Anomaly Dataset Augmentation Using the Sequence Generative Models

1st SungUk Shin
Agency for Defense Development
Daejeon, Republic of Korea
ssw1419@add.re.kr

2st Inseop Lee
Agency for Defense Development
Daejeon, Republic of Korea
dlstjq0711@add.re.kr

3st Changhee Choi
Agency for Defense Development
Daejeon, Republic of Korea
changhee84@add.re.kr

***Abstract*—In cyberspace, anomalies including intentional attacks grow up in their size and diversity. Although using the Intrusion Detection System (IDS) as a solution is helpful to some degree, there is an unsolved problem; the low performance of IDS due to lack of enough attack data. Recent approaches to solving this problem use an unsupervised deep learning-based technique called Generative Adversarial Networks (GANs). Because GAN variants show great performance in image augmentation, some research tries to apply GANs to cyberspace by domain conversion from binary to image. However, the attribute of cyberspace benchmarks is different from that of images. In this paper, we propose using sequence-based generative models such as Sequence Generative Adversarial Nets (SeqGAN) and Sequence to Sequence (Seq2Seq) to augment the ADFA-LD dataset, a sequence call based benchmark. Experimental results show that the performance is better when training ADFA-LD with augmented data from SeqGAN and Seq2Seq than training only the original dataset.**

***Index Terms*—SeqGAN, Seq2Seq, cyber dataset augmentation, ADFA-LD**

I. Introduction

Attacks in cyberspace are becoming diverse and widespread today. People are trying to prevent an attack from inside and outside by using an Intrusion Detection System (IDS). However, since attacks are becoming more diverse and wider over time, it is important to update the IDS periodically.

To stabilize the IDS, the collection of data becomes important and the collected data becomes an important asset. According to existing researches, data collection methods include directly collecting from users, automatically collecting using templates, or creating new data by processing existing data [1], [2]. Considering that it is data for learning IDS, it is disadvantages that the amount of data may be limited or take a long time when directly collecting or collecting using templates. If the amount of attack data is small, there is a possibility of biasing the normal data, which may cause the model to become unstable. Therefore, we can solve this problem by processing and creating data.

Proposed in 2014, Generative Adversarial Networks (GAN) is an innovative technology for creating new data from existing data [3]. This algorithm is a deep learning technique in which a generator generates data and a discriminator determines the answer so that it can compete with each other. As an representative application of GAN, synthesizing real-like human faces has evolved and has begun to be applied to other fields. In 2016, Sequence Generative Adversarial Nets (SeqGAN) was made by combining GAN with reinforcement learning for augmentation of sequences, which are not in image domains [4]. There is a recent paper about generating opcode of malicious code using SeqGAN [5].

There is also a Sequence to Sequence (Seq2Seq) technique that is not a GAN in sequence data generation technique [6]. The Sequence to Sequence (Seq2Seq) has an encoder that analyzes input data and a decoder that generates data, and the model works well in translator or chatbot. In recent, Bidirectional LSTM (BiLSTM) [7] and attention techniques [8] show better performance.

For this reason, ADFA-LD dataset was augmented by using Cycle-Consistent Adversarial Networks (CycleGAN), and comparing performance with building an Artificial Neural Network (ANN) for detecting abnormalities appeared [9], [10]. However, it is more appropriate to treat the system call as a sequence rather than a 2D image, and the ANN used in this paper shows that the performance is increased because of the unstable performance graph. So it is an unpractical experimental result.

We propose applying both SeqGAN and Seq2Seq models to augment the data. To compare which data is more effective, we use representative machine learning algorithms for intrusion detection [11], [12]. we evaluated the algorithms before and after training augmented dataset, in addition to original dataset. Experimental results show that adding augmented data to training can improve the IDS performance. Among various cyber benchmarks, the ADFA-LD is selected [13], [14]. Although ADFA-LD is not the most recent data set, it was selected because it is a well-gathered data set that is suitable for the data of abnormal detection systems based on the host using a system call.

There are some researches about intrusion detection algorithms using ADFA-LD. An approach using deep learning shows 92.1% accuracy with five DNN layers and embedding layer [15]. In an aspect of feature processing, an experiment shows 96.23% accuracy with 5-gram and top 25% high frequency of features [16]. In this paper, we compare the performance with 3-gram and top 3% high frequency.

Section 2 provides backgrounds on datasets, SeqGan, Seq2Seq and intrusion detection algorithms. Section 3 discusses what the paper suggests. The experimental results and

978-1-7281-4550-1/19/$31.00 ©2019 IEEE
DOI 10.1109/ICMLA.2019.00190

analysis are discussed in Section 4, and finally, Section 5 concludes with a conclusion.

II. Background

A. ADFA-LD dataset

The ADFA dataset is divided into the Linux and the Windows version, but only the Linux version (ADFA-LD) is used in this paper. The ADFA-LD has three directories; Attack, Training, and Validation. Attack has 746, Training 833, and Validation 4373 files. Both training and validation are normal behavior data. Attack actions include 6 attacks of Adduser, HydraFTP, HydraSSH, JavaMeterpreter, Meterpreter, and WebShell. There are 10 directories in each attack directory. In ADFA-LD, the attack behavior dataset is much smaller than the normal behavior dataset. To solve the data imbalance problem, we augment only the attack data for balanced train dataset.

B. SeqGan

SeqGAN applies reinforcement learning to the generator due to using sequence instead of an image. The generator uses Long Short-Term Memory (LSTM) to generate the sequence. Since the reward value of the intermediate sequence cannot be obtained, the generator applies to Monte Carlo search [17] to predict the final sequence and the reward value is used. ADFA-LD data is a dataset that lists system calls. System call depends on the operation of the process and can be thought of as a sequence because it is affected by the previous system call. When learning SeqGAN generator, we learned one attack dataset with one sequence. Fig. 1 shows a discriminator and a generator. Discriminator classifies the true data and the generated data. The generator generates the data and makes the data better using reward value.

C. Seq2Seq with attention

Seq2Seq is a model for generating sequences and often used for a translator. The Seq2Seq has an encoder and a decoder. The encoder learns hidden layers through the Recurrent Neural Network (RNN) model, and the decoder learns hidden layers through the context for the input and output. When encoding and decoding with only one context vector, the context vector carries the burden. To solve this problem, attention concept is proposed [18], [19]. At the decoder output stage, the attention weight is calculated based on the weight of the encoder. The attention weight is a probability vector in which words are related to each other. Seq2Seq using attention has the shape of Fig. 2 and most of the known models are based on Seq2Seq.

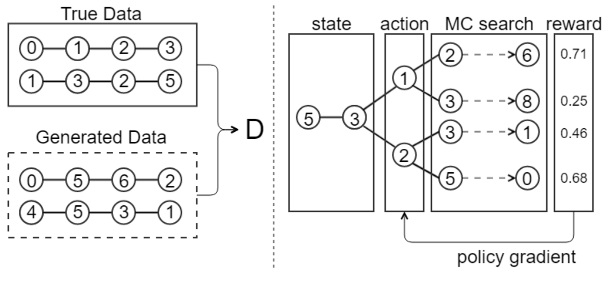


Fig. 1: SeqGAN model. Left is a discriminator, and right is generator.

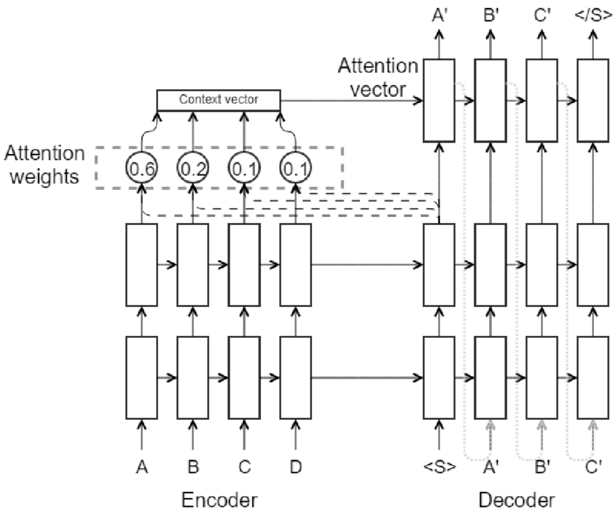


Fig. 2: Seq2Seq with the Attention Model

D. Intrusion detection algorithm

Fig. 3 explains why data collection is important in IDS. The solid line refers to population data(true data) and the boundaries created by IDS when used true data. The dashed line refers to collected data(observed data) and the boundaries created by IDS when used collected data. Observed normal data is similar to true normal data because it is easy to collect, but abnormal data is difficult to collect, so only a part can be collected. This does not matter when the abnormal data are distributed, but IDS can have difficulty in detecting when the biased data is collected as shown in Fig. 3.

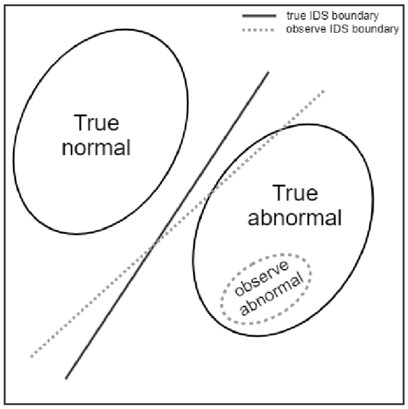


Fig. 3: True and collected data and IDS boundary.

Therefore, we evaluate the intrusion detection algorithm to verify that the dataset is well-made effectively. If the dataset is well-generated, the IDS performance will be similar or improved. If it is not well-generated, the performance will be adversely affected and the performance will be reduced.

However, preprocessing is required to apply the ADFA-LD dataset to the intrusion detection algorithm. In this paper, preprocessing was performed using the feature of 3-gram frequency. First, we separate the data into 3-grams and then select only the 3% of the highest frequency of 3-gram words in the entire dataset. This 3% of words are the feature and the frequency is the data.

III. Challenges and solutions

In 2018, Milad Salem suggested augmenting cyber data by CycleGAN to solve data imbalance problem. However, because proposed approaches have some limitations, this section describes recommended solutions.

- The first, cyber data was augmented by CycleGAN, which is a generative model for image domain just as it replaces an apple image to an orange image. Because image is affected by the surrounding pixels, CycleGAN uses 2D convolution. However, a generative model for image domain is not suitable for ADFA-LD, a collection of sequence-based system calls.
- The second, since a pixel of the image is mapped to 8 bits, it has a value from 0 to 255. However, because the maximum value of system call in ADFA-LD is 340, the system call cannot be expressed as an 8-bit. As a result, in conversion to an image, some data information can be lost.
- The last, performance was too low in Milad Salem's experiments. In evaluation, the result AUC was 55.06, 68.89, 71.30. When the AUC value was calculated after augmentation with CycleGAN, the value dropped to around 50. AUC closed to 50 means a little better than taking one of the two. Because of this, the intrusion detection model was unstable, rather than a good use of CycleGAN. It shows that augmented data leads to sometimes better and sometimes worse results.

Therefore, we would like to solve these issues by suggesting the following.

- To solve the first and the second issues, we used sequence generative models such as SeqGAN and Seq2Seq. The advantage of a sequence is no maximum value, which means values bigger than 255 can be input without preprocessing. We applied SeqGAN and Seq2Seq to the sequence based benchmark called ADFA-LD.
- To solve the third issue, extensive simulations are done for intrusion detection algorithms. Due to the difficulty of evaluation with only one algorithm, various machine learning algorithms are used in analysis with 3-gram frequency. For implementation, we use scikit-learn, one of the most widely used python machine learning library. For intrusion detection algorithms, "all-estimator" of scikit-learn is used. It includes many algorithms such as ensemble-based algorithms (e.g., AdaBoost, GradientBoosting, RandomForest, and ExtraTree), naive bayes-based algorithm (e.g., BernoulliNB, GaussianNB, and ComplementNB) and deep learning-based algorithm (e.g., MLP). In addition to all-estimator, we also use XGBoost, which is state-of-the-art algorithm and shows great performance [20].
  In addition, all parameters in each estimator are set to default values.

IV. Experiment and Analysis Results

For the experiment, we divide the training and test set to train and evaluate the intrusion detection model.

First, if we use only a given ADFA-LD dataset without augmentation the attack data, the distribution of the dataset is as follows. As the training data of IDS, the normal dataset is used in the training directory of ADFA-LD which has 833 files. Each of 6 attacks is divided into 10 directories, and 1,2 directory is used by training attack data which has 143 files. As test data of IDS, 4372 normal data are used in the validation directory of ADFA-LD, and attack dataset has 603 files except training data.

Second, when the attack data is augmented, the distribution of the data set is as follows. Distribution of normal data proceeds in the same as in the previous case. Attack data proceeds in the same as the previous case and adds data generated through SeqGAN and Seq2Seq to IDS learning data. The data for training SeqGAN and Seq2Seq is used with 143 training attack files.

A. SeqGAN

In the case of models such as RNN, there is a disadvantage that the input length is longer, the speed and performance are lower. In the case of SeqGAN, the Monte Carlo search must be applied, so using long sequences increased the path to search. The average sequence length of ADFA-LD attack data is about 426 which is very long. It creates a SeqGAN model with an average length of 426 lengths, it can be very time consuming and difficult to tune and verify. Because of this, we slice the ADFA-LD dataset into pieces and generate data. The purpose of this paper is to improve the performance of the intrusion

Algorithm 1 SeqGAN with ADFA

```
Require: data processing
Pre-train G and D
repeat
    for g-steps=1 or 3 do
        generate 32 or 64 sequences using G
        Update generator parameters via policy gradient
    end for
    for d-steps do
        Train discriminator
    end for
until SeqGAN converges
```

TABLE I: IDS using SeqGAN AUC result

kind of classifier	without augment	number of Monte Carlo search, g-step			
		32,3	64,3	32,1	64,1
AdaBoostClassifier	0.9008	0.8999	0.8968	0.9032	0.9036
BaggingClassifier	0.9145	0.9219	0.9173	0.9186	0.9216
BernoulliNB	0.8823	0.8883	0.8872	0.8910	0.8925
CalibratedClassifierCV	0.9124	0.9118	0.9181	0.9178	0.9214
ComplementNB	0.8951	0.8970	0.8983	0.8991	0.9019
DecisionTree	0.8712	0.8717	0.8742	0.8674	0.8761
ExtraTreeClassifier	0.8679	0.8532	0.8726	0.8695	0.8718
ExtraTreesClassifier	0.9231	0.9264	0.9268	0.9270	0.9283
GaussianNB	0.8041	0.8666	0.8651	0.8873	0.8913
GradientBoosting	0.8924	0.9040	0.9045	0.9051	0.9042
KNeighborsClassifier	0.8910	0.8911	0.8913	0.8918	0.8924
LDA	0.9181	0.9140	0.9166	0.9157	0.9214
LogisticRegression	0.9251	0.9242	0.9268	0.9268	0.9286
LogisticRegressionCV	0.9247	0.9253	0.9256	0.9286	0.9274
MLPClassifier	0.9325	0.9371	0.9343	0.9374	0.9366
MultinomialNB	0.8957	0.8974	0.8986	0.8994	0.9022
RandomForest	0.9244	0.9285	0.9267	0.9322	0.9313
XGBoost	0.9023	0.9009	0.9030	0.9047	0.9036

TABLE II: IDS using SeqGAN F1-score result

kind of classifier	without augment	number of Monte Carlo search, g-step			
		32,3	64,3	32,1	64,1
AdaBoostClassifier	0.6103	0.5773	0.5841	0.5911	0.5939
BaggingClassifier	0.6136	0.5909	0.5941	0.5904	0.5886
BernoulliNB	0.5255	0.5233	0.5325	0.5379	0.5399
CalibratedClassifierCV	0.6116	0.6267	0.6239	0.6388	0.6271
ComplementNB	0.5041	0.4923	0.4937	0.5001	0.5090
DecisionTree	0.5863	0.5594	0.5691	0.5472	0.5697
ExtraTreeClassifier	0.5800	0.5088	0.5590	0.5513	0.5591
ExtraTreesClassifier	0.6284	0.6065	0.5869	0.6057	0.5969
GaussianNB	0.3732	0.5084	0.4996	0.5685	0.5868
GradientBoosting	0.6421	0.6284	0.6331	0.6292	0.6353
KNeighborsClassifier	0.5222	0.5036	0.4938	0.4914	0.5051
LDA	0.6095	0.5768	0.5875	0.5810	0.5922
LogisticRegression	0.6150	0.5888	0.5871	0.5913	0.5953
LogisticRegressionCV	0.6173	0.5851	0.5902	0.6036	0.5969
MLPClassifier	0.6160	0.6048	0.5909	0.6037	0.5976
MultinomialNB	0.5081	0.4955	0.4960	0.5028	0.5116
RandomForest	0.6224	0.5941	0.5891	0.5988	0.5959
XGBoost	0.6366	0.6242	0.6316	0.6243	0.6282

TABLE III: IDS using Seq2Seq AUC result

kind of classifier	without augment	input length	
		20	40
AdaBoostClassifier	0.9135	0.9030	0.8956
BaggingClassifier	0.8154	0.8309	0.8456
BernoulliNB	0.8373	0.8312	0.8315
CalibratedClassifierCV	0.8709	0.6566	0.6716
ComplementNB	0.8546	0.8546	0.8546
DecisionTreeClassifier	0.7573	0.6428	0.6471
ExtraTreeClassifier	0.6158	0.7346	0.7173
ExtraTreesClassifier	0.8402	0.8402	0.8085
GaussianNB	0.8181	0.7651	0.7655
GaussianProcessClassifier	0.5528	0.5528	0.5528
GradientBoostingClassifier	0.9167	0.9165	0.9142
KNeighborsClassifier	0.8266	0.8222	0.8240
LinearDiscriminantAnalysis	0.6726	0.6609	0.6713
LogisticRegression	0.8949	0.8892	0.8926
LogisticRegressionCV	0.8865	0.8660	0.8670
MLPClassifier	0.9065	0.8995	0.8989
MultinomialNB	0.8544	0.7147	0.7129
QuadraticDiscriminantAnalysis	0.5376	0.5184	0.5183
RandomForestClassifier	0.7720	0.8195	0.8342
XGBoost	0.9204	0.9204	0.9204

TABLE IV: IDS using seq2seq F1-score result

kind of classifier	without augment	input length	
		20	40
AdaBoostClassifier	0.5871	0.5477	0.5661
BaggingClassifier	0.5495	0.5552	0.5835
BernoulliNB	0.4476	0.4127	0.4143
CalibratedClassifierCV	0.3790	0.4328	0.4451
ComplementNB	0.4835	0.4835	0.4835
DecisionTreeClassifier	0.5006	0.3031	0.3057
ExtraTreeClassifier	0.2824	0.4878	0.4545
ExtraTreesClassifier	0.5453	0.5717	0.5641
GaussianNB	0.4994	0.4989	0.4986
GaussianProcessClassifier	0.2757	0.2757	0.2736
GradientBoostingClassifier	0.5905	0.5691	0.5786
KNeighborsClassifier	0.4425	0.4265	0.4325
LinearDiscriminantAnalysis	0.3100	0.3043	0.3110
LogisticRegression	0.5642	0.5461	0.5565
LogisticRegressionCV	0.5134	0.4611	0.4570
MLPClassifier	0.6029	0.5898	0.6050
MultinomialNB	0.4814	0.3717	0.3736
QuadraticDiscriminantAnalysis	0.1433	0.0750	0.0749
RandomForestClassifier	0.5606	0.5679	0.5555
XGBoost	0.5888	0.5888	0.5888

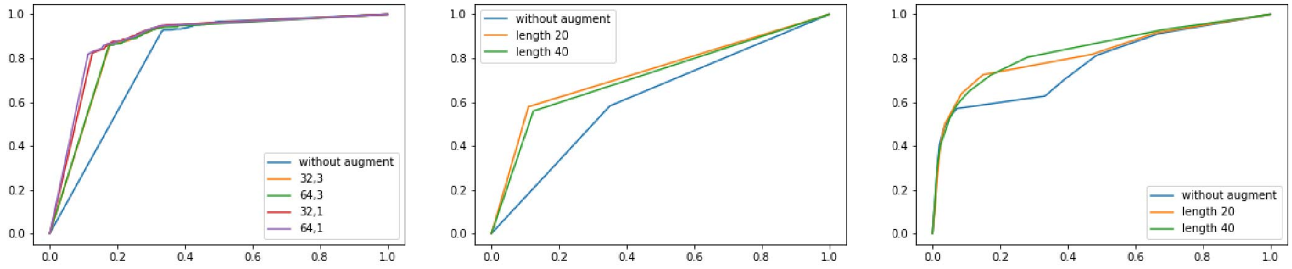
detection algorithm by generating data, it can be considered to generate meaningful data even if only part of the attack is generated without generating the entire attack sequence.

if the length of the sequence is set too short, it may generate meaningless sequences that are not part of the attack. Since the shortest sequence length in the ADFA-LD data set is 124, a smaller value of 25 is set as the sequence length in this paper. In the case of generating data through SeqGAN, since the output has a sequence of 25 lengths, all data for learning and evaluating the intrusion detection algorithm are also divided into 25 sequences. Divided by a sequence of length 25, the train normal is divided into more than 10,000 sequences, and the train attack is divided into approximately 2500 sequences. So about we generate 4000 data through SeqGAN.

To learn the SeqGAN model, we first learned G and D through pre-training like SeqGAN paper, and then proceeded further learning. In this case, only the number of searches with Monte Carlo and the number of g-steps are changed. According to SeqGAN paper, the size of g-step and d-step is set similarly. Since the pre-train is sufficient when the algorithm is executed, the while training is performed only 20 times to generate data.

As a result, the results of learning IDS without augmentation and four cases of augmentation data by changing the number of searches with Monte Carlo and g-step were obtained.

Table I and Table II show the results. Although most of the performance does not change much, the red box in the Table I shows that the performance is increased significantly. Also, the Naive Bayes model shows a slight performance improvement.



(a) GaussianNB ROC curve using SeqGAN (b) ExtraTree ROC curve using Seq2Seq (c) Randomforest ROC curve using Seq2Seq

Fig. 4: ROC curve using sequence generative models

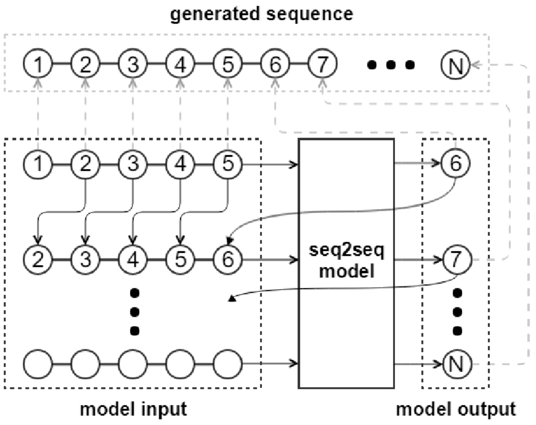


Fig. 5: How to generate data using Seq2Seq

B. Seq2Seq

Seq2Seq requires input data and output data, unlike SeqGAN which only requires real data. Therefore it is necessary to define input data and output data. In this paper, we create a window and predict the next word. For example, if the sequence is (1, 1, 2, 3, 2, 4) and the window size is set to 3, a learning pair of [(input), (output)] can be [(1, 1, 2), (3)], [(1, 2, 3), (2)], [(2, 3, 2), (4)]. However, if a single system call is considered to be a single word, the set of words becomes smaller and a specific word may be repeated repeatedly. Therefore n-gram is used and 5-gram is chosen to make the set of words the appropriate size. The input uses 20 and 40 sequences of 5-gram words, and the output is 1 sequence of 5-gram words. As shown in Fig. 5, when the first seed value is selected, the next word is predicted. And the data is generated by proceeding to predict the word by attaching the latter part of the seed and the predicted word and stop until the output word indicating the end.

Table III and Table IV show the results and the red box is a more performance model than without augmentation. After augmentation of the data through Seq2Seq, using it to train IDS is not effective for all algorithms. However, for some algorithms, we can see that it has an effective result. Shown in Fig. 4b and Fig. 4c, in the case of ExtraTreeClassifier, AUC and F1-score increased significantly regardless of the input length and there is some improvement in RandomforestClassifier.

C. Analysis of SeqGAN and Seq2Seq results

Even if using a simple model, if train with a good dataset we can increase the model's accuracy. When using SeqGAN, the word was divided into length 25, and Seq2Seq used the data without preprocessing, so the performance is better in SeqGAN IDS when without any data augmentation. In SeqGAN, some classifier models don't work because the data is divided into tens of thousands, so it takes a long time.

In the case of SeqGAN, setting the number of Monte Carlo search to 64, g-step to 1, all AUC is increased for all models with augmentation data. In other cases, AUC is increased for most models and only a small number of models decreased. In the case of Seq2Seq, there are some increased models, but others are poor. However, due to the different preprocessing of the data, it is difficult to compare the two models directly and it would be advantageous to use it in such situations.

- There is no time to tune the parameters well, and SeqGAN is expected to be more advantageous when trying to improve only a small amount of performance.
- There is plenty of time to tune the parameters well, and Seq2Seq will be more advantageous when it comes to significantly improve performance.

In the case of SeqGAN, there is not much work to be done or parameter tuning because there is not much research in recent years. On the other hand, Seq2Seq is currently being actively researched and there is a possibility of defining inputs and outputs or adding more techniques to the model.

V. Conclusion

In this paper, we augmented ADFA-LD, one of the cyber datasets, by using a sequence-based deep learning model and evaluated it through IDS. As the data generation model, we used the SeqGAN model which processes the sequence by modifying the most famous model GAN, and the Seq2Seq model which performs well in translator and chatbot. Since it is difficult for humans to see and judge the data augmented above, we apply it to IDS a system that utilizes cyber data and evaluates its performance.

There are some specific models that are effective when learning IDS using the augmentation data and then evaluating how well IDS learned. In designing this paper, effective models existed although the data were preprocessed simply and focused on specific models using various models. It has also been shown that promoting data through SeqGAN can generally be effective. In the case of Seq2Seq, there are models with increased performance, but there were certain models with no increased performance. To do this, if tune the parameters well, use a new model that can be applied to Seq2Seq or organize the data input and output more systematically, the model will also be able to generate good enough data to train IDS well.

References

[1] F. Skopik, G. Settanni, R. Fiedler, and I. Friedberg, "Semi-synthetic data set generation for security software evaluation," in *2014 Twelfth Annual International Conference on Privacy, Security and Trust*. IEEE, 2014, pp. 156–163.

[2] H. G. Kayacik, M. Heywood, and N. Zincir-Heywood, "On evolving buffer overflow attacks using genetic programming," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 2006, pp. 1667–1674.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[4] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[5] C. Choi, S. Shin, and I. Lee, "Opcode sequence amplifier using sequence generative adversarial networks," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 2019.

[15] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41 525–41 550, 2019.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[7] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXov:1508.01991*, 2015.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[9] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. Courville, "augmented cyclegan: Learning many-to-many mappings from unpaired data," *arXiv preprint arXiv:1802.10151*, 2018.

[10] M. Salem, S. Taheri, and J. S. Yuan, "Anomaly generation using generative adversarial networks in host based intrusion detection," *arXiv preprint arXiv:1812.04697*, 2018.

[11] D. Wagner and P. Soto, "Mimicry attacks on host-based intrusion detection systems," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*. ACM, 2002, pp. 255–264.

[12] H. G. Kayacik, A. N. Zincir-Heywood, M. I. Heywood, and S. Burschka, "Generating mimicry attacks using genetic programming: a benchmarking study," in *2009 IEEE Symposium on Computational Intelligence in Cyber Security*. IEEE, 2009, pp. 136–143.

[13] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system call patterns," *IEEE Transactions on Computers*, vol. 63, no. 4, pp. 807–819, 2013.

[14] G. Creech, "Developing a high-accuracy cross platfrom host-based intrusion detection system capable of reliably detecting zero=day attacks." Ph.D. dissertation, University of New South Wales, Canberra, Australia, 2014.

[16] B. Subba, S. Biswas, and S. Karmakar, "Host based intrusion detection system using frequency analysis of n-gram terms," in *TENCON 2017-2017 IEEE Region 10 Conference*. IEEE, 2017, pp. 2006–2011.

[17] G. M. J. Chaslot, M. H. Winands, H. J. V. D. HERIK, J. W. Uiterwijk, and B. Bouzy, "Progressive strategies for monte-carlo tree search," *New Mathematics and Natural Computation*, vol. 4, no. 03, pp. 343–357, 2008.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[19] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[20] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 2016, pp. 785–794.