2024 10th International Conference on Control, Decision and Information Technologies
CoDIT 2024 | Valletta, Malta / July 01-04, 2024 Technically co-sponsored by IEEE & IFAC

2024 10th International Conference on Control, Decision and Information Technologies (CoDIT) | 979-8-3503-7397-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/CoDIT62066.2024.10708352

Multi-Classification Decision Fusion Based on Stacked Sparse Shrink AutoEncoder and GS-Tabnet for Network Intrusion Detection

Ziqi Wang[1], Ziyue Guan[1], Xiangxi Wu[1], Jing Bi[1], Haitao Yuan[2] and MengChu Zhou[3]

[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China
[2]School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[3]Dept. of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA

***Abstract*—With the rapid development of the Internet, various network invasive behaviors are increasing rapidly. This seriously threatens the economic development of individuals, enterprises, and society. Network intrusion detection is important in network security systems, which can be regarded as a classification problem. It aims to distinguish between the specific categories of various network behaviors and determine whether the behavior belongs to network intrusion. However, network intrusions present a diverse and fast-changing trend, making categorizing difficult. Due to feature redundancy, uneven distribution of sample numbers, and inefficient parameter optimization, traditional rule-based approaches fail to achieve satisfying classification accuracy. This work proposes a multi-classification intrusion detection model based on Stacked Sparse Shrink AutoEncoder (SSSAE), Genetic Simulated annealing-based particle swarm optimization optimized Tabnet classifier (GS-Tabnet), and Decision Fusion (DF), called for SGTD short. Among them, SSSAE extracts multiple feature sets from the input data. Then GS-Tabnet trains a classifier for each feature set. Finally, the decision fusion fuses the results from these classifiers to obtain the final classification result. SGTD is compared with eight multi-classification benchmark models, and its intrusion detection accuracy is superior to its peers.**

***Index Terms*—Feature learning, autoencoder, network intrusion detection, intelligent optimization algorithm.**

I. INTRODUCTION

Nowadays, human society has entered the information age. With the continuous development of the Internet, network demand and dependence continue to increase. In this case, network applications and behaviors increase dramatically, and the number of users grows rapidly [1]. However, different kinds of network invasions also continue to emerge, which exposes the network environment to risk. Therefore, detecting network intrusion behavior has become important to maintain network security [2]. Intrusion detection can be regarded as a classification problem that identifies categories of network behaviors through various types of data features to distinguish normal network activities from network intrusions [3]. However, as network intrusion scenarios become more complex, they show diverse and fast-changing trends, turning network intrusion detection into a large-scale multi-feature extraction and multi-classification task for massive amounts of data. The problem's complexity and the large data size make traditional approaches ineffective for solving network intrusion detection problems.

Traditional machine learning methods are widely adopted in classification problems, *e.g.*, K-Nearest Neighbor (KNN) [4], and Support Vector Machine (SVM) [5]. They are utilized to construct a single classifier for behavioral data classification to detect network intrusions. However, their simple network structures lack deep exploration of the relationship between data features and behavioral categories. They cannot perform feature screening and high-level feature extraction effectively [6]. In addition, they have poor stability and cannot adapt to multi-classification tasks. To avoid these limitations, deep learning models are applied in classification tasks. Recurrent Neural Networks (RNN) [7] and Convolutional Neural Networks (CNN) [8] are classical methods used in sequence modeling and classification tasks. Among them, Long Short-Term Memory (LSTM) [9] has a unique cell state and gating mechanism that can better capture long-term dependencies and avoid falling into gradient vanishing [10]. It is well utilized in network intrusion detection problems. However, most deep learning methods based on feature extraction use a linear layer for classification processing, and there is a lack of research on classifiers for processing feature sets, limiting their scalability. In addition, deep learning methods based on feature extraction and feature learning have gained widespread attention in recent years. These methods first extract high-dimensional features from the input datasets by a feature extractor and then use the resulting high-dimensional feature sets to train a classifier for classification. In this case, the relationships between the original data features are explored, and the high-dimensional features that are easier to train the classifier are extracted, improving the model's classification accuracy.

Based on the above analysis, this work constructs a multi-classification intrusion detection model based on Stacked Sparse Shrink AutoEncoder (SSSAE), Genetic Simulated annealing-based particle swarm optimization optimized Tabnet classifier (GS-Tabnet), and Decision Fusion (DF) [11], called for SGTD short. First, multiple Sparse Shrink

This work was supported by Beijing Natural Science Foundation (L233005), National Natural Science Foundation of China (NSFC) under Grants 62173013 and 62073005.

AutoEncoders (SSAEs) are stacked to form an SSSAE. Specifically, the feature set extracted from the previous SSAE is used as the input to the subsequent SSAE, which is stacked sequentially to obtain a higher-dimensional feature set based on the previous one. Accordingly, feature extraction is performed using the SSSAE, with one feature set extracted for each SSAE. Then, each feature set is utilized by a GS-Tabnet. It learns the weight of each feature's impact on the detection result through a sequential attention mechanism. Moreover, Genetic Simulated annealing-based Particle Swarm Optimization (GSPSO) [12] optimizes the hyperparameters of the Tabnet [13] to find the combination of hyperparameters that maximizes the classification effect. Finally, the DF fuses the detection results of multiple GS-Tabnet to output the final classification result. SGTD is compared with eight benchmark models, and the experimental results show the effectiveness and accuracy of SGTD in multi-classification network intrusion detection problems.

II. PROPOSED FRAMEWORK

A. Stacked Sparse Shrink Autoencoder

The network structure of an autoencoder [14] is simple, which fails to adequately fit samples when facing the multi-classification problem, leading to poor detection results. In this work, we consider stacking multiple autoencoders, and the features extracted from the former autoencoder are used as inputs to the latter one, forming a stacked autoencoder. Therefore, it has a more complex network structure and can further extract higher-dimensional features [15] from the previous autoencoder, which is suitable for more detailed multi-classification tasks. Moreover, certain neurons in the autoencoder are activated when the model undergoes training, and it relies too much on them. When the input data changes, the overall extraction ability deteriorates, and the generalization ability weakens. To solve this problem, Kullback–Leibler (KL) scatter [16] is introduced as the regular term of the stacked autoencoder, *i.e.*,

$$\mathrm{KL}(\rho \,||\, \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j} \tag{1}$$

where $\hat{\rho}_j$ denotes the activation level of the neuron j during the training process, ρ indicates the average level of activation desired for a neuron. $\mathrm{KL}(\rho \,||\, \hat{\rho}_j)$ represents the gap between $\hat{\rho}_j$ and ρ. Moreover, the model achieves control over the neuron's activation level by making $\mathrm{KL}(\rho \,||\, \hat{\rho}_j)$ as small as possible. When neuronal activation is high, $\mathrm{KL}(\rho \,||\, \hat{\rho}_j)$ also increases and the neural network penalizes this phenomenon during training and tries to keep neuronal activation at the desired level.

Moreover, the dataset contains small disturbances during processing. Thus, it is imperative to make the model robust to the perturbation of the input data. This work introduces a shrinkage loss term based on the Frobenius paradigm [17] of the Jacobian matrix [18], *i.e.*,

$$\parallel J_f(x) \parallel_F^2 = \sum_{ij} \left(\frac{\partial h_j(x)}{\partial x_i}\right)^2 \tag{2}$$

where $\parallel J_f(x) \parallel_F^2$ denotes the Frobenius of the Jacobian matrix for the input x. $h_j(x)$ denotes the neuron j's input representation. The shrinkage loss forces all mappings learned by the autoencoder to have a small gradient concerning the input data. The autoencoder reconstruction loss learns a constant mapping that makes the output equal to the input. Driven by these two losses, most mappings have small gradients concerning the input, while only a few have large gradients. In this case, when the input has small perturbations, the small gradients reduce these perturbations, thereby increasing the robustness of the autoencoder. Moreover, sparse loss and shrinkage loss are added to the stacked autoencoder to construct the SSSAE feature extractor. The loss function of SSSAE is shown in (3).

$$L_{\mathrm{S}}\left(\hat{W}, b\right) = L_{\mathrm{A}}\left(\hat{W}, b\right) + \kappa \parallel J_f(x) \parallel_F^2 + \tau \sum_{j=1}^{J} \mathrm{KL}(\rho \parallel \hat{\rho}_j) \tag{3}$$

where $L_{\mathrm{S}}\left(\hat{W}, b\right)$ denotes the loss function of the SSSAE. $\hat{W}$ and b denote parameter matrices. $L_{\mathrm{A}}\left(\hat{W}, b\right)$ denotes the loss function of a stacked autoencoder. $\kappa \parallel J_f(x) \parallel_F^2$ and $\tau \sum_{j=1}^{J} \mathrm{KL}(\rho \parallel \hat{\rho}_j)$ denote the shrinkage loss and sparse loss, respectively. κ and τ are used to regulate the weights of the two loss terms.

The SSSAE consists of multiple SSAEs. The input data is passed into the first SSAE, and the first feature set is generated in its hidden layer. Then, it is used as the input to the second SSAE, and the second feature set is obtained in the hidden layer of the second SSAE. By analogy, the SSSAE is constructed, and the input data is extracted into multiple feature sets.

B. GS-Tabnet

In this subsection, the Tabnet is introduced and further optimized. The network structure of Tabnet is shown in Fig. 1. It is a multi-step additive model, and each decision step is generated based on the data from the previous decision step through the attentive transformer. The input to each step is a $B \times D$ matrix, where B denotes the number of samples, and D denotes the number of features of the input data.

First, the input samples are passed to the Batch Normalization (BN) layer [19] for normalization, which is then computed by the feature transformer. This process is shown in Fig. 2. Specifically, the feature transformer consists of two parts: a part shared by all decision steps, and a part unique to each decision step. It can prevent large fluctuations in the variance of the model. The Fully Connected layer (FC)+BN+Gated Linear Unit (GLU) structure is the residual connection. After data input, it first passes through the shared across decision steps, which are connected by two FC+BN+GLU structures. Then, it enters the decision step dependent, which has the same structure as the previous one, and this layer is mainly used to calculate the uniqueness of the features. After this, the computation of the common and unique parts of the features is completed. Feature selection

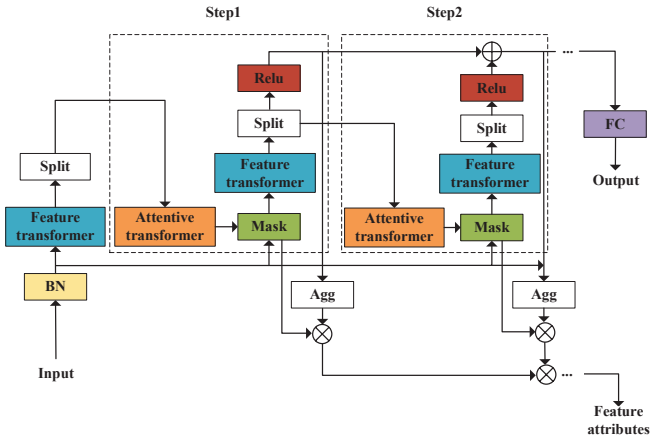

Fig. 1. Structure of the Tabnet.

is realized in the attentive transformer. The structure of the attentive transformer is shown in Fig. 3. Specifically, it consists of FC, BN, Prior scales, and SparseMax. Among them, FC and BN are used to realize linear combinations to get higher dimensional features. Prior scales are the prior information when choosing features in the current decision step [20]. SparseMax is used to regulate the weights of each feature of a sample, thus enabling feature selection.

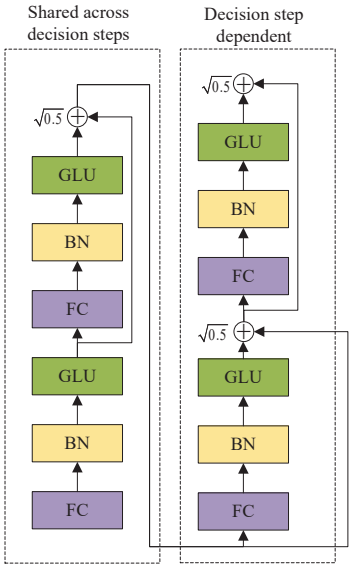

Fig. 2. Structure of feature transformer.

It is worth noting that the Tabnet has several hyperparameters, among which n_d, n_a, and n_s have a greater impact on the final results of the model. n_d represents the width of the decision classification layer, n_a represents the width of the attention embedding, and n_s represents the number of decision steps of the model. It is important to find the optimal combination of parameters so that the classification effect of the model can be maximized.

This work applies GSPSO [21] to optimize the hyperparameter, including n_d, n_a, and n_s. GSPSO is an intelligent

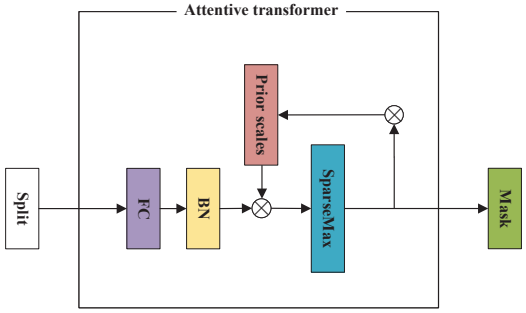

Fig. 3. Structure of attentive transformer.

optimization algorithm that searches the decision space to find the globally optimal solution. In this problem, n_d, n_a, and n_s are decision variables, and the objective function is classification accuracy. It first initializes the position and velocity of each particle and updates its objective function value. Then, it generates the offspring by a certain strategy. Moreover, each individual has a mutation probability, and the Metropolis acceptance rule selects individuals for the next iteration. Finally, the global optimal solution is obtained when the termination condition is met.

C. Combined Feature Decision Fusion Algorithm

A single classifier faces the problem of insufficient stability and generalization ability. Accordingly, this work integrates multiple classifiers and makes comprehensive decisions to improve the model's stability and accuracy. The SSSAE is adopted to extract multiple sets of features. Although each is a higher-dimensional feature set obtained based on the previous one, there is inevitably a loss of information in them. Therefore, a GS-Tabnet classifier is constructed for each feature set to utilize its information. Finally, a decision fusion algorithm is proposed to summarize the classification results of multiple classifiers and output the final classification result. Specifically, the results of the N classifiers are passed into the decision fusion algorithm to output the final network intrusion detection result.

The pseudo codes of the combined feature decision fusion algorithm are shown in Algorithm 1. Specifically, the input sample set is denoted as x, and the number of categories is denoted as k. GS-Tabnet outputs the probability of each sample i belonging to each category j, and it is denoted as p_{ij}. Moreover, the weight coefficient of each classifier is W_i. The final classification result is obtained after the combined feature decision fusion algorithm.

D. Overall Framework of SGTD

Fig. 4 shows the overall framework of SGTD. Specifically, the input dataset is first passed into the first SSAE to extract the input dataset and obtain a feature set. Then, it passes to the next SSAE, stacking N SSAEs to form SSSAE and extracting N feature sets. Next, a GS-Tabnet classifier is trained for each feature set. It can well distribute feature weights through its unique sequential attention mechanism. Moreover, GSPSO optimizes the hyperparameters of Tabnet

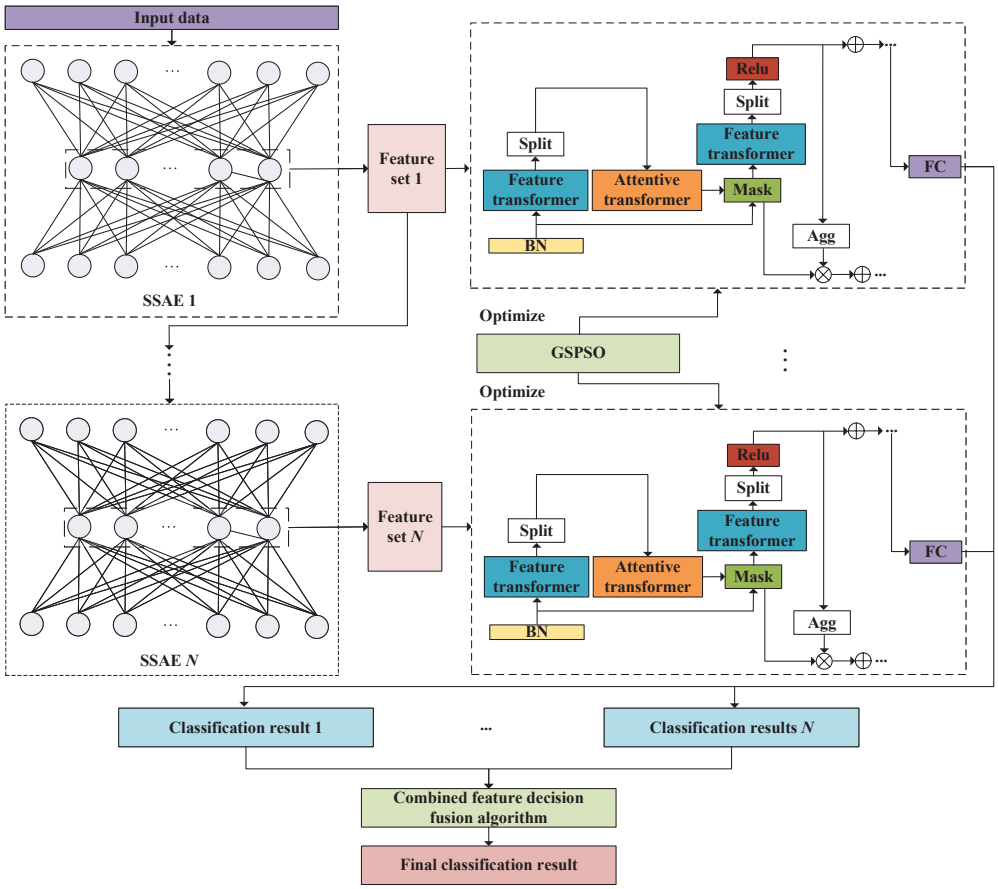

Fig. 4. Structure of SGTD.

Algorithm 1 Combined feature decision fusion algorithm

Initialize weight W_i=0 $i \in \{1, 2, ..., N\}$
while i<N **do**
  For each feature set, GS-Tabnet is adopted to determine whether the classification is correct and obtain $p_{ij}(x)$
  if classified correctly **then**
    W_i++
  end if
  P_{ij}=$W_i p_{ij}(x), j \in \{1, 2, ..., k\}$
  R_i= $\max_{j\in\{1,2,...,k\}} P_{ij}$
end while
Calculate the category $y = \max_{i\in\{1,2,...,N\}} R_i$
Return y

to maximize its classification effect. N GS-Tabnet classifiers output N classification results. Finally, the obtained N classification results are input into the combined feature decision fusion algorithm to output the final classification result.

III. Experimental Results and Discussion

A. Datasets Selection and Evaluation Metrics

1) Datasets: This work adopts the KDD99 dataset that contains records from the military cyber environment in which the attack is embedded. Table I describes the details of the dataset. Specifically, it contains 114 features and has five broad categories: Normal, Probing, U2R, DOS, and R2L. Normal represents normal behavior, and the other four types represent four different kinds of intrusion behavior. The training set contains 494,021 pieces of data, and the test set contains 311,029 pieces of data.

TABLE I
Detailed Information of KDD99

Characteristic number	Categories	Sample size	
		Training set	Test set
114	Normal	97278	60593
	Probing	4107	4166
	U2R	1126	16189
	DOS	391458	229853
	R2L	52	228
	Total	494021	311029

2) Evaluation Metrics: The accuracy rate is chosen as the evaluation index of the model. Moreover, because network intrusion detection is a multi-classification problem, there are multiple categories of data. Thus, it is necessary to consider the weighted average of the samples of each category as the final result. F_1 score is adopted as an evaluation indicator, *i.e.*,

$$F1 = \sum_{j=1}^{J} N_j (F1)_j \tag{4}$$

where N_j denotes the proportion of the sample size of category j to the total sample size. $(F1)_j$ denotes the F_1 score of category j.

B. Hyperparameter Tuning

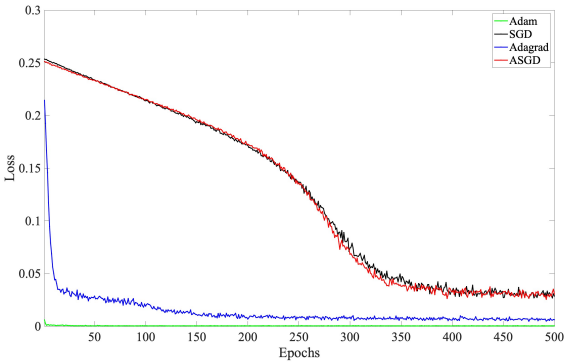

Fig. 5. Comparison of different optimizers and epochs of AE.

TABLE II
COMPARISON OF DIFFERENT ACTIVATION FUNCTIONS OF AE.

Serial number	Activation function	Accuracy	$F1$
1	Sigmoid	88.37%	86.22%
2	**Relu**	**88.69%**	**86.98%**
3	Leaky Relu	88.50%	86.35%
4	Tanh	87.37%	85.81%

TABLE III
COMPARISON OF DIFFERENT HIDDEN LAYER NUMBER AND HIDDEN LAYER NEURON OF AE

Hidden layer	Hidden layer neuron	Accuracy	$F1$
1	32	89.32%	87.58%
1	64	88.78%	85.98%
1	96	88.03%	85.77%
2	32-64	89.15%	87.69%
2	64-96	90.11%	87.91%
2	32-96	89.27%	87.90%
3	**32-64-96**	**90.27%**	**87.95%**

The comparative models include AutoEncoder (AE), LSTM, CNN, and Variational AutoEncoder (VAE). Each model's parameter selections are conducted to optimize the model detection effect. For AE, the training optimizer directly affects the model's fitting degree. Four optimizers (Adam, ASGD, SGD, and Adagrad) are chosen to be trained 500 times each. The best optimizer is chosen by comparison. It is shown in Fig. 5 that the Adam optimizer achieves the minimal loss. Moreover, AE's activation function needs to be determined with experiments. The Sigimoid, Relu, Leaky Relu, and Tanh functions are selected for the experiments. The results are shown in Table II. It is concluded that AE achieves the best classification results when using the Relu function. Finally, the number of hidden layers of the AE needs to be determined, where the number of hidden layers is selected in (1,2,3), and the hidden layer neurons are selected in (32,64,96). The experimental results are shown in Table III. It is shown that when the hidden layer of AE is three, and the neurons in each layer are 32, 64 and 96, the AE intrusion detection model achieves the optimal accuracy and $F1$. Due to space issues, the tuning process for the rest of the models is similar and not shown in detail. For SGTD, Adam and Relu are selected as the optimizer and activation function, respectively. SSSAE stacks 3 SSAEs with 32, 64 and 96 hidden layer neurons per SSAE. τ=0.4 and ρ=0.6. Moreover, the effectiveness of GSPSO is shown in Table IV. It is shown that GSPSO optimized Tabnet can achieve higher accuracy in network intrusion detection and avoid the bias of manual parameter adjustment.

TABLE IV
COMPARISON OF GSPSO AND MANUAL PARAMETERS ADJUSTMENT

Serial Number	n_d	n_a	n_s	Accuracy	$F1$
1	4	4	3	89.29%	87.11%
2	4	6	3	90.42%	88.70%
3	4	8	3	90.03%	88.17%
4	6	4	3	88.74%	86.90%
5	6	6	3	88.91%	87.23%
6	6	8	3	90.17%	88.39%
7	8	4	3	88.54%	87.02%
8	8	6	3	88.87%	86.78%
9	8	8	3	89.49%	87.56%
10	4	4	3	90.23%	88.41%
11	4	6	3	90.15%	88.29%
12	4	8	3	89.55%	87.99%
13	6	4	3	88.64%	86.35%
14	6	6	3	87.80%	85.98%
15	6	8	3	90.31%	87.77%
16	8	4	3	88.68%	86.78%
17	8	6	3	90.48%	88.68%
18	8	8	3	90.89%	89.01%
GSPSO	**3**	**6**	**6**	**91.65%**	**89.76%**

C. Experimental Results and Discussion

In addition to the LSTM, CNN, AE and VAE, this work also uses AE and VAE as feature extractors combined with LSTM and CNN classifiers to construct AE-LSTM, AE-CNN, VAE-LSTM, and VAE-CNN models. As a result, after parameter tuning, SGTD is compared with eight benchmark models on the KDD99 dataset for intrusion detection. Table V depicts the accuracy of SGTD with other benchmark models. It is shown that the SGTD achieves a higher accuracy and F_1 than the other eight benchmark models, proving its effectiveness in network intrusion detection.

TABLE V
COMPARISON OF DETECTION EFFECTS OF DIFFERENT MODELS ON KDD99 DATASET

Serial Number	Model	Accuracy	$F1$
1	LSTM	90.25%	87.89%
2	CNN	89.78%	87.45%
3	AE	90.27%	87.67%
4	VAE	90.12%	88.03%
5	AE-LSTM	90.45%	88.23%
6	AE-CNN	90.36%	88.06%
7	VAE-LSTM	90.29%	88.12%
8	VAE-CNN	90.15%	88.16%
9	**SGTD**	**91.65%**	**89.76%**

D. Ablation Experiments

The SGTD consists of three parts: SAE, GS-Tabnet, and the combined feature decision fusion algorithm. Ablation experiments are performed to validate the effectiveness of each part. The results are shown in Table VI. Among them, No. 1-3 indicate network intrusion detection with SAE, SSSAE and SSSAE-Tabnet. No. 4 adds GSPSO to optimize Tabnet, and No. 5 adds the combined feature decision fusion algorithm, which is the SGTD. It is shown that each part in the SGTD contributes positively to the model to improve detection accuracy, proving the validity of SSSAE, GS-Tabnet, and the combined feature decision fusion algorithm.

TABLE VI
SGTD ABLATION EXPERIMENT

Serial Number (No.)	Model	Accuracy	$F1$
1	SAE	90.19%	87.72%
2	SSSAE	90.40%	88.06%
3	SSSAE-Tabnet	90.89%	88.98%
4	SSSAE-GS-Tabnet	91.12%	89.34%
5	**SGTD**	**91.65%**	**89.76%**

IV. CONCLUSIONS

Network intrusion detection is important in constructing network security prevention and control systems, which is significant in maintaining network security. However, traditional network intrusion detection methods are often adopted to directly process the original data, resulting in incomplete consideration of the correlation between features and interference by invalid features. Moreover, using a single classifier has problems of poor generalization and stability. This work proposes a novel multi-classification intrusion detection model based on Stacked Sparse Shrink AutoEncoder (SSSAE), Genetic Simulated annealing-based particle swarm optimization optimized Tabnet classifier (GS-Tabnet), and Decision Fusion (DF), called for SGTD short. Among them, SSSAE is designed to extract multiple features from the input data. Then, GS-Tabnet trains a classifier for each feature set. Finally, the combined feature decision fusion algorithm fuses the results from these classifiers to obtain the final classification result. SGTD is compared with eight multi-classification benchmark models. The experimental results demonstrate the effectiveness and accuracy of SGTD in multi-classification network intrusion detection.

In future work, we will consider further accessing the time dimension to obtain the time information of each network behavior to achieve more comprehensive intrusion detection. Moreover, we use SSSAE to extract features from the input data, which is unsupervised learning. To improve the effectiveness of the feature extraction, labels can be introduced in the training phase of the feature extractor, outputting a more differentiated feature set.

REFERENCES

[1] X. Chen, M. Liu, and S. Li, "Echo State Network With Probabilistic Regularization for Time Series Prediction," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 8, pp. 1743–1753, Aug. 2023.

[2] H. Yuan, S. Wang, J. Bi, J. Zhang, and M. Zhou, "Hybrid and Spatiotemporal Detection of Cyberattack Network Traffic in Cloud Data Centers," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2024.3360294.

[3] P. Marteau, "Sequence Covering for Efficient Host-Based Intrusion Detection," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 994–1006, Apr. 2019.

[4] M. Zhang, D. Liu, Q. Wang, B. Zhao, O. Bai, and J. Sun, "Gait Pattern Recognition Based on Plantar Pressure Signals and Acceleration Signals," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–15, Sept. 2022.

[5] Q. Tang, W. Qiu, and Y. Zhou, "Classification of Complex Power Quality Disturbances Using Optimized S-Transform and Kernel SVM," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, pp. 9715–9723, Nov. 2020.

[6] J. Bi, Z. Wang, H. Yuan, J. Zhang, and M. Zhou, "Self-adaptive Teaching-learning-based Optimizer with Improved RBF and Sparse Autoencoder for High-dimensional Problems," *Information Sciences*, vol. 630, pp. 463–481, Jun. 2023.

[7] Z. Xie, L. Jin, X. Luo, Z. Sun, and M. Liu, "RNN for Repetitive Motion Generation of Redundant Robot Manipulators: An Orthogonal Projection-Based Scheme," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 615–628, Feb. 2022.

[8] Z. Wang, H. Lu, Y. Shi, and X. Wang, "Lightweight CNN Architecture Design Based on Spatial-Temporal Tensor and Its Application in Bearing Fault Diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, Nov. 2023.

[9] H. Yuan, J. Bi, S. Li, J. Zhang, and M. Zhou, "An Improved LSTM-Based Prediction Approach for Resources and Workload in Large-scale Data Centers," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2024.3383512.

[10] H. Yuan, J. Bi, Z. Wang, J. Yang, and Jia Zhang, "Partial and Cost-minimized Computation Offloading in Hybrid Edge and Cloud Systems," *Expert Systems with Applications*, vol. 250, pp. 1–13, Sept. 2024.

[11] W. Yang, B. Chen, and L. Yu, "Bayesian-Wavelet-Based Multisource Decision Fusion," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, Jul. 2021.

[12] J. Bi, Z. Wang, H. Yuan, J. Zhang, and M. Zhou, "Cost-Minimized Computation Offloading and User Association in Hybrid Cloud and Edge Computing," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2024.3354348.

[13] Z. Liu, Q. Fan, Z. Wang, and Y. Cai, "A Novel Algorithm for Credit Default Prediction using TabNet," *2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 2023, Changchun, China, pp. 24–27.

[14] H. El-Fiqi, M. Wang, K. Kasmarik, A. Bezerianos, K. C. Tan, and H. A. Abbass, "Weighted Gate Layer Autoencoders," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 7242–7253, Aug. 2022.

[15] F. Cheng, F. Chu, Y. Xu, and L. Zhang, "A Steering-Matrix-Based Multiobjective Evolutionary Algorithm for High-Dimensional Feature Selection," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9695–9708, Sept. 2022.

[16] S. Ji, Z. Zhang, S. Ying, L. Wang, X. Zhao, and Y. Gao, "Kullback–Leibler Divergence Metric Learning," *IEEE Transactions on Cybernetics*, vol. 52, no. 4, pp. 2047–2058, Apr. 2022.

[17] Y. Huang, G. Liao, Y. Xiang, L. Zhang, J. Li, and A. Nehorai, "Low-Rank Approximation via Generalized Reweighted Iterative Nuclear and Frobenius Norms," *IEEE Transactions on Image Processing*, vol. 29, pp. 2244–2257, Jun. 2020.

[18] X. Yu, J. Zhao, H. Zhang, X. Wang, and X. Bian, "Data-Driven Distributed Grid Topology Identification Using Backtracking Jacobian Matrix Approach," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 2, pp. 1711–1720, Feb. 2024.

[19] H. Peng, Y. Yu, and S. Yu, "Re-Thinking the Effectiveness of Batch Normalization and Beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 1, pp. 465–478, Jan. 2024.

[20] Y. Park and J. C. Ho, "Tackling Overfitting in Boosting for Noisy Healthcare Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 7, pp. 2995–3006, Jul. 2021.

[21] H. Yuan, Q. Hu, M. Wang, J. Bi, and M. Zhou, "Cost-minimized User Association and Partial Offloading for Dependent Tasks in Hybrid Cloud–edge Systems," *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, Mexico City, Mexico, pp. 1059–1064.