6G Network Traffic Intrusion Detection Using Multiresolution Auto-encoder and Feature Matching Discriminator

Yuhai Li[1(✉)], Yuxin Sun[1(✉)], Dong He[2], and Liang Xi[2]

[1] Science and Technology on Electro-Optical Information Security Control Laboratory, Tianjin 300300, China
liyuhai.cn@qq.com, sunyuxin@tju.edu.cn

[2] School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

Abstract. With the development of 6G technology, security and privacy have become extremely important in the face of larger network traffic bandwidth. An effective intrusion detection system can deal with the network attacks. Deep learning has been developed in the field of intrusion detection, which can identify normal and abnormal traffic. However, existing methods cannot guarantee good performance in accuracy and efficiency. In this paper, based on the autoencoder and generative adversarial network, the multiresolution autoencoder is adopted in the network traffic feature extraction, which can obtain different encoding lengths and guarantee better data reconstruction. In addition, we add an extra feature matching loss to encourage the discriminator to get more discriminative information from the reconstructed samples. Our experimental results on the CIC-IDS2018 dataset indicates that compared with autoencoder and generative adversarial network, our model can effectively improve the detection accuracy and can be applied to 6G network traffic security detection.

Keywords: 6G · intrusion detection · autoencoder · generative adversarial network

1 Introduction

6G will bring network software and cloudification into network intelligence, revolutionizing wireless networks from connected things to "connected intelligence" [1, 2]. Hence, artificial intelligence (AI) technology plays an irreplaceable role in the whole network, especially for network security. With the rapid improvement of data transmission rate and coverage, people are seriously worried that the security and privacy of 6G may be worse than those of previous generations. For example, if the safety of the communication devices is not guaranteed, the probability of personal information leakage will be increased. Network attacks may cause irreparable losses and bring severe threats to private property, personal reputation, and even life. In addition, many criminals may use the excellent AI technologies for network attacks and network monitoring [3]. Therefore, how to ensure its security and privacy will be the key to 6G.

A. Li et al. (Eds.): 6GN 2022, LNICST 504, pp. 208–218, 2023.
https://doi.org/10.1007/978-3-031-36011-4_18

Artificial intelligence technology can detect and identify network intrusions. AI algorithms can be used to learn and extract network intrusion features based on the traffic behavioral characteristics, and identify attacks such as distributed denial of service (DDoS) [4]. An Intrusion Detection System (IDS) system is a network security device that contains multiple methods and mechanisms for identifying network intrusions. Traditional machine learning methods [5] have been proven to be effective in identifying important features and patterns in network traffic and identify network attacks skillfully. These methods can be used in IDS. However, machine learning methods fail to deal with huge datasets, and it has been demonstrated that the performance of machine learning does not perform well in detecting intrusions and network attacks when network nodes are extremely dispersed [6]. In recent years, deep learning has been one of the most popular data mining techniques. Many deep learning methods continue to emerge for intrusion detection. They can automatically extract data features to identify traffic types, and handle intrusions and network attacks well [7]. Compared with traditional machine learning methods, deep learning methods can further improve the accuracy of intrusion detection, making the IDS applications more flexible and efficient.

The autoencoder is shown in Fig. 1. Autoencoder mainly compresses the input data, x, into the latent representation space and reconstruct it. Its purpose is to ensure the data consistency between the output, $\hat{x}$, and the input, x. There are many autoencoder-based anomaly detection methods. For example, Sakurada M et al. [8] use an autoencoder with nonlinear dimension reduction in the anomaly detection task.

In addition, as shown in Fig. 2, GAN (Generative Adversarial Network) was proposed by Goodfellow I et al. [9]. GAN mainly trains both a generator, G, and a discriminator, D. The purpose of G is to learn the data distribution, while D is to distinguish between real data and fake data generated by G. G and D can reach Nash equilibrium through continuous iterations. The network can be combined with an autoencoder and applied to traffic attack detection to increase identification accuracy. Therefore, we proposes unsupervised intrusion detection using the autoencoder and GAN by improving the diversity of data and the output structure of the discriminator.

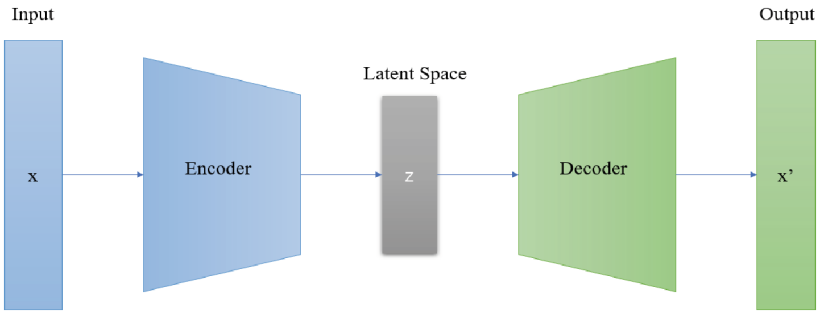

Fig. 1. The autoencoder framework

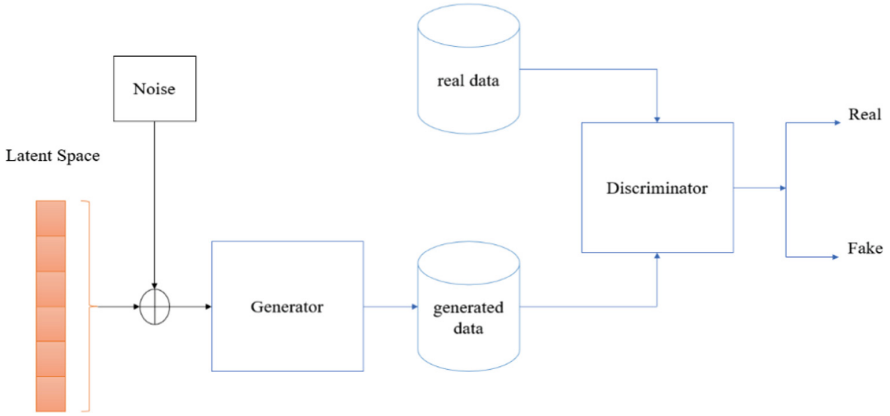

Fig. 2. The generative adversarial network framework

2 Related Work

2.1 Security and Privacy in 6G Networks

5G networks have already coped with many business application scenarios, which can facilitate the connection of more devices and provide services for various devices at the same time. The supported devices include smartphones, IoT devices, etc. However, the network security problems remain the biggest obstacles and will continue to exist in 6G.

Existing researches on 6G security and privacy include 6G wireless network [10], IoT [11] (such as wireless sensor network [12], Vehicle Networks [13, 14]), cellular networks [15], etc. With the development of machine learning and deep learning technologies, many related methods have been able to achieve adequate protection and monitoring [16].

2.2 6G IoT Intrusion Detection

A large amount of IoT traffics are transmitted and communicated between IoT devices [17], and these information-carrying devices and communications are vulnerable to cyberattacks to compromise IoT communications. Therefore, to protect the security and privacy in 6G IoT network, network intrusion detection is indispensable [18]. An Intrusion Detection System (IDS) is a technology designed to monitor the network events in time. The intrusion detection system constructs a normal behavior pattern. If the behavior of some network event does not match the normal pattern, it is classified as an attack.

Machine Learning Methods. Machine learning (ML) techniques are effective at extracting important information from network traffic to identify cyberattacks. To prevent the exposures of information and control flow, Yang et al. [19] proposed an effective KNN classification algorithm, which supports large-scale data classification on distributed servers. Ravi et al. [20] proposed a machine learning method to detect network

attacks by combining the *k*-means algorithm with the repeated random sampling technique of data clustering. In addition, Ravi et al. [21] proposed a new mechanism called learning-driven detection mitigation, which can detect and mitigate DDoS [22, 23]. However, when the amount of data increases, ML cannot process large amounts of data. ML also faces the shortcomings, such as inaccurate identification of attack types.

Deep Learning Methods. In recent years, deep learning (DL) technology has been continuously applied to intrusion detection systems. Facing the complexity and diversity of network traffic, DL can work well on intrusion detection tasks, identify various types of network attacks promptly. Gao et al. [24] improve IDS performance by combining feed-forward neural networks and LSTM. Gamage et al. [25] conduct experimental research on intrusion detection using deep learning methods, such as autoencoder, deep belief network, and feed-forward neural network. In addition, Wu et al. [26] conduct unsupervised anomaly detection with training data only containing normal data. They propose a fault-attention generative probability adversarial autoencoder, which can find low-dimensional manifolds in high-dimensional space. Due to the advantages of the autoencoder, the information loss in the feature extraction process is reduced. Afterward, the abnormal data can be identified by the reconstruction errors and probability distribution of low-dimensional features.

In the above-mentioned network traffic intrusion detection methods using deep learning, however, for supervised scenarios, because a large number of labels need to be manually labeled, a lot of time and resource consumption are required, and the real operation is difficult. Although semi-supervised methods use some labeled data, there is still a large amount of unlabeled traffic data, which are still not possible to fully utilize these data for detection of network attacks. Therefore, to be more in line with the real 6G IoT network scenarios, the unsupervised methods for intrusion detection is worthiest attention. Because they do not require any traffic labels, the cost is greatly reduced compared to supervised and semi-supervised methods. Meanwhile, the inherent features in IoT traffic samples can be used to distinguish different network traffic types.

In this article, we present an unsupervised deep learning method based on reconstruction, which can reduce the cost of manual labeling and more closely to the real network traffic scenarios. Given the incompleteness of the compressed data of the original autoencoder and the inaccurate identification of the discriminator in the GAN, a multiresolution encoder and a discriminator with feature match loss are proposed, respectively. The multiresolution encoder can obtain multiple latent representations with different scales. Combining all the latent representations as a whole can get better reconstructed data. In addition, feature match loss can constrain the discriminator in the GAN to discriminate between real and fake data. At the same time, we combine the improved autoencoder with the GAN, and replace the entire autoencoder with the generator in the GAN network. The output is used as the fake reconstruction data. In the next stage, the discriminator makes inferences between fake data and real input data to determine whether it is real data or reconstructed fake data by the autoencoder. Through such repeated training, the generator and discriminator will eventually reach some balance. Finally, we can perform the 6G network traffic detection tasks.

3 Methodology

In this section, we describe the component details of our intrusion detection framework, which contains multiresolution autoencoder and feature matching discriminator. First, we introduce the multiresolution encoder to be the generator to perform the feature extraction and fusion; Next, we add a discriminator after the generator for adversarial training. The discriminator helps identify anomalies by comparing the features of the input network traffic data and the reconstructed data; Finally, we design the loss function and the evaluation method of anomaly score. The framework of our model is shown in Fig. 3.

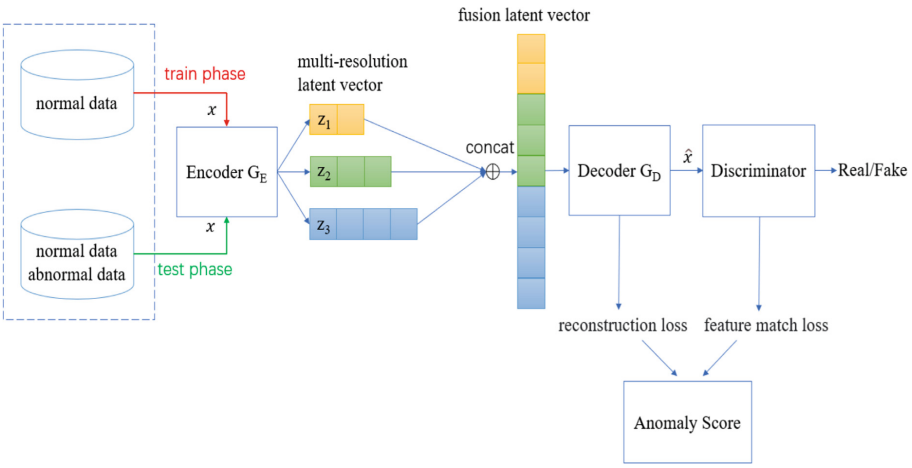

Fig. 3. The model framework.

3.1 Network Framework

Multiresolution Encoder. Our multiresolution autoencoder and discriminator network adopts an encoder, G_E, with three different scales channel and a decoder, G_D. The encoder network converts an input, x, into three latent representation vectors z_i, $i = 1, 2, 3$. After that, we concatenate these three representation vectors along the channel direction to obtain a fusion vector, z. Being asymmetric to G_E, the decoder G_D reconstructs the latent representation vector, z, back to a vector with the same size as the input, denoted as $\hat{x}$.

$$z_i = G_E(x), i = 1, 2, 3 \tag{1}$$

$$z = z_1 \oplus z_2 \oplus z_3 \tag{2}$$

$$\hat{x} = G_D(z) \tag{3}$$

where $\oplus$ is the concatenation operation.

Adversarial Training. In the face of the complexity of high-dimensional data, only the reconstruction loss calculated by the Euclidean distance is used as the anomaly score is easily affected by the training data, resulting in overfitting and cumulative error. Therefore, in order to better training and learning data features [27], we connect a discriminator after the autoencoder. Specifically, the multiresolution encoder and decoder network are treated as a generator, G, and a discriminator, D is appended after the decoder. As shown in Fig. 3, the discriminator, D, predicts whether the label is true or fake for a given input data.

During the training phase, D tries to distinguish real data, x, from the fake one, $\hat{x}$, generated by G. On the other hand, the objective of G is to fool the discriminator by minimizing the distance between the normal traffic data and the reconstructed data.

During the testing phase, D is utilized as a feature extractor to obtain features of the input, x, and the reconstructed $\hat{x}$. In addition to using the reconstruction error as the anomaly score, the feature match error is added to the abnormal score to enlarge the gap between the normal traffic data and abnormal intrusion traffic data, which can make the model more robust when it obtains poor reconstruction.

3.2 Loss Function

Reconstruction Loss. The goal of reconstruction loss is to make the reconstructed traffic data by generator similar to the real traffic data. Further, we compute the distance between the input, x, and the reconstructed output, $\hat{x}$, which ensures the generator learns enough latent feature representations from normal traffic data. The reconstruction loss is shown as follows:

$$\mathcal{L}_{rec} = \mathbb{E}\|x - \hat{x}\|_2 \tag{4}$$

Adversarial Loss. As shown in Eq. (5), we utilize the traditional adversarial loss function proposed in [9] to train the generator and discriminator. The generator tries to generate the real reconstructed data, $\hat{x}$, as much as possible, and the discriminator strives to distinguish the real input data and the fake reconstruction data. The adversarial loss, L_{adv}, is shown as follows,

$$\mathcal{L}_{adv} = \mathbb{E}_{x\sim p(data)}\big[logD(x)\big] + \mathbb{E}_{\hat{x}\sim g(data)}[log(1 - D\big(G(\hat{x})\big)] \tag{5}$$

Feature Match Loss. The purpose of feature match loss is to force the distribution of feature representations to be consistent with the reconstructed $\hat{x}$ and the input x. By feeding x and $\hat{x}$ to the discriminator, D, we can obtain the high-dimensional features at each layer of D. The corresponding features of x and $\hat{x}$ are denoted as $H_D(x)$ and $H_D(\hat{x})$, respectively. Therefore, the feature match loss is defined as follows:

$$\mathcal{L}_{fm} = \mathbb{E}\|\mathrm{H}_D(x) - \mathrm{H}_D(\hat{x})\|_2 \tag{6}$$

The final loss function can be expressed as follows,

$$\mathcal{L} = \lambda_{rec}\mathcal{L}_{rec} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{fm}\mathcal{L}_{fm} \tag{7}$$

3.3 Intrusion Detection

During the testing phase, anomaly traffic data are detected by fusing the reconstruction error and the feature match error. For the input traffic data, x, we define the anomaly score as follows:

$$S = \eta R(x) + (1 - \eta)M(x) \tag{8}$$

$R(x)$ is the reconstruction score calculated by the distance between the input data and the reconstructed data based on Eq. (4). $M(x)$ is the feature match score calculated by the distance between the feature representations of the input and the reconstructed data based on Eq. (6). η is a hyperparameter used to adjust the ratio of $R(x)$ and $M(x)$. In addition, we normalized $R(x)$ and $M(x)$ before calculating the whole anomaly score to perform the network traffic detection tasks.

Algorithm 1 shows the process in detail.

Algorithm 1 Network Traffic Detection

Input: traffic data, $x = \{(x_i, y_i)\}_{i=1}^{N}$; Iterations, L; threshold, τ; Initialize l=0, $S = 0$

1. while l<L do
2. Sample $\{(x_1, y_1), ..., (x_m, y_m)\}$ from normal and abnormal data
3. Generate $\{(\hat{x}_1, y_1), ..., (\hat{x}_m, y_m)\}$ by multiresolution Encoder, G_E, and Decoder, G_D
4. The reconstruction loss is calculated using Equation (4)
5. Distinguish the input data and reconstruct data by the discriminator, D
6. The adversarial loss and feature match loss are calculated using Equations (5) and (6)
7. Calculate the anomaly score, S, using Equation (8) by reconstruction loss and feature match loss
8. Classify each traffic data as an anomaly data if $S > \tau$, otherwise normal data
9. End

4 Experiment

4.1 Dataset Description

We select the CIC-IDS2018 [28] dataset to evaluate the performance of the network traffic detection. CIC-IDS2018 is a well-known real-world heterogeneous intrusion detection dataset containing various attack types, missing values, and irrelevant features. It includes millions of IoT traffic samples collected from the network traffic for 10 days. The dataset contains 79-dimensional data features and one-dimensional label features. The data

labeled Benign represent normal data, and the rest ones represent abnormal data. Details of the dataset and corresponding data distributions used in our experiments are shown in Table 1.

Table 1. Details of CIC-IDS2018 Datasets and Data Distributions

Dataset	Number of data	Number of attack types	Normal type	Data attributes	Attack ratio
CIC-IDS2018 [28]	16233002	DDOS, attack-HOIC, DDoS attacks-LOIC-HTTP, DoS attacks-Hulk, Bot, FTP-BruteForce, SSH-Bruteforce, Infilteration, DoS attacks-SlowHTTPTest, DoS attacks-GoldenEye, DoS attacks-Slowloris, DDOS attack-LOIC-UDP, Brute Force-Web, Brute Force-XSS, SQL Injection	Benign	79	17%

4.2 Performance Metrics

We use five metrics to evaluate the model, including AUC, Accuracy, Precision, Recall, and F1-score. AUC is the area under the receiver operating characteristic curve. The other metrics are described and presented as follows.

Accuracy indicates the ratio of correctly predicting traffic samples in the overall sample. The formula is shown as follows,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

Precision indicates the ratio of correctly predicting normal traffic samples in the overall sample. The formula is shown as follows,

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

Recall indicates the ratio of correctly predicting normal traffic samples among all normal samples. The formula is shown as follows,

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

F1 is calculated by considering both precision and recall, and is the weighted average of both. We use the threshold that maximizes the F1-score as the optimal threshold to calculate the corresponding indicators. The formula is shown as follows,

$$F1 - score = \frac{2TP}{2TP + FN + FP} \tag{12}$$

where false negative (FN) indicates the number of normal traffic samples that were falsely judged to be abnormal traffic samples. The false positive (FP) represents the number of abnormal traffic samples that were falsely identified as normal traffic samples. The true positive (TP) represents the number of normal traffic samples correctly identified as positive. The true negative (TN) is the number of abnormal traffic samples correctly identified as negative.

4.3 Implementation Detail

We treat the experiment as a one-classification problem, with 14 attack types of samples as anomalies and the samples with Benign label as normal values. Only normal data are used for unsupervised training in the training phase, and normal and abnormal data are used for anomaly detection in the testing phase.

Specifically, we eliminated three useless features (i.e., Dst Port, Protocol, Timestamp). The whole data was normalized by column.The Benign label was set to 0 and the others arer set to 1. We utilize 80% of the Benign data for training data, and the rest 20% and all abnormal data for testing data.

The number of multiresolution encoders is set as 3. We use the Adam optimizer to train generators and discriminators, and the learning rate is set to 0.001. In addition, the weight decay is 5e-7. The model is trained by 200 epochs and the batch size is 512. We conduct our experiments on an NVIDIA RTX 3090ti GPU on Pytorch 1.9.1.

4.4 Result and Discussion

Table 2 shows the detection results on the CICIDS-2018 dataset. We can conclude that the AUC of our model can reach 97.6%, and the Accuracy, Precision, Recall, and F1 can be achieved 97.1%, 94%, 99.7%, and 96.8%, respectively. Compared with auto-encoder and GAN, our model extracts data features from multiple resolutions and obtains more comprehensive feature information, which is beneficial to reconstruct information closer to the real data in the decoding stage. Additionally, our feature match loss distinguishes between normal and abnormal data more clearly than merely using the reconstruction error, to further improve the model performance.

Table 2. Intrusion detection in terms of several metrics on CICIDS-2018 Dataset

Method/Metric	AUC	Accuracy	Precision	Recall	F1-score
AE	0.756	0.734	0.625	0.951	0.754
GAN	0.906	0.924	0.845	0.995	0.919
Our model	**0.976**	**0.971**	**0.94**	**0.997**	**0.968**

5 Conclusion

This paper proposes an unsupervised deep learning model to detect network attacks from network traffic. We extract features using a multiresolution encoder from traffic data, and then the decoder reconstructs the fused features. Furthermore, we compute anomaly scores based on three errors. The experimental results highlight that our model outperforms the original auto-encoder and generative adversarial network by more than 20% and 7% on AUC, respectively. This can be applied to future 6G network traffic intrusion detection systems. In the future, we will improve the components of the model to accommodate more complex data characteristics, and practice the model on more 6G application scenarios and improve its usability and robustness.

Acknowledgment. This work was supported by Heilongjiang Province Natural Science Foundation under Grant LH2022F034.

References

1. Saad, W., Bennis, M., Chen, M.: A vision of 6G wireless systems: applications, trends, technologies, and open research problems. IEEE Network **34**(3), 134–142 (2019)
2. De Alwis, C., Kalla, A., Pham, Q.V., et al.: Survey on 6G frontiers: trends, applications, requirements, technologies and future research. IEEE Open J. Commun. Soc. **2**, 836–886 (2021)
3. Sun, Y., Liu, J., Wang, J., et al.: When machine learning meets privacy in 6G: a survey. IEEE Commun. Surv. Tutorials **22**(4), 2694–2724 (2020)
4. Mitrokotsa, A., Komninos, N., Douligeris, C.: Intrusion detection with neural networks and watermarking techniques for MANET. In: IEEE International Conference on Pervasive Services. IEEE, pp. 118–127 (2007)
5. Shafiq, M., Tian, Z., Bashir, A.K., et al.: CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine-learning techniques. IEEE Internet Things J. **8**(5), 3242–3254 (2020)
6. Li, L., Yan, J., Wang, H., et al.: Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. IEEE Trans. Neural Networks Learn. Syst. **32**(3), 1177–1191 (2020)
7. Wang, X., Han, Y., Leung, V.C.M., et al.: Convergence of edge computing and deep learning: a comprehensive survey. IEEE Commun. Surveys Tutorials **22**(2), 869–904 (2020)
8. Sakurada, M., Yairi, T.: Anomaly detection using autoencoders with nonlinear dimensionality reduction. In: Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis, pp. 4–11 (2014)

9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: Generative adversarial nets. Advances in Neural Information Processing Systems 27, Curran Associates, Inc. (2014)
10. Alsharif, M.H., Kelechi, A.H., Albreem, M.A., et al.: Sixth generation (6G) wireless networks: vision, research activities, challenges and potential solutions. Symmetry **12**(4), 676 (2020)
11. Ferrag, M.A., Maglaras, L., Derhab, A.: Authentication and authorization for mobile IoT devices using biofeatures: recent advances and future trends. Secur. Commun. Networks **2019**, 1–20 (2019)
12. Majid, M., et al.: Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: a systematic literature review. Sensors **22**(6), 2087 (2022)
13. Tang, F., Kawamoto, Y., Kato, N., et al.: Future intelligent and secure vehicular network toward 6G: machine-learning approaches. Proc. IEEE **108**(2), 292–307 (2019)
14. Zhang, Z., Cao, Y., Cui, Z., et al.: A many-objective optimization based intelligent intrusion detection algorithm for enhancing security of vehicular networks in 6G. IEEE Trans. Veh. Technol. **70**(6), 5234–5243 (2021)
15. Ahmad, I., Shahabuddin, S., Kumar, T., et al.: Security for 5G and beyond. IEEE Commun. Surveys Tutorials **21**(4), 3682–3722 (2019)
16. S.A., et al.: 6G white paper on machine learning in wireless communication networks. https://arxiv.org/pdf/2004.13875.pdf. Accessed 10 Aug 2021
17. Anthi, E., Williams, L., Słowińska, M., et al.: A supervised intrusion detection system for smart home IoT devices. IEEE Internet Things J. **6**(5), 9042–9053 (2019)
18. Pu, C.: Sybil attack in RPL-based internet of things: analysis and defenses. IEEE Internet Things J. **7**(6), 4937–4949 (2020)
19. Yang, H., Liang, S., Ni, J., et al.: Secure and efficient k NN classification for industrial Internet of Things. IEEE Internet Things J. **7**(11), 10945–10954 (2020)
20. Ravi, N., Shalinie, S.M.: Semisupervised-learning-based security to detect and mitigate intrusions in IoT network. IEEE Internet Things J. **7**(11), 11041–11052 (2020)
21. Ravi, N., Shalinie, S.M.: Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture. IEEE Internet Things J. **7**(4), 3559–3570 (2020)
22. Wang, J., Jiang, C., Zhang, H., et al.: Thirty years of machine learning: the road to Pareto-optimal wireless networks. IEEE Commun. Surveys Tutorials **22**(3), 1472–1514 (2020)
23. Hussain, F., Hussain, R., Hassan, S.A., et al.: Machine learning in IoT security: current solutions and future challenges. IEEE Commun. Surveys Tutorials **22**(3), 1686–1721 (2020)
24. Gao, J., Gan, L., Buschendorf, F., et al.: Omni SCADA intrusion detection using deep learning algorithms. IEEE Internet Things J. **8**(2), 951–961 (2020)
25. Gamage, S., Samarabandu, J.: Deep learning methods in network intrusion detection: a survey and an objective comparison. J. Netw. Comput. Appl. **169**, 102767 (2020)
26. Wu, J., Zhao, Z., Sun, C., et al.: Fault-attention generative probabilistic adversarial autoencoder for machine anomaly detection. IEEE Trans. Industr. Inf. **16**(12), 7479–7488 (2020)
27. Grill, J.B., Strub, F., Altché, F., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Adv. Neural Inf. Process. Syst. **33**, 21271–21284 (2020)
28. CSE-CIC-IDS2018 Dataset, CIC, Fredericton, NB, Canada (2018). https://www.unb.ca/cic/datasets/ids-2018.html. Accessed 4 Oct 2020