Data-Driven Learning Methods for Network Intrusion Detection Systems

Subjects: Computer Science, Artificial Intelligence

Contributor: Jessica Whitaker

An effective anomaly-based intelligent IDS (AN-Intel-IDS) must detect both known and unknown attacks. Hence, there is a need to train AN-Intel-IDS using dynamically generated, real-time data in an adversarial setting. Furthermore, the lack of real-time data produces potentially biased models that are less effective in predicting unknown attacks. Therefore, training AN-Intel-IDS using imbalanced and adversarial learning is instrumental to their efficacy and high performance. Unfortunately, the public datasets available to train AN-Intel-IDS are ineluctably static, unrealistic, and prone to obsolescence. Furthermore, the lack of real-time data produces potentially biased models that are less effective in predicting unknown attacks. Therefore, training AN-Intel-IDS using imbalanced and adversarial learning is instrumental to their efficacy and high performance.

generative adversarial networks | Data-Driven Learning | Adversarial Learning | Imbalanced Learning

1. Data-Driven Learning (DDL) Methods for Imbalanced Learning

The study in [1] suggested that most approaches that employ methods other than generative adversarial networks (GAN) suffered from data loss or overfitting and proposed the use of GAN to solve the data imbalance instead of resampling and SMOTE techniques to avoid overfitting caused by resampling and class overlapping or noise caused by SMOTE. The GAN generated virtual data similar to the minority class of the imbalanced data. The researchers used the balanced data generated by the GAN, which solved the problem of overfitting and overlapping by specifying the desired resampling rate, to train an anomaly-based detection model based on the random forest (RF) method by increasing the weight of the minority attack class in the intrusion detection evaluation dataset (CICIDS). The GAN-based data augmentation method using resampling boosted the rare classes of the CICIDS 2017 dataset, which constituted less than 0.1% of the dataset, by generating 10,000 data of Bot, infiltration, and heartbleed. The batch size or the number of data learned at a time is 10 for Bot, 1 for the remaining two classes due to tiny data size (less than 30), and 20 for the epoch. The research compared the performance of the GAN-RF model, Single-RF model, and SMOTE-RF model using accuracy, precision, recall, and f-score. The GAN with Random Forest algorithm (GAN-RF) model used GAN for data resampling and RF for classification, standalone Random Forest algorithm (Single-RF) used RF for classification only, and SMOTE with Random Forest algorithm (SMOTE-RF) used SMOTE for data resampling and RF for classification. The GAN-RF performed better than the Single-RF and SMOTE-RF. Using the average score, GAN-RF had an accuracy of

99.83% and f-score of 95.04% compared to 99.19% accuracy and 87.79% f-score for the Single-RF model and 99.51% accuracy and 88.16% f-score for SMOTE-RF.

In addition to augmenting data by producing more examples to balance the minority class examples in the dataset, the GAN can simulate new unforeseen attacks. For example, the researchers in [2] used GAN to augment network traffic represented using imagery to train a Convolutional Neural Network (CNN)-based intrusion detection model, and to simulate unforeseen attacks, researchers refer to this method as GAN 2D imagery CNN or GAN-2CNN for simplicity. However, the two-dimensional image of network flow, produced using two-dimensional mapping techniques, suffered from the unequal representation of normal and abnormal examples. The GAN addressed the imbalanced imagery issue by generating new images of unforeseen attacks, and the CNN classified the 2-D imagery, leading to better predictive accuracy for the GAN-2CNN model. The GAN-based imagery data augmentation method trained the auxiliary classifier GAN (AC-GAN), where it used the generator of the AC-GAN to create new synthetic attacks' images and balance the training dataset. A variant of GAN, AC-GAN, takes a class label and noise as input and generates images [3]. The employed AC-GAN's generator created fake images from a 100-dimensional input random noise vector of a uniform distribution and a two-dimensional one-hot label. The research analyzed the performance of the GAN-based imagery data augmentation using CICIDS17 and AAGM17 datasets with imbalanced traffic data and the full implementation of the model (referred to by the researchers as MAGNETO, a supervised deep learning methodology for learning a robust intrusions model that deals with data imbalance). In addition, it measured the effectiveness of the data augmentation method compared to the SMOTE and adaptive synthetic (ADASYN), proposed by [4] data augmentation methods, and the effectiveness of training the 2D CNN using GAN-augmented data of varying balance sizes. Using a Variant of MAGNETO, i.e., MAGNETO with SMOTE (SMOTE-MAGNETO) and MAGNETO with ADASYN (ADASYN-MAGNETO), the MAGNETO with GAN (GAN-MAGNETO) outperformed the other variants on both datasets in terms of f-score and precision. However, GAN-MAGNETO exhibited a drop in recall, though negligible, using CICIDS17 compared to its performance using the AAGM17 dataset.

Imbalanced data can hinder the proper training of an AN-Intel-IDS and thus its performance. Publicly available datasets, such as the KDD-99 and CIDDS-001, are mostly imbalanced and often contain more 'normal' examples than anomalous examples. The GAN-based IDS (G-IDS) in [5] for securing cyber-physical systems (CPS) addressed the issue of imbalanced data by generating more data to train the IDS, which is a multi-layer artificial neural network. It used the NSL KDD-99 to generate synthetic data that augmented the original data, thus, increasing the distribution of attack examples in the dataset. The proposed G-IDS framework consisted of four modules: database, IDS, controller, and synthesizer.

In addition to the generated synthetic data by the synthesizer module generator, the database module contained real-world intrusion detection data. The controller module decided whether to accept or reject pending data, i.e., synthetic data was data that had not been accepted or rejected by the controller. The GAN, which is part of the synthesizer module, generated the synthetic data. The synthesizer labeled the generated data as pending, due to the uncertainty of the GAN, before sending it to the database module. The researchers used the controller module to evaluate the IDS module twice. First, they trained the IDS module on a hybrid dataset only, i.e., the combined

original and synthetic data already accepted by the controller. Then, they trained the IDS module using a combination of the hybrid and pending datasets. The controller accepted or rejected the pending data based on the IDS performance. By measuring the detection rate for each data class (normal or attack) and comparing it to a pre-established performance threshold, the controller identified the weakly detected classes and sent the data examples to the synthesizer module to generate more examples. The process repeated until a satisfactory IDS performance was obtainable.

Unbalanced distribution of normal and attack examples in a dataset can lead to detection inaccuracies. Further, the detection accuracy of an IDS may vary based on the degree of class imbalance. The method in [6] addressed the issue of imbalanced learning using GAN-augmented data to train a supervised and unsupervised host-based intrusion detection system (HIDS), i.e., Support Vector Machines (SVM) and CNN, respectively. In addition to data augmentation using GAN, the researchers considered data oversampling using SMOTE to evaluate the performance of the GAN-based approach. The SMOTE-based approach over-sampled minority classes from unbalanced data, whereas the GAN-based approach generated the data (similar to the training dataset) itself. The dataset contained system-call trace data represented as a series of integer numbers mappable to system calls made on a Linux OS. Both approaches augmented the abnormal examples by creating new data that was invariably synthetic. The researchers applied both approaches to the pre-processed ADFA-LD dataset and then used SVM and CNN to classify process operation based on system call trace data into normal or malicious behavior. The GAN-based approach to data augmentation was slightly reliable compared to the SMOTE-based approach. In addition, models trained using augmented data had better classification accuracy than models trained using original data. In both cases, models that used GAN-based augmented data performed better. As the number of minority class examples increased by 30%, 50%, 70% and 100%, the classification accuracy and classification performance increased as well. In general, when using data augmentation, CNN performed better than SVM for largedata sizes, whereas SVM showed a better performance for moderate data sizes.

The number of attack examples in the smart home environment, is often smaller than the normal examples, thus creating data imbalance. Therefore, detecting intrusions in a smart home environment requires designing intelligent anomaly-based IDS capable of handling disproportions in the datasets. The researchers in [7] proposed an embedded intrusion detection scheme on the smart homes edge nodes that exploited GAN to reduce the impact of disproportionate datasets, where normal examples are more frequent than attack examples, on the performance of the classifier. The researchers used AC-GAN to generate synthetic data to balance the proportion of normal and attack examples in the UNSW-NB15 training dataset. The researchers converted the network data into images prior to feeding the pre-processed data to the AC-GAN generator. In addition to a noise, the AC-GAN generator took the class label as input to generate synthesized data for the minority attack class. The researchers then combined the synthesized data with the original data to train the classifier. The evaluation results showed that the proposed scheme, which included GAN-based data augmentation, improved the classifier precision for the minor attack class; the precision and recall of the anomaly detection was about 96% and 98%. However, when comparing the precision given the different categories of attacks, the precision of some of the attacks belonging to the majority class declined due to the low quality of the generated synthetic data.

2. DDL Methods for Adversarial Learning

2.1. GAN-Generated Regular Network Traffic

Training an anomaly-based intrusion detection system to detect intrusions in IoT environments is challenging due to the lack of sufficiently-large benign IoT data and the inability to collect IoT data from IoT devices directly due to high scalability and privacy restrictions. In addition, device disparity and activity scarcity make it harder to acquire reliable benign IoT data. The researchers in [8] addressed these issues by proposing a data aggregation and privacy preservation hierarchical approach in which a GAN and an AE cooperated to reconstruct IoT benign data for training a global anomaly-detection IDS and set of local anomaly-detection IDS implemented at the local gateways. The hierarchical method used local GANs implemented at the local IoT networks to generate benign data and a global GAN to reproduce the aggregated benign data, which is double the size of the real data in the local IoT networks. Each local IoT network consisting of a set of local IoT devices and their generated data are aggregated at the global level using a centralized controller. The data generation occurred at the local GANs. First, the generator, which consisted of sequential layers, took a Gaussian Noise with random dimension size as input and generated a series of random outputs. Next, the discriminator combined the generated sample with the benign local data. Then, the generator and discriminator, which had symmetrical structures, were trained simultaneously. Finally, the data from the local generators were aggregated at the centralized AE to reproduce new benign data to train the global AE model that was double the size of the local networks' training data.

The study in [9] provided a tool to solve small data challenges in machine learning, where it is difficult and time-consuming to collect a representative amount of ground truth data. The researchers used GAN to augment sequential IoT data, i.e., time-based sensor readings for predictive maintenance, and generate synthetic household energy consumption data. The generated data was subjectively similar to the original data. Before applying the data to the GAN, the researchers first converted the one-dimensional sequential data into two-dimensional data by exploiting periodic behavior. This was necessary to exploit locality using the GAN and apply CNN methods. In doing so, they aimed at investigating if GAN with two-dimensional convolutions can generate one-dimensional sequential data to enable the use of sophisticated CNN methods such as sharing, pooling, and striding. However, the researchers used a WGAN, adopted from the Keras WGAN implementation, instead of a deep convolutional GAN (DCGAN) due to vanishing gradients during the training of the DCGAN and the replacement of the discriminator's transfer function with a gradient penalty. The researchers trained two of the WGANs; each WGAN was able to generate an energy consumption heatmap similar to the real data. To evaluate the quality of the GAN-generated data, they designed an evaluation workflow where they trained the generator with a subset of all data and used the generator output to train the classifier. The classifier training and evaluation involved using fake and real data, respectively. The data set contains two classes; a minority and majority class comprised of energy consumption data with and without swimming pool data. Further, the researchers combined the WGAN with a convolutional neural network (CNN) and labeled data. The quantitative evaluation using labels revealed that it is possible to generate sequential data from small ground truth data or noise with fixed output size based on data with unique representation. In addition, the evaluation revealed an almost perfect classification for the majority class, where f-score was 0.95–1. However, the minority class f-score was 0.31 indicating poor classification.

Historically, GANs are rooted in image recognition applications where they generate synthetic but realistic images from a given set of images as input. To generate realistic network traffic from GANs, the researchers in [10] proposed a convolutional neural network GAN traffic generator, named PAC-GAN, to generate packet-level network traffic that adheres to network standards and protocols. The proposed network traffic generator used an encoding scheme to convert and map network traffic data into images using image-based matrix representations. The PAC-GAN generated realistic variants of different types of network traffic, such as ICMP pings, DNS queries, and HTTP get requests to transmit through real networks by learning and manipulating the byte values of data packets. The encoding scheme encoded the GAN-generated network traffic and the training network traffic using the matrix.

Unlike the PAC-GAN method, the three synthetic flow-based network traffic generators based on the improved WGAN-GP proposed in [11] indirectly generated new flow-based network traffic based on the CIDDS-001 dataset by learning from the characteristics of previously collected network traffic to mimic the traffic flow. Further, they transformed the categorical attributes of the network traffic, such as protocol, IP addresses, and ports, into continuous attributes for processing by the GAN using three different pre-processing strategies: numeric transformation, binary transformation, and embedding transformation. First, the numeric transformation strategy transformed the IP address and the ports into numerical values. Second, the binary transformation strategy transformed the IP addresses, ports, bytes, and packets categorical values into binary attributes. Finally, the embedded transformation strategy transformed the categorical values of IP addresses, ports, bytes, and packets into vectors or embeddings in an m-dimensional continuous feature space. Three methods based on WGAN-GP, numeric WGAN-GP (N-WGAN-GP), binary WGAN-GP (B-WGAN-GP), and embedding WGAN-GP (E-WGAN-GP) implemented the numeric, binary, and embedding transformation, respectively. Given the processed flow, the WGAN-GP with the two time-scale update rule generated new flow-based network traffic whose quality was evaluated by the researchers using a domain knowledge checks method. Further, the researchers derived several properties to assess whether the generated data are realistic. The evaluation results indicated the ability of the E-WGAN-GP and B-WGAN-GP methods to generate realistic traffic. On the contrary, the N-WGAN-GP did not generate convincing, realistic data. A limitation of the WGAN-GP methods was that they generated single flows instead of sequences of flows.

2.2. GAN-Generated Network Intrusion Traffic

In general, there is a need to evaluate the robustness of intrusion detection systems such as the AN-Intel-IDS and improve their detection. One way to achieve this is by designing malicious traffic that can evade detection in real-world attack scenarios. To that effect, adversarial learning using GANs, such as the framework of GANs in [12], called IDSGAN, performed adversarial black-box attacks to deceive the IDS and to evade detection by generating new malicious traffic based on the original attack traffic. For example, the IDSGAN generated adversarial attacks based on the NSL-KDD by modifying the nonfunctional features in the original attack traffic that enabled it to deceive and bypass the IDS and launch an actual attack. The IDSGAN consisted of a generator, discriminator, and black-box IDS. Similar to [11], the researchers used Wasserstein GAN to create the IDSGAN where the discriminator learned from a black-box IDS that mimicked a real IDS to ensure convergence and instability of the GAN. In addition, the generatorproduced a restricted form of adversarial malicious traffic by modifying limited

features to ensure the validity of the generated adversarial traffic when launching a network attack in reality. shows the training of the IDSGAN framework. The researchers evaluated the capacity and generality of IDSGAN against seven black-box IDS models they formed using different machine learning algorithms and trained using training sets based on the NSL-KDD dataset before the models generated adversarial attacks. Further, they used detection rate (DR) (number of correctly detected attacks divided by the number of all attacks) and evasion increase rate or EIR (one minus the adversarial detection rate divided by the original detection rate) as metrics. Further, they set the goal of the IDSGAN optimization such that a low detection rate and higher evasion increase rate were desirable. The evaluated IDSGAN had a good capacity generating adversarial malicious network traffic resulting in a very low detection rate for the black-box IDS. For non-modified adversarial malicious data, the IDSGAN maintained its evasion capacity.

Focusing on labeled data scarcity or sparsity and cost of data collection and labeling, the researchers in [13] proposed the use of adversarial domain adaptation that leveraged GANs to transfer the knowledge gained from a domain with an adequate and existing training dataset to related but different domains with limited or no new training dataset, for example, transferring knowledge from the traditional network domain to the IoT domain.

Apart from creating a domain invariant mapping between the two datasets, the proposed approach was feature-independent, i.e., it was applicable irrespective of the similarity or differences of the feature spaces of the source and target datasets. In addition, it was universal. Thus, it enabled the re-purposing of deep learning models in the target environment to operate in another environment that used similar data but different data representation using small labeled data from the target environment. Further, it reduced the large amount of labeled data required to train deep learning classifiers. The researchers evaluated the proposed approach using publicly available network intrusion detection (NID) datasets and two scenarios where the source and target datasets had the same feature space (homogeneous DA where data was collected from devices using the same communication protocol). The source and target datasets had a different feature space (heterogeneous DA where data was collected from different types of devices using different protocols), respectively. Further, they used the same dataset, which they split into two parts to account for the source and target datasets, for the homogeneous scenario and two different datasets; one for the target and the other for the source, for the heterogeneous scenario. The proposed approach outperformed the base case, where the researchers used the target dataset to train the deep learning model. The fine-tuning approach for a small dataset was better in terms of deep learning classification accuracy. The researchers used the accuracy and f-score metrics when the source and target dataset had similar features and the f-score only when the features were different. As the number of samples increased, the GAN-DA approach performed better than the base and fined-tuned approaches. However, one issue with the proposed approach was the requirement to use the source and the target datasets, which is challenging to maintain when the source and target data collectors are different.

Similar to the IDSGAN framework, the synthetic GAN (SynGAN) framework in [14] used WGAN-GP to address the complexity and high quality of the generated synthetic network flow. However, unlike IDSGAN, which focused on generating synthetic normal flow, SynGAN applied the WGAN-GP to generate synthetic network attacks using NSL-KDD and CICIDS2017 public datasets. The researchers used the two public datasets to measure the quality

of the generated synthetic packets using a similarity index, i.e., the similarity between the synthesized and real network packets and the DDoS family of attacks to evaluate the SynGAN framework. The SynGAN framework consisted of three modules: the generator, the discriminator, and the evaluator. The researchers used the Gradient Boosting as the evaluator. While the GAN discriminator differentiated between actual and artificial attacks, the evaluator differentiated between actual and artificial packets using a quality measure based on the root mean square error. The preliminary evaluation showed that the SynGAN framework could generate high-quality adversarial attacks with a root mean square error of 0.10, indicating that the proposed framework was incapable of distinguishing between actual and synthesized attacks.

The researchers in [15] focused on efficiently generating adversarial examples with high perceptual quality using a GAN that accelerated adversarial training as defenses. They proposed adversarial GAN (AdvGAN), a conditional adversarial network similar in paradigm to GAN, that once trained instantly generated perturbations for any instances without the need to access the model. The generator of the AdvGAN was a feed-forward network that generated perturbations to create adversarial examples, whereas the discriminator ensured that the generated examples were realistic.

Rather than emulating malicious traffic to evade detection, a GAN can emulate normal traffic to bypass detection. In addition to generating adversarial malicious traffic to evade detection, a GAN can generate network traffic to mimic traffic of a legitimate application to evade detection, thus enabling the malware to adapt to the behavior of the IDS. To that extent, the researchers in [16] used GANs, where the generators and discriminators were recurrent neural networks (RNN) to modify the network behavior of a real malware to mimic the behavior of Facebook chat network traffic. Their primary purpose was to create malware that can avoid detection by ML-based intrusion prevention systems (IPS) that exploit behavioral characteristics to detect malware. The researchers used a threat model to demonstrate their approach that consisted of three components: detector, malware, and server. They deployed the GAN and malware in their laboratory local network, IPS in the router, and the server in the cloud. For each flow, the GAN modified the timing, duration, and request size. The adapted malware was tested if it was being blocked, and the GAN loss was fed back to the GAN. The malware and the blocking of the malware were real.

Using 217 network flows from normal traffic and training the GAN 400 times, the researchers reported a drop in the blocking percentage to zero using enough numbers of epochs and a relatively small dataset, signaling a successful malicious action and the ability of the GAN to modify the malware traffic to avoid detection. The GAN was able to unblock 63.42% of the actions and allow 36.58% of the traffic to go undetected. However, the proposed method operated at the flow level rather than the packet level, and the improvement in the GAN performance was mainly attributed to additional training rather than data augmentation.

3. Hybrid DDL Methods

IoT traffic flow is bidirectional; therefore, methods for generating IoT synthetic data for training IoT intelligent IDS must consider bidirectional flow generation and the relationship between packet-level and flow-level features. The flow is composed of individual packets; thus, the packets' sizes are closely related to the flow duration. To this

purpose, the researchers in [17] leveraged GAN to generate bidirectional flow that mimicked the bidirectional flow generated by actual IoT devices to train and test intelligent IoT IDS that used a set of sparse autoencoders; unsupervised neural networks. Unlike most of the surveyed synthetic data generation methods, which generated either packet-level features or flow-level features, the proposed generator created packet-level features while implicitly learning to comply with the flow-level characteristics to generate synthetic data that looks realistic. The flow-level features included packets' ordering, the total number of packets, and the total duration of the flow (total number of bytes). In contrast, features related to the packet-level included the packets' sizes. In general, packet-level features are describable using different fields of the network layer and the transport layer headers. The generated synthetic bidirectional flow consisted of a sequence of packets and their duration value. The trained generators using Autoencoder/WGAN with weight clipping(WGAN-C) model generated the sequence of packets. The trained mixture density networks (MDN), which took the generated packets sequence as input, determined their duration. The researchers used the WGAN to overcome the issues of GAN generating a sequence of categorical data, i.e., a sequence of packet sizes. The WGAN first converted the sequence of categorical data into a latent vector in a continuous space using the autoencoder and then trained the WGAN on the generated latent space to decode latent vectors into realistic sequences. Further, the researchers assessed the quality of the synthetic bidirectional flow by comparing the distribution of the duration of the synthetic bidirectional flow with that of the actual bidirectional flow and the sequence of packets sizes by using a Google Home Mini Show. The generated data are of quality if their duration is close to the duration of the real bidirectional flow. In both cases, the generated flow had a duration close to the real flow indicating the generated synthetic bidirectional flow was of high quality.

While the G-IDS framework in [5] focused on solving the imbalanced or missing data using adversarial learning, the network intrusion detection (NID) framework in [18] focused on solving the small and imbalanced dataset challenges using statistical learning and adversarial learning. The NID framework tackled both data scarcity and data imbalance by incorporating adversarial learning with statistical learning and exploiting learning using a data augmentation module (DA) consisting of a probabilistic generative model (PGM) and GAN. While the probabilistic model estimated the data feature distribution and generated synthesized intrusions using Monte Carlo methods, the deep generative neural network (DGNN) created high-quality intrusions by augmenting the synthesized data with actual data to provide high-quality training data. In addition, the PGM model initialized the DGNN, thus enabling it to converge on limited intrusion data.

References

1. Lee, J.; Park, K. GAN-based imbalanced data intrusion detection system. Pers. Ubiquit. Comput. 2021, 25, 121–128.
2. Andresini, G.; Appice, A.; De Rose, L.; Malerba, D. GAN augmentation to deal with imbalance in imaging-based intrusion detection. Future Gener. Comput. Syst. 2021, 123, 108–127.

3. Odena, A.; Olah, C.; Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In Proceedings of the 34 th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR: Cambridge, MA, USA, 2017; pp. 2642–2651.
4. Habibo, H.; Yang, B.; Garcia, E.; Shutao, L. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; IEEE: New York, NY, USA, 2008; pp. 1322–1328.
5. Shahriar, M.H.; Haque, N.I.; Rahman, M.A.; Alonso, M., Jr. G-IDS: Generative adversarial networks assisted intrusion detection system. arXiv 2020, arXiv:2006.00676.
6. Kim, K. GAN based augmentation for improving anomaly detection accuracy in host-based intrusion detection systems. Int. J. Eng. Res. Technol. 2020, 13, 3987.
7. Yuan, D.; Ota, K.; Dong, M.; Zhu, X.; Wu, T.; Zhang, L.; Ma, J. Intrusion detection for smart home security based on data augmentation with edge computing. In Proceedings of the ICC 2020, 2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; IEEE: New York, NY, USA, 2020; pp. 1–6.
8. Zixu, T.; Liyanage, K.S.K.; Gurusamy, M. Generative adversarial network and auto encoder based anomaly detection in distributed IoT networks. In Proceedings of the GLOBECOM 2020, 2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; IEEE: New York, NY, USA, 2020; pp. 1–7.
9. Tschuchnig, M.E.; Ferner, C.; Wegenkittl, S. Sequential IoT data augmentation using generative adversarial networks. In Proceedings of the ICASSP 2020, 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; IEEE: New York, NY, USA, 2020; pp. 4212–4216.
10. Cheng, A. PAC-GAN: Packet generation of network traffic using generative adversarial networks. In Proceedings of the 2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 17–19 October 2019; IEEE: New York, NY, USA, 2019; pp. 0728–0734.
11. Ring, M.; Schlör, D.; Landes, D.; Hotho, A. Flow-based network traffic generation using generative adversarial networks. Comput. Secur. 2019, 82, 156–172.
12. Lin, Z.; Shi, Y.; Xue, Z. IDSGAN: Generative adversarial networks for attack generation against intrusion detection. arXiv 2021, arXiv:1809.02077.
13. Singla, A.; Bertino, E.; Verma, D. Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation. In Proceedings of the 15th ACM Asia Conference on Computer and Communications Security, Ser. ASIA CCS ’20, Taipei Taiwan, 5–9 October 2020; ACM: New York, NY, USA, 2020; pp. 127–140, ISBN 978-1-4503-6750-9/20/10.

14. Charlier, J.; Singh, A.; Ormazabal, G.; State, R.; Schulzrinne, H. SynGAN: Towards generating synthetic network attacks using GANs. arXiv 2019, arXiv:1908.09899.
15. Xiao, C.; Li, B.; Zhu, J.Y.; He, W.; Liu, M.; Song, D. Generating adversarial examples with adversarial networks. arXiv 2019, arXiv:1801.02610.
16. Rigaki, M.; Garcia, S. Bringing a GAN to a knife-fight: Adapting malware communication to avoid detection. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), IEEE Symposium on Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; IEEE: New York, NY, USA, 2018; pp. 70–75.
17. Shahid, M.R. Deep Learning for Internet of Things (IoT) Network Security. Ph.D. Thesis, Institut Polytechnique de Paris, Palaiseau, France, 2021.
18. Zhang, H.; Yu, X.; Ren, P.; Luo, C.; Min, G. Deep adversarial learning in intrusion detection: A data augmentation enhanced framework. arXiv 2019, arXiv:1901.07949.

Retrieved from https://encyclopedia.pub/entry/history/show/47697