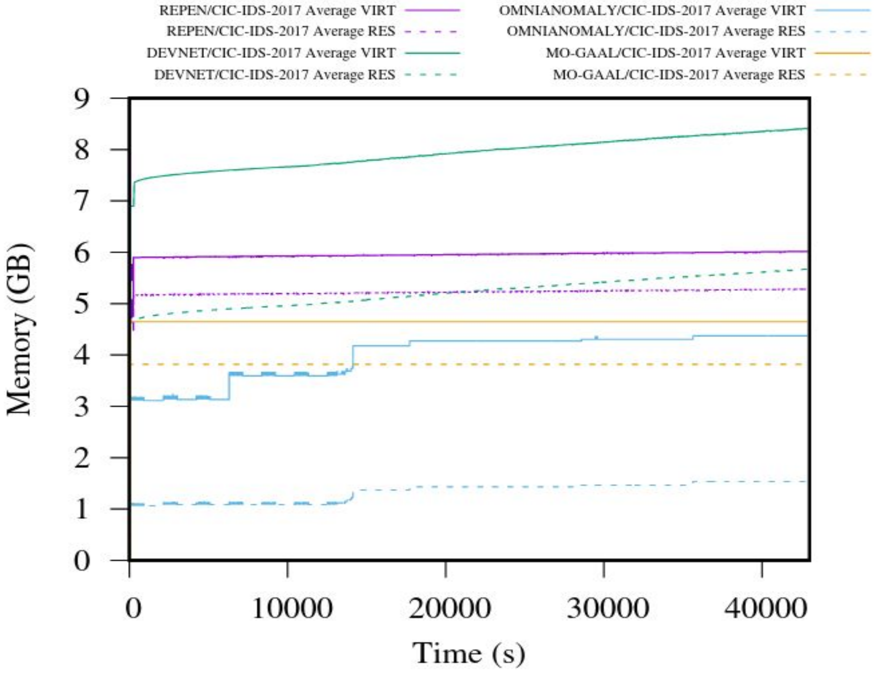


Fig. 6. Memory Usage Over Time Time with CIC-IDS-2017

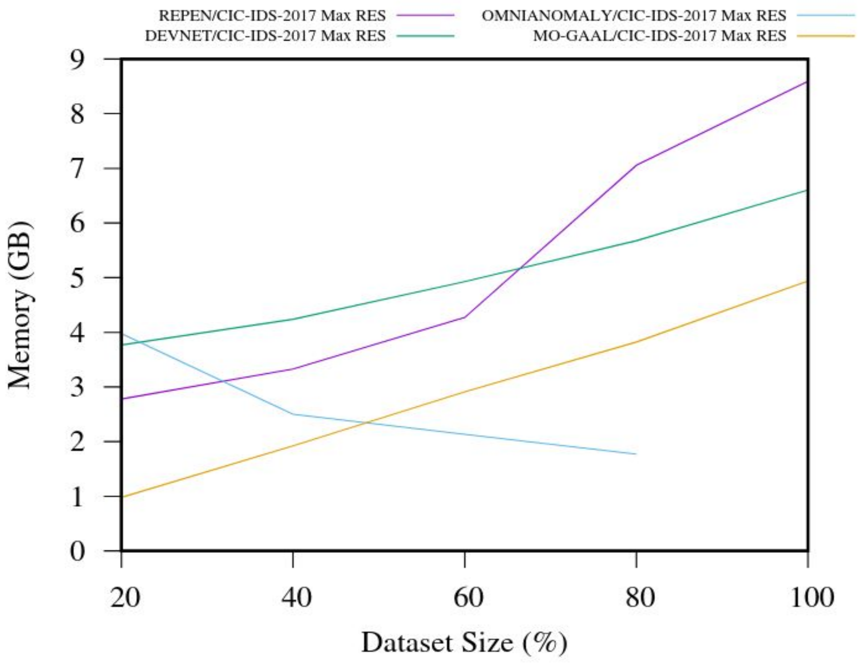


Fig. 7. Memory Usage vs CIC-IDS-2017 Subset Size

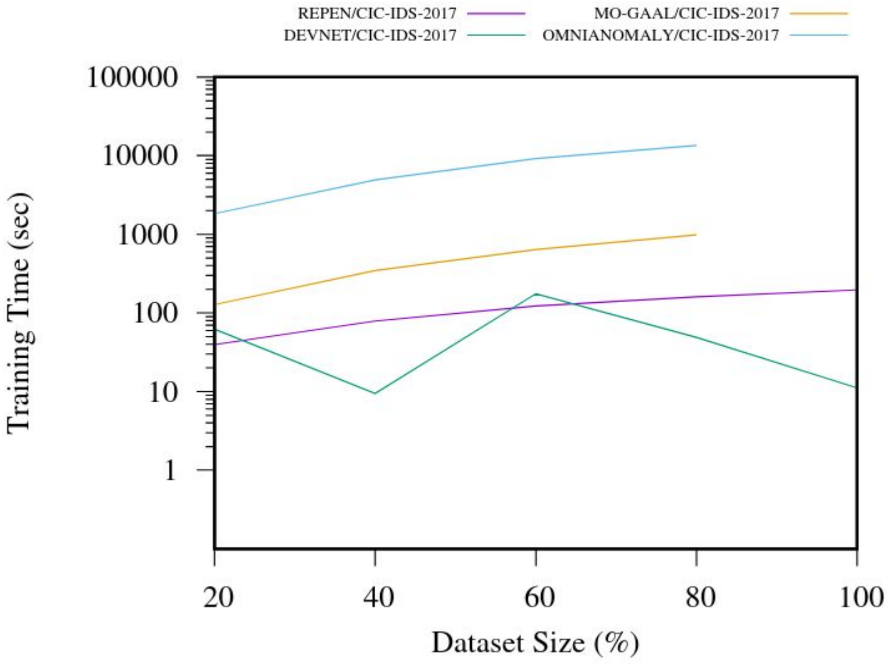


Fig. 8. Training Time vs CIC-IDS-2017 Subset Size

Conclusion and Future Works

- The RNN, OmniAnomaly, outperforms all algorithms in Effectiveness
- REPEN is the quickest with lower Effectiveness
- DevNet has the best trade off of Effectiveness to Performance
- MO-GAAL scales poorly to larger datasets and tends to have the lower Effectiveness

Algorithm	**Effectiveness**	**Performance**
OmniAnomaly	HIGH	Low
REPEN	Low	HIGH
DEVNET	Decent	HIGH
MO-GAAL	Low	Low

References

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

[2] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou et al., “Time-series anomaly detection service at Microsoft,” inProc. of ACM SIGKDD ’19,2019, pp. 3009–3017.

[3] G. Pang, L. Cao, L. Chen, and H. Liu, “Learning representations of ultra high-dimensional data for random distance-based outlier detection,”inProc. of ACM SIGKDD ’18, 2018, pp. 2041–2050. https://github.com/GuansongPang/deep-outlier-detection

[4] G. Pang, C. Shen, and A. van den Hengel, “Deep anomaly detection with deviation networks,” inProc. of ACM SIGKDD ’19, 2019, pp. 353–362. https://github.com/GuansongPang/deviation-network

[5] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, “Robust anomaly detection for multivariate time series through stochastic recurrent neural network,” in Proc. of ACM SIGKDD ’19, 2019, pp. 2828–2837. https://github.com/NetManAIOps/OmniAnomaly

[6] Y. Liu, Z. Li, C. Zhou, Y. Jiang, J. Sun, M. Wang, and X. He, “Generative adversarial active learning for unsupervised outlier detection,”IEEETrans. Knowl. Data Eng., vol. 32, no. 8, pp. 1517–1528, 2019. https://github.com/leibinghe/GAAL-based-outlier-detection