Intrusion Detection Method for Networked Vehicles Based on Data-Enhanced DBN

Yali Duan[1], Jianming Cui[1], Yungang Jia[2], and Ming Liu[2(✉)]

[1] School of Information Engineering, Chang'an University, ShaanXi 710064, China
cjianming@chd.edu.cn

[2] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing 100029, China
liuming@cert.org.cn

Abstract. At present, cyber attacks on vehicle network have are proliferating, one of the most significant difficulties in the current detection methods is that the malicious flows are small and discrete in the whole link. In view of the above issues, this paper proposed a detection model based on the integration of Generative Adversarial Networks (GANs) and Deep Belief Networks (DBN). In this model, GANs is first used to enhance the few malicious flow samples, and then an improved DBN is used to evaluate the effect of data generation, so as to improve the uneven distribution of samples in the data set. In the testing section, open data set CIC-IDS2017 was selected for data enhancement and evaluated the performance of the proposed model. The experimental results show that the proposed model has significantly improved the detection performance of few cyber attacks samples compared with traditional detection algorithms. In addition, compared with the method of merge-generate data set approach, the accuracy rate, recall rate, F1 value and other evaluation indexes of the proposed model for the few samples detection have been greatly improved. Therefore, it can be considered that the proposed model is effective than current methods in dealing with the uneven distribution of data sets in traditional cyber attack detection.

Keywords: Generative Adversarial Networks · Networked vehicles · Intrusion detection · Sample distribution

1 Introduction

Traditional vehicular networks (VANETs) [1] are gradually evolving into intelligent vehicular networks. While achieving network communication, vehicles are vulnerable to malicious network flow and may lead to privacy leakage due to the lack of security mechanisms such as firewalls and gateways in some of the devices [1–3]. Improving the active defense capability and security of vehicular

This work is financially supported by the National Natural Science Foundation of China under Grant 62106060.

Z. Tari et al. (Eds.): ICA3PP 2023, LNCS 14488, pp. 40–52, 2024.
https://doi.org/10.1007/978-981-97-0801-7_3

networks is an important and popular research direction [3,4]. Traditional intrusion detection techniques can detect ongoing and existing malicious attacks in a timely manner. However, in uneven distribution massive network flow, malicious cyber attacks often hide in a large amount of normal data, making traditional intrusion detection methods difficult to deal with evolving malicious attacks and network threats [1,5].

Currently, the main methods for handling uneven distribution data [6] include resampling methods [7], cost-sensitive algorithms, ensemble methods, feature representation and classification decoupling, etc. These methods attempt to rebalance the class weight norms in the machine learning model by increasing the number of samples of minority attacks. However, these traditional algorithms still have some problems. Generative adversarial networks (GANs) [8,9] can learn the distribution of given data and generate new sample data. Currently, GANs are mostly used in natural images [10,11], and have achieved significant results. Inspired by its success in these fields, scholars are gradually starting to use GANs to generate adversarial network flow for intrusion detection.

It can be argued that the current intrusion detection approach for vehicular networks has the following two drawbacks: (1) The network flow data is uneven distribution and the number of samples in the minority class is too small. The commonly used data augmentation algorithm, SMOTE algorithm [12,13] does not consider noise data and boundary issues, which may cause overlap between different categories, leading to decreased accuracy and overfitting problems. (2) Some current intrusion detection models for vehicular networks perform a lower detection rate and weak classification ability. On this basis, a new intrusion detection model is proposed, which combines GANs with DBN [14], and uses a GAN-based data augmentation method [15] to generate adversarial attack samples for the minority class, in order to expand the dataset CIC-IDS2017, an improved DBN classifier is designed to evaluate the effectiveness of this method [14,16]. How to achieve better classification effect, higher classification accuracy and higher precision, which is the main innovation and challenge of this paper.

2 Intrusion Detection Model Based on Data Augmentation

2.1 Data Processing and Augmentation

Dataset Analysis and Preprocessing. In this paper, since the vehicle data set may lead to user privacy leakage, the general vehicle data set is not open to the public, so the open source network intrusion detection data set is adopted. In the existing open source datasets, CIC-DDOS2019 dataset proposes an attack classification method for DDOS, and the CSE-CIC-IDS2018 is mostly used for anomaly detection. However, this paper studies data enhancement and intrusion detection for a few categories in the vehicle network. So, the dataset used is the CIC-IDS2017 dataset [17] provided by the Canadian Institute for Cybersecurity. It contains 5 d of normal and attack flow data collected by the institute, with

each record having 78 network features. The dataset includes the latest cyber attacks and meets all standards of real-world attacks, and can fully simulate the attack to vehicle network [18].

After merging the 8 csv files of the CIC-IDS2017 dataset, missing instances were removed from the dataset along with NaN values to avoid redundancy and exploding gradients when training the model. The dataset was then transformed into numerical values and normalized, with the most frequently occurring class being labeled as 'Normal' and all other classes labeled as 'Attack' to meet the conditions of inputting the dataset into the GAN network.

GANs-Based Data Augmentation Methods. To address the problem of highly uneven distribution data in vehicle network datasets, a data augmentation method based on Generative Adversarial Networks (GANs) is used to generate adversarial attack samples for the minority class, in order to expand the dataset CIC-IDS2017 [17] used in the study.

Algorithm 1. Data Augmentation Algorithm based on GANs

Input: $s = (r, y)$ r is the eigenvector y is the category label
Output: $S_G = [G(z, y'), y']$

while $0 <= p_{\text{data}} <= \frac{1}{2}$ **do** /* train GAN*/
  for k steps **do** /* train Discriminator*/
    Sample random noise $\left\{z^{(1)}, z^{(2)}, \ldots z^{(m)}\right\}$ from $p_z(z)$
    Sample real data $\left\{x^{(1)}, x^{(2)}, \ldots x^{(m)}\right\}$ from $p_{\text{data}}(x)$
    $\eta_{\theta_D} \leftarrow \nabla\theta_D \frac{1}{m} \sum_1^m \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right]$
    /* update the weight parameters and gradients */
    $\theta_D \leftarrow \theta_D + \alpha_D \cdot \text{Adam}(\theta_D, \eta_{\theta_D})$
  end for
  Sample random noise $\left\{z^{(1)}, z^{(2)}, \ldots z^{(m)}\right\}$ from $p_z(z)$/* train Generator*/
  $\eta_{\theta_D} \leftarrow \nabla\theta_G \frac{1}{m} \sum_1^m \left[\log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right]$ /* update the weight parameters and gradients */
  $\theta_D \leftarrow \theta_G - \alpha_G \cdot \text{Adam}(\theta_G, \eta_{\theta_G})$
end
return data /* Generate data */

Algorithm 1 is the data augmentation training process of GANs. Where Θ_G, η_{θ_G}, θ_D, η_{θ_D} are the generated weight parameters, gradients, and discriminator weight parameters and gradients, respectively. Its main steps are:

- (1) The category label y' of the preprocessed minority data is input to the generator G with the random noise vector z for training and the data sample S_G generated;
- (2) Fix the generator G, train the discriminator D, and gradually update the weight parameter θ_D of the discriminator;

– (3) Fixed discriminator D, train generator G, and gradually update the weight parameter θ_G of the generator;
– (4) Loop (1)–(3) until $p_g = p_{data} = 1/2$, the discriminator cannot distinguish between the two distributions, so that the generated sample keeps approaching the real data sample.

Figure 1 shows the occurrence of classes in the original dataset and the minority class after data augmentation, it can be observed that data augmentation is effective in increasing the number of samples in classes with fewer than 5000 samples in the dataset, particularly for extremely rare classes such as Heartbleed and Infiltration, which are increased from 11 and 36 samples, respectively, to 5632 and 8704 samples.

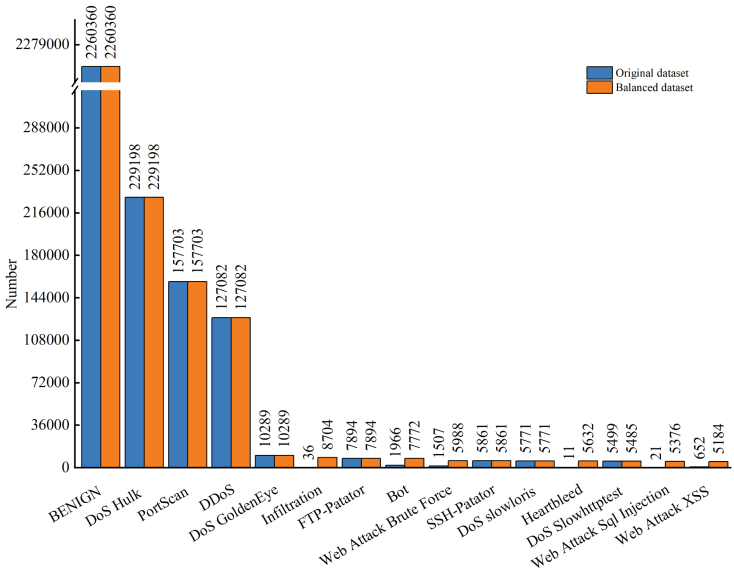

Fig. 1. Comparison of Original Data and Quantity after Data Augmentation

2.2 Improved DBN Model

Structurally, Deep Belief Networks [19] is a probabilistic generative model composed of multiple layers of unsupervised Restricted Boltzmann Machines (RBM) and a supervised Back-Propagation (BP) network, DBN is composed of multiple stacked RBMs, each consisting of a hidden layer and a visible layer.

The training of DBN [20] consists of the layer-wise pre-training stage and the back-propagation fine-tuning stage. In the pre-training stage, it uses the Contrastive Divergence (CD) algorithm proposed by Hinton to quickly train RBM to obtain an approximate representation of the input vector v. In the back-propagation fine-tuning stage, the BP algorithm and stochastic gradient

descent are used to optimize the connection weights in the DBN to obtain the optimal model parameters. For the pre-training stage, the Mean Square Error (MSE) and Pseudo-Likelihood (PL) loss functions were used to evaluate the accuracy of RBM training. The calculation method for MSE is Eq. (1):

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(x_i - \bar{x})^2 \tag{1}$$

where m is the number of samples, $x_i (i = 1, 2, 3 \ldots m)$ is the sample, $\bar{x}$ is the average of m samples. For the reverse tuning stage, in order to determine the learning rate of the model, the loss and accuracy of the training set and the validation set are used to evaluate its performance.

Algorithm 2 is the specific algorithm trained by DBN. Where W,W^k are the weight matrix with the training stage and the fine-tuning stage respectively, $a_i (i = 1, 2, 3 \ldots M)$, $b_j (j = 1, 2, 3 \ldots N)$ are the biases of the visible layer and the hidden layer respectively, ϵ,ϵ_{ft} are the learning rate of the pre-training stage and the fine-tuning stage, $V = v_1, v_2 \ldots, v_m$ is the training sample of RBM, l is the number of layers of RBM, a^k, b^k $(k = 1, 2, 3 \ldots l)$ are the bias of the visible layer and the hidden layer at the kth layer, respectively.

3 Experimental Design and Result Analysis

The experiments were conducted in a Win10 environment, with a 64-bit Intel(R) Xeon(R) Silver 4100 CPU and 32 GB RAM. The implementation was done using Python 3.8 language and the Pytorch 1.9 framework.

3.1 Dataset Labels

The proposed intrusion detection model was evaluated using the CIC-IDS2017 dataset. After data enhancement, similar attack classes with similar characteristics and behaviors were merged into a new class, and the dataset was re-labeled. The final standard dataset was divided into 9 classes. Table 1 shows the number of labels in the standardized dataset after re-labeling.

3.2 Experiments and Analysis

To evaluate the detection performance of the proposed intrusion detection model, the following experiments were designed.

Experiments on Training GANs-DBN Model. GANs were used to generate samples for the 8 minority classes in the dataset. The GANs training parameters are shown in Table 2. After training, the dataset was re-labeled to generate the standard dataset, which was then divided into training set, testing set, and validation set in a 60%, 20%, 20% ratio. Finally, the data was input into

Algorithm 2. DBN Training Algorithm

Initialize $W = W^k = a_i = b_j = 0$

the first phase: Train RBM
for s steps **do**/*Set the number of iterations s^*/
  for v_i **do** $i = 1, 2 \ldots m$
    for k **do** /*Gibbs sampling*/
      for i **do** $i = 1, 2 \ldots M$
        $h_j^{(k)} \leftarrow p\left(h_j \mid v^{(k)}\right)$
      end for
      for j **do** $j = 1, 2 \ldots N$
        $v_i^{(k+1)} \leftarrow p\left(v_i \mid h^{(k)}\right)$
      end for
      for i, j **do**
        $i = 1, 2 \ldots M j = 1, 2 \ldots N$ /* Update weights and biases */
        ${W_{ij}}^{(k)} \leftarrow {W_{ij}}^{(k+1)} + \varepsilon\left(p\left(h_j \mid v^{(0)}\right)\right)$
        $a_i^{(k+1)} \leftarrow a_i^{(s)} + \varepsilon\left(v_i^{(0)} - v_i^{(k)}\right)$
        $b_i^{(k+1)} \leftarrow b_i^{(s)} + \varepsilon\left(p\left(h_j \mid v^{(k)}\right)\right) - p\left(h_j \mid v^{(k)}\right)$
      end for
    end for
  end for
end for
the second phase: Fine Tune DBN
% Forward propagation
for l **do**
  Initialization $W^k = a^k = b^k = 0, \quad \varepsilon = \varepsilon_0$
  Train RBM /* Traverse each layer RBM*/
end for
for i **do**
  $i = 1, 2 \ldots m$
  Compute $o_i\left(x_i\right)$
end for
% Backpropagation
for k **do** $k = l, l-1 \ldots 1$
  if $k = l$
    $\delta_k \leftarrow o_k\left(1 - o_k\right)\left(t_k - o_k\right)$
  else
    $\delta_h \leftarrow o_h\left(1 - o_k\right) \sum_{k \in \text{ outputs}} \varepsilon_{kh} \delta_k$
    $\theta_{ji} \leftarrow \theta_{ji} + \Delta\theta_{ji}, \quad \theta_{ji} = \varepsilon_{ft} \delta_j x_j$
  end if
end

Table 1. Number of Labels After Relabeling

New labels	Origin labels	Numbers
Benign	BENIGN	2260360
Brute Force	SSH-Patator	13755
	FTP-Patator	
DoS	DoS-Hulk	250743
	DoS-GoldenEye	
	DoS-slowloris	
	DoS-Slowhttptest	
Heartbleed	Heartbleed	5632
Infiltration	Infiltration	8704
Web Attack	Web Attack-Brute Force	16548
	Web Attack-XSS	
	Web Attack-Sql Injection	
DDoS	DDoS	127082
PortScan	PortScan	157703
Bot	Botnet ARES	7772

Table 2. GANs Training Parameters

	Parameter
Activation function	Leaky ReLU
Learning rate	0.0002
Optimister	Adam
Loss function	Cross-entropy
Batch_size	5

Table 3. DBN Network Parameters

Parameter	Pre-training phase	Fine-tuning phase
Epochs	10	5
Learning rate	0.015	0.005
Batch size	64	128
Optimister	SGD	Adam
Gibbs steps	1	

the DBN classifier for model evaluation. The parameter settings for the DBN classifier are shown in Table 3.

During the training of the DBN classifier, in the pre-training stage, the learning rate of the RBM were determined by changing the learning rate within an approximate range of [0.001, 0.1]. As shown in Fig. 3, when the learning rate lr = 0.015, MSE = 0.538, PL = -0.818. Compared with other learning rates, the training accuracy of RBM is optimal at this learning rate. Therefore, this method

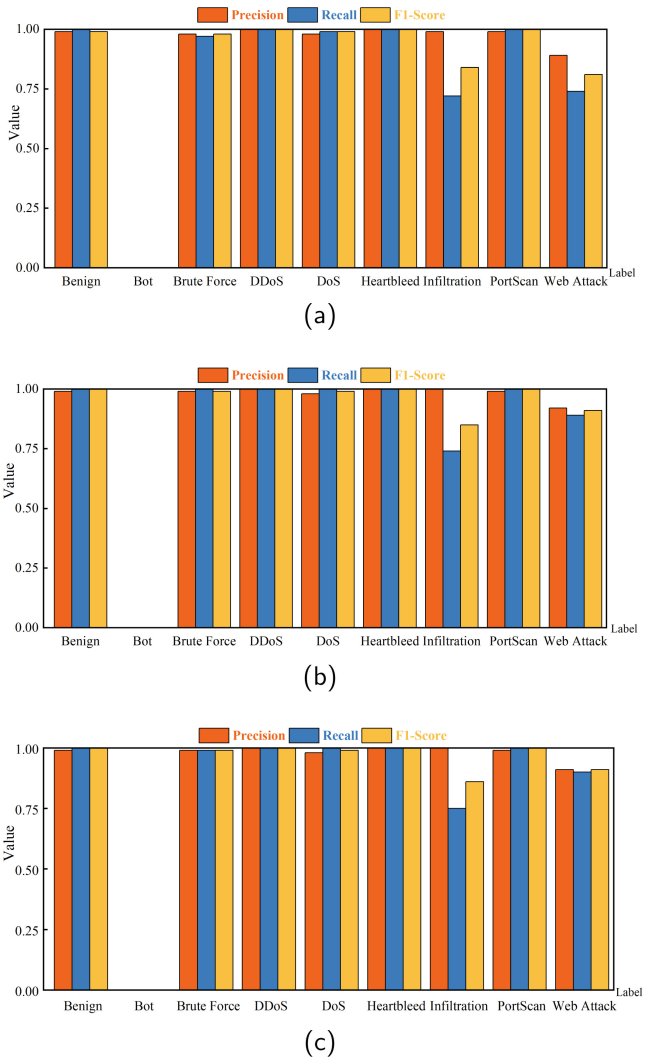

Fig. 2. Dataset Classification Results:(a) Training Set Classification Results. (b)Testing Set Classification Results. (c) Verification Set Classification Results.

selected the learning rate lr = 0.015 as the learning rate for RBM training. In the back-propagation fine-tuning stage, the performance of the model was evaluated using the loss and accuracy of the training set and the validation set to determine the optimal learning rate for this stage. From Fig. 4, it can be clearly seen that when the learning rate is lr = 0.005, the loss reaches its minimum value with train-loss = 0.453 and val-loss = 0.419, and the accuracy of the training set and validation set reaches its maximum value with train-acc = 0.993 and val-acc = 0.990. However, when the learning rate is too high, such as lr = 0.1, the model's

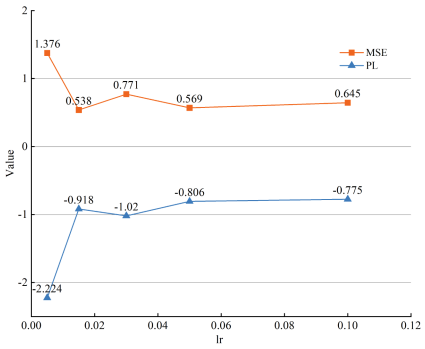

Fig. 3. Comparison of RBM Training Performance at different learning rates.

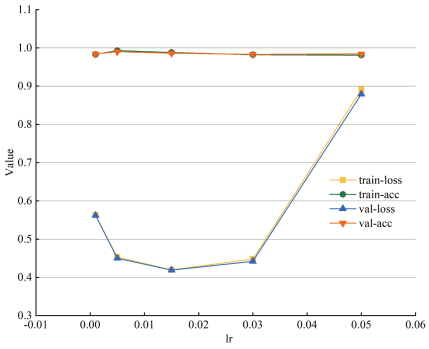

Fig. 4. Comparison of Accuracy and Loss at different learning rates

Table 4. Testing Set Confusion Matrix

True Label	Predicted Label									Recall
	Benign	Bot	Brute Force	DDoS	DoS	Heartbleed	Infiltration	PortScan	Web Attack	
Benign	451120	0	30	33	988	3	4	188	203	0.994
Bot	0	0	0	0	0	0	0	0	0	0
Brute Force	12	0	2733	0	1	0	0	0	0	0.989
DDoS	51	0	0	25138	3	0	0	0	0	0.998
DoS	167	0	0	0	49741	0	0	0	46	0.978
Heartbleed	0	0	0	0	0	1100	0	0	0	0.997
Infiltration	455	0	0	0	0	0	1270	0	0	0.997
PortScan	24	0	0	6	17	0	0	31440	2	0.994
Web Attack	276	0	0	0	92	0	0	0	2976	0.922
Precision	0.997	0	0.995	0.998	0.996	1.000	0.736	0.998	0.890	

loss reaches 0.98 and the model fails to converge. Therefore, based on the above results, this method selected the learning rate lr=0.005 as the training learning rate for the back-propagation fine-tuning stage of the model.

The classification results of the proposed model for the training set, test set, and validation set are shown in Fig. 2, and the confusion matrix for the predicted classes in the test set is presented in Table 4. It show that the proposed model can correctly classify most of the network flow, with high precision, recall, and F1 score. The precision, recall, and F1 score for the minority classes such as Brute Force, DDoS and PortScan are close to 1. Additionally, the precision, recall, and F1 score for extremely rare classes like Heartbleed and Infiltration are also above 70%, with a recall rate of 99.7% and a precision rate of 100% for Heartbleed. Therefore, the proposed model has strong detection performance for attacks on minority classes in the vehicular networks while maintaining high performance in detecting other attacks.

Performance Comparison Experiments of Different Data Augmentation Methods. To verify the effectiveness of the proposed data augmentation method, this study compared different data augmentation methods, including the SMOTE algorithm, class weight strategy, combination of SMOTE algorithm and class weight strategy, and GANs and combines it with DBN classifier for model evaluation. The parameters of the DBN network were kept consistent for each method, and the specific parameter settings are shown in Table 3.

Table 5. Comparison of GANs with Other Data Augmentation Methods

Model	Accuracy	F1-score	AUC
Class Weights+DBN	96	84	97
SMOTE+DBN	98	82	96
SMOTE+Class Weights+DBN	95	80	91
GANs+DBN	99	86	99

Table 5 shows that compared to the other three commonly used data augmentation methods, the proposed model improves accuracy, F1 score, and AUC by at least 1%, 2%, and 2%, respectively. Figure 5 compares the offline AUCs for different classes using various data augmentation methods. It can be concluded that the proposed intrusion detection method based on GANs-DBN outperforms other classification algorithms in overall performance, although it may not perform as well as some other methods for certain classes. Overall, this method greatly improves the accuracy of intrusion detection for each class.

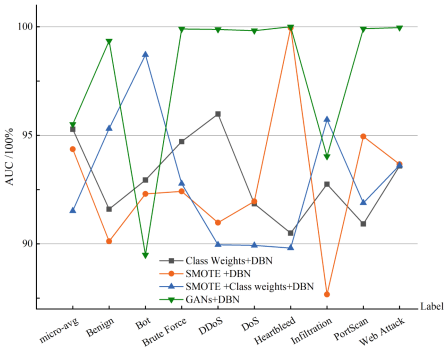

Fig. 5. Comparison of AUC for Different Data Augmentation Methods

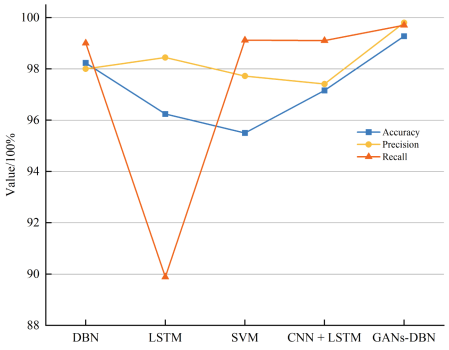

Fig. 6. Performance Comparison of Different Models

Performance Comparison Experiment of Different Models. To verify the intrusion detection performance of the proposed model, the performance of GANs-DBN was compared with several existing intrusion detection models using the CIC-IDS2017 dataset. Othmane Belarbi [21] To verify the intrusion detection performance of the proposed model, the performance of GANs-DBN was compared with several existing intrusion detection models using the CIC-IDS2017 dataset. Monika Roopak et al. [22] proposed deep learning models including LSTM, CNN + LSTM, and SVM, and evaluated DDoS attack detection using the CIC-IDS2017 dataset. For the LSTM model, the final accuracy reached 86.34%; for the CNN + LSTM model, the final accuracy reached 97.16%; for the SVM model, the accuracy reached 95.5%. By comparing the performance data of the above reference papers with the GANs-DBN model used in this paper, the detection performance of various intrusion detection models was evaluated.

From Fig. 6, it can be seen that the proposed model outperformed other models in all three indicators, reaching 99.27%, 99.80%, and 99.70% respectively, which represents at least a 1.03%, 1.36%, and 0.58% improvement, respectively. Thus, the proposed model significantly improved the detection performance for multi-class intrusion detection compared to other models.

4 Conclusion

This paper presents an integrated network intrusion detection model, GANs-DBN, designed to address the issue of low detection performance for small quantities of malicious flow in vehicle networks due to the discrete distribution of network attacks. The performance of the model is evaluated using the CIC-IDS2017 dataset. Specifically, GANs are employed for data augmentation, expanding the dataset and enriching its distribution, while an improved DBN classifier is utilized to assess the model's classification capability. Experimental results demonstrate that the proposed model outperforms alternative methods in overall detection performance, effectively enhancing the detection rate for specific classes of attacks and thereby improving overall accuracy. However, it is worth noting that the current research only partially simulates the real network conditions, and future efforts should focus on identifying and defending against the complex traffic characteristics encountered in actual vehicle networks, particularly APT attacks.

References

1. Cui, J., Ma, L., Wang, R., Liu, M.: Research and optimization of GPSR routing protocol for vehicular ad-hoc network. China Commun. **19**(10), 194–206 (2022)
2. Zhang, Y., Cui, J., Liu, M.: Research on adversarial patch attack defense method for traffic sign detection. In: Lu, W., Zhang, Y., Wen, W., Yan, H., Li, C. (eds.) Cyber Security: 19th China Annual Conference, CNCERT 2022, Beijing, China, August 16–17, 2022, Revised Selected Papers, pp. 199–210. Springer, Singapore (2022). https://doi.org/10.1007/978-981-19-8285-9_15

3. Liu, M., et al.: Modeling and analysis of the decentralized interactive cyber defense approach. China Commun. **19**(10), 116–128 (2022)
4. Kang, M.J., Kang, J.W.: Intrusion detection system using deep neural network for in-vehicle network security. PLoS ONE **11**(6), e0155781 (2016)
5. Qu, F., Wu, Z., Wang, F.Y., Cho, W.: A security and privacy review of vanets. IEEE Trans. Intell. Transp. Syst. **16**(6), 2985–2996 (2015)
6. Zhang, Y., Li, X., Gao, L., Wang, L., Wen, L.: Imbalanced data fault diagnosis of rotating machinery using synthetic oversampling and feature learning. J. Manuf. Syst. **48**, 34–50 (2018)
7. He, H., Yang, B., Garcia, E., Li, S.A.: 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 (2008)
8. Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A.A.: Generative adversarial networks: an overview. IEEE Signal Process. Mag. **35**(1), 53–65 (2018)
9. Goodfellow, I., et al.: Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020)
10. Wang, Z., She, Q., Ward, T.E.: Generative adversarial networks in computer vision: a survey and taxonomy. ACM Comput. Surv. **54**(2), 1–38 (2021)
11. Yu, X., Cui, J., Liu, M.: An embedding carrier-free steganography method based on Wasserstein GAN. In: Lai, Y., Wang, T., Jiang, M., Xu, G., Liang, W., Castiglione, A. (eds.) Algorithms and Architectures for Parallel Processing (ICA3PP 2021). LNCS, vol. 13156. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-95388-1_35
12. She, X., Sekiya, Y.: A convolutional autoencoder based method with smote for cyber intrusion detection. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 2565–2573. IEEE (2021)
13. Soltanzadeh, P., Hashemzadeh, M.: Rcsmote: range-controlled synthetic minority over-sampling technique for handling the class imbalance problem. Inf. Sci. **542**, 92–111 (2021)
14. Zhang, Y., Li, P., Wang, X.: Intrusion detection for iot based on improved genetic algorithm and deep belief network. IEEE Access **7**, 31711–31722 (2019)
15. Tanaka, F.H.K.d.S., Aranha, C.: Data augmentation using gans. arXiv preprint arXiv:1904.09135 (2019)
16. Liu, J., Wu, N., Qiao, Y., Li, Z.: Short-term traffic flow forecasting using ensemble approach based on deep belief networks. IEEE Trans. Intell. Transp. Syst. **23**(1), 404–417 (2020)
17. Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. ICISSp **1**, 108–116 (2018)
18. Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Synthetic data augmentation using gan for improved liver lesion classification. In: 2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), pp. 289–293. IEEE (2018)
19. Sohn, I.: Deep belief network based intrusion detection techniques: a survey. Expert Syst. Appl. **167**, 114170 (2021)
20. Gao, N., Gao, L., Gao, Q., Wang, H.: An intrusion detection model based on deep belief networks. In: 2014 Second International Conference on Advanced Cloud and Big Data, pp. 247–252. IEEE (2014)

21. Belarbi, O., Khan, A., Carnelli, P., Spyridopoulos, T.: An intrusion detection system based on deep belief networks. In: Su, C., Sakurai, K., Liu, F. (eds.) Science of Cyber Security: 4th International Conference, SciSec 2022, Matsue, 10–12 August 2022, Revised Selected Papers, pp. 377–392. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17551-0_25
22. Roopak, M., Tian, G.Y., Chambers, J.: Deep learning models for cyber security in IoT networks. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0452–0457. IEEE (2019)