applied sciences

MDPI

Article

An Intrusion Detection Method Based on Hybrid Machine Learning and Neural Network in the Industrial Control Field

Duo Sun, Lei Zhang *, Kai Jin, Jiasheng Ling and Xiaoyuan Zheng

School of Artificial Intelligence and Data Science, Hebei University of Technology, Tianjin 300132, China; 202132803145@stu.hebut.edu.cn (D.S.); 202132803089@stu.hebut.edu.cn (K.J.); 202232804040@stu.hebut.edu.cn (J.L.); 2021019@hebut.edu.cn (X.Z.)

* Correspondence: zhanglei@hebut.edu.cn

Abstract: Aiming at the imbalance of industrial control system data and the poor detection effect of industrial control intrusion detection systems on network attack traffic problems, we propose an ETM-TBD model based on hybrid machine learning and neural network models. Aiming at the problem of high dimensionality and imbalance in the amount of sample data in the massive data of industrial control systems, this paper proposes an IG-based feature selection method and an oversampling method for SMOTE. In the ETM-TBD model, we propose a hyperparameter optimization method based on Bayesian optimization used to optimize the parameters of the four basic machine learners in the model. By introducing a multi-head-attention mechanism, the Transformer module increases the attention between local features and global features, enabling the discovery of the internal relationship between features. Additionally, the BiGRU is used to preserve the temporal features of the dataset, while the DNN is used to extract deeper features. Finally, the SoftMax classifier is used to classify the output. By analyzing the results of the comparison and ablation experiments, it can be concluded that the F1-score of the ETM-TBD model on a robotic arm dataset is 0.9665 and the model has very low FNR and FPR scores of 0.0263 and 0.0081, respectively. It can be seen that the model in this paper is better than the traditional single machine learning algorithm as well as the algorithm lacking any of the modules.

Keywords: industrial control system; machine learning; neural networks; intrusion detection

Citation: Sun, D.; Zhang, L.; Jin, K.; Ling, J.; Zheng, X. An Intrusion Detection Method Based on Hybrid Machine Learning and Neural Network in the Industrial Control Field. *Appl. Sci.* **2023**, *13*, 10455. https://doi.org/10.3390/app131810455

Academic Editor: Gianluigi Ferrari

Received: 5 September 2023
Revised: 13 September 2023
Accepted: 16 September 2023
Published: 19 September 2023

1. Introduction

1.1. Research Background

With the rapid development of the internet, the in-depth integration of informatization and industrialization has been promoted, leading to the birth of the industrial internet platform. This integration has brought both opportunities and challenges. The Industrial Control System (ICS) is a management and control system composed of various programmable logic controllers (PLC) and real-time data collection systems [1]. As the scope of application of industrial control systems continues to expand, they have been applied to various important national infrastructures and people's livelihood service facilities [2]. For example, starting from 7 March 2019, a large-scale power outage occurred in Venezuela that lasted for 6 days, setting a new record for the largest power outage in the world so far. Among the 23 states in Venezuela, 20 states had a total power outage at one point. The power outage caused the Caracas Metro to fail to operate, causing large-scale traffic congestion. Schools, hospitals, factories, airports, etc. were seriously affected, and mobile phones and the internet were also unable to work and be used normally. After the Venezuelan power outage, through the relevant analysis of the industrial control network system before and after the Venezuelan power outage, it was found that all data of the industrial control network system changed significantly after the power outage. As a result, it has been suggested that this was the result of an unspecified cyberattack. These cyberattacks caused widespread damage to Venezuela's electrical system by directly injecting an

attack code. The large-scale power outage has brought heavy losses to Venezuela, and it has also served as a serious warning to countries around the world, including China, that there must be no slack in safeguarding the security of the country's critical infrastructure. The security of industrial control network systems has become the nerve center of critical infrastructure and even the entire economic society. Major security incidents can lead to the paralysis of energy, transportation, communication, finance and national defense infrastructure, resulting in catastrophic consequences and seriously endangering national security and public interests.

Intrusion detection, as the second line of defense in the defense in-depth strategy of the industrial control system, is a necessary supplement to the firewall [3]. As an important part of the industrial control system, intrusion detection can monitor network traffic and system behavior, detect and prevent potential attacks in time, and ensure the security and stable operation of industrial control systems [4]. The resulting network intrusion detection technology recognizes and classifies network traffic, discovers malicious intrusion traffic in time and reports it to users to remind users to take further measures to prevent major security incidents. However, due to the continuous growth of the scale of the internet, the network traffic has also shown explosive growth, and its dimensional characteristics have become more and more complex. Therefore, improving the accuracy of detecting malicious traffic and efficiently distinguishing different types of malicious traffic has become a top priority [5,6].

1.2. Related Works

Machine learning and neural network models have achieved good research results in the fields of natural language processing and image recognition. They are gradually applied to intrusion detection in industrial control systems due to their ability to mine high-dimensional and large-scale data [7–9].

Kanna, P.R. identifies attack patterns by detecting and analyzing signatures [10]. An efficient hybrid IDS model is presented which is built using a MapReduce-based Black Widow Optimized Convolutional-Long Short-Term Memory (BWO-CONV-LSTM) network. The time complexity of the BWO-CONV-LSTM network in parameter optimization is too large and the time cost is larger than the training cost of this paper's model for machine learning. Foley, J. proposes that anomaly-based recognition links consumer actions to predetermined states to detect suspicious behaviors that may be intrusions, but this method often has significant false positive scores [11]. Machine learning has been analyzed in the literature only for cyberattacks on a single dataset, but he has the advantage of being able to analyze for growing datasets, which is an incremental learning process. Vinayakumar, R, based on hybrid detection technology, used multiple detection methods to improve the performance of specific methods, which can reduce the false positive rate and provide effective detection [12]. The literature explores a deep neural network-based model to deal with unforeseen and unpredictable cyberattacks. However, our goal in this task is to improve the detection rate against known attacks, so the deep learning model proposed in the literature is slightly slower in terms of training speed. Peng, W. proposed an IDS based on deep AE and ML methods, which was greatly improved in terms of accuracy and time complexity compared with the algorithms at that time [13]. The literature does not go far enough in feature extraction. Many redundant features are not avoided, which leads to excessive complexity in model training as well as reduced accuracy. Othman, S.M. adopted the fundamentals of AE to introduce a multiphase model, incorporating the ID convolution operation and stacked layers. Two different AEs were trained and evaluated using normal flow and attack flow [14]. The literature has mainly investigated performance for training on large data, which is not quite in line with the goals of our task. This is because the amount of data in the network traffic of industrial control systems is often not particularly large. And when dealing with small sample data, the SVM classifier does not classify as well as when dealing with large data. Lv, H. proposed a new data enhancement algorithm for DDoS attacks [15]. The literature mainly addresses the problem of unbalanced datasets

in the face of DDoS attack types. In dealing with unbalanced datasets, it is similar to our task. In this paper, we also draw on SMOTE-based ideas to deal with unbalanced datasets. Chen, C. proposed an SCV-GA-WELM intrusion detection model for the problem of poor detection in unbalanced environments. They use SCV to ensure consistent data distribution for all subsets to avoid the data imbalance problem [16]. Kilichev, D. has conducted a comprehensive study on the hyperparameter optimization of 1DCNN models. They used two well-established computational methods, GA and PSO, to optimize all nine hyperparameters in the model and evaluated their performance on three different datasets. They demonstrated the importance of hyperparameter optimization in training intrusion detection models [17]. Yang, H. also makes an outstanding contribution to solving the imbalance problem of sample classes. They proposed an SPE-ACGAN-based generative adversarial network model to oversample the minority class samples and undersample the majority class samples based on SPE [18].

In this paper, we propose an SMOTE-based oversampling method for unbalanced data. This is the more mainstream way to deal with the data imbalance problem nowadays. Further, we propose a hybrid machine learning and neural network model, which is a newer approach in the field of network intrusion detection. In the next subsection, we will show the main contribution points of this paper in detail.

1.3. Main Contributions

The contributions of this paper are mainly categorized into the following three points:

1. This paper proposes the IG-based feature selection method and SMOTE method, which effectively reduces the dimensionality of massive data and solves the problem of data imbalance [19–21];
2. We propose the ETM-TBD intrusion detection model to improve the efficiency and accuracy of detecting potential security threats. We propose some neural network models on top of the machine learning base model to further optimize the accuracy of the model;
3. We propose a neural network model based on Transformer and BiGRU [22] models, which improves the attention between local and global features, preserves the temporal features of the original data, and effectively improves the accuracy of the model.

2. Materials and Methods

2.1. Dataset Introduction

The dataset in this paper is collected from an industrial robotic arm control laboratory in March 2023. This dataset collects hundreds of thousands of samples, including data on normal system operation, data on DDoS attacks, and scene data of other types of attacks.

In general, the dataset contains 78 columns and 1,048,575 rows (704,402 are scene samples where the system is not under attack during normal operation, 4064 are scene samples where the system is subjected to violent intrusion, 338,326 are scene samples where the system is attacked by DDoS, 56 are scene samples where the system suffers from SQL injection attacks and 1727 are scene samples where the system suffers from XSS attacks), as shown in Table 1.

Table 1. Initial dataset details table.

Type	Records	Percentages
BENIGN	704,402	67.18%
Brute Force	4064	0.39%
DDoS	338,326	32.27%
Sql Injection	56	0.005%
XSS	1727	0.16%

More detailed data processing will be shown in Section 4.2.

Firstly, the raw data is standardized and normalized into a dataset with a small range of numerical fluctuations. In addition, because the labeled data of the dataset is string type data, we need to character encode it into five numerical types such as 0, 1, 2, 3, 4, etc., and the numerical types correspond to the string type data, respectively. In addition, we performed an IG-based feature selection process for 78 columns of data features, and finally selected 20 columns of the most effective and basically nonredundant feature data. For less sample data, we propose SMOTE-based methods to avoid class imbalance. Finally, the data was divided into training and testing sets—80% of the data was used for training and 20% for testing. The training set data was randomly shuffled to improve the effectiveness and robustness of the training.

2.2. Feature Engineering

Generally, we generate a high-quality and representative subdataset through data preprocessing. Then, we can remove some redundant features from the dataset using feature engineering works, which make the model training more accurate and efficient. The model in this paper first proposes to remove irrelevant features after the preprocessing work using an integrated feature engineering method consisting of Information Gain (IG) [23] and Fast Correlation-Based Filter (FCBF) [24] to retain the important features.

(1) Information Gain (IG): The IG algorithm is a commonly used feature selection method to select important features. IG indicates the amount of information obtained or entropy change, which can be used to measure how much information a feature can bring to the target variable. Moreover, the time complexity of the IG algorithm is O(n), and it can quickly obtain the importance score of each feature, so we believe that the IG algorithm is well suited to our preconceived ideas in the feature engineering phase. We can calculate the importance score for each feature. This will help us to select some features that are most relevant to the detection task. Assuming that T is the target variable, for each feature represented by the random variable X, the IG value of the feature X is represented by (1):

$$IG(T \mid X) = H(T) - H(T \mid X) \tag{1}$$

where $H(T)$ is the entropy of the target variable T, and $H(T \mid X)$ is the conditional entropy of $T \mid X$.

(2) Fast Correlation-Based Filter (FCBF): Although the IG-based feature selection method can remove some unimportant features while guaranteeing a low time complexity, we can still find many redundant features that are not removed. Feature redundancy may increase the time complexity and space complexity and also increase the probability that the model will be misled by the redundant data. In that case, the performance of our model may be degraded and may also increase the risk of overfitting. Therefore, we will introduce the FCBF algorithm to solve the above problem. We remove the redundant features by calculating the correlation of the input features, which can help to improve the performance and efficiency of the model. In FCBF, the symmetric uncertainty (SU) is calculated to measure the correlation between features by normalizing the IG value, as shown in Formula (2):

$$SU(X, Y) = 2\frac{IG(X \mid Y)}{H(X) + H(Y)} \tag{2}$$

$SU(X,Y)$ is in the range [0,1]; a value of 1 means that the two features of X and Y are completely correlated, and a value of 0 means that the two features are completely independent. The FCBF algorithm searches for features in the feature space according to the SU value of the feature space until the entire feature space is searched. We will consider highly correlated features as redundant and keep only one of them. In the proposed FCBF algorithm, the SU value of each pair of features is calculated as their correlation. The correlation threshold α is optimized by BO-GP [25]. The BO-GP is a Gaussian Process-based Bayesian Optimization algorithm for solving the global optimal solution of a black-box function. The principle of BO-GP algorithm is to approach the global optimal solution step

by step by continuously evaluating the objective function and adjusting the position of the next evaluation according to the evaluation result. When the correlation value between two features is greater than α, the feature with higher feature importance is retained, and the feature with lower feature importance is discarded. Repeat the correlation calculation and feature rejection process until every pair of features in the feature list is not highly correlated (SU <= α). In this paper, the model combining the IG algorithm and the FCBF algorithm is called IG-FCBF.

2.3. Machine Learning Tree

After the data preprocessing work and feature engineering work, we can obtain a highly representative subdataset. An integrated machine learning tree model proposed in this paper will train the obtained dataset to train a signature-based intrusion detection model. In the proposed signature-based intrusion detection model, four tree-based machine learning algorithms, Support Vector Machine (SVM), Random Forest (RF), Extra Tree (ET) and Extreme Gradient Boosting (XGBoost), are selected as the basic learners [26–28].

The main idea of SVM is to divide the dataset into two classes by finding an optimal hyperplane and finding a maximum interval between the two classes after division, which makes the model more generalizable and robust. The RF algorithm is an ensemble learning model that combines multiple decision tree classifiers using majority voting rules. The ET algorithm, on the other hand, combines random collections of decision trees built on different subsets of the dataset. The XGBoost is an algorithm based on gradient-boosting decision trees, designed to improve speed and performance. Obviously, for RF, ET and XGBoost the hyperparameters of DT are also their important hyperparameters. And they all have a basic hyperparameter that needs to be adjusted, which is the number of basic DTs built for each model—n estimator. These parameters have a very important impact on the performance of the model. In addition, XGBoost has a hyperparameter called learning rate, which determines the speed of convergence.

We set the number of instances to be n, the number of features to be f, and the number of DTs in the integrated model to be t. Then, the time complexities of RF, ET and XGBoost are $O(n^2\sqrt{ft})$, $O(nft)$ and $O(nft)$, respectively. In the training phase, the time complexity of SVM is usually $O(n^3)$, and in the testing phase, the time complexity of SVM depends mainly on the number of support vectors, which is usually $O(kf)$, where k is the number of support vectors.

In this paper, we will first train these four base tree models and then further stack them using integrated learning to improve the model performance. Stacking is a standard ensemble learning technique. With stacking, information from four base learners can be learned simultaneously, thereby reducing the error of a single learner and obtaining a more reliable and robust metaclassifier. In this paper, we use the output labels of the four basic learners (SVM, RF, ET and XGBoost) as the input features of the integrated model. Thereby, we can train a more robust metalearner that can be used to further improve the recognition accuracy.

2.4. Transformer Module

We know that the original model of the Transformer model includes both encoding and decoding parts. But due to the intrusion detection, traffic is characterized by the length of each datum which is of fixed length. Therefore, we have conducted it in our proposed model mainly based on the encoding part of Transformer model and modified some of his parameters [29]. This part includes a multihead attention mechanism and a feed-forward neural network. The attention mechanism in dot product attention contains three inputs which are query, key and value. The Transformer model computes the weight scores for each feature using query and key, and then computes the weighted sum of the weights and

values for each feature to obtain the output. Using point active attention can be operated in parallel to reduce training time. Its calculation Formula (3) is as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (3)$$

In Formula (3), Q, K and V represent the three matrices of Query, Key and Value, respectively, and d_k is the dimension of Key. In this paper, in order to enrich the extracted features, a multihead attention structure is used. The calculation Formulas (4)–(6) of multihead attention are as follows:

$$Q_i = QW_i^Q, K_i = KW_i^K, V_i = VW_i^V, i = 1, \ldots, n \quad (4)$$

$$head_i = Attention(Q_i, K_i, V_i), i = 1, \ldots, n \quad (5)$$

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_n)W^Q \quad (6)$$

Since the *GELU* activation function involves an error function (erf), and the error function is a nonlinear function, noise will be introduced. In some cases, introducing output randomness can be beneficial, as it can help the neural network explore more solutions and avoid overfitting. Therefore, using the Gaussian error linear unit activation function *GELU* as the activation function increases randomness, as shown in Formula (7):

$$GELU(x) = 0.5x(1 + \tanh(\sqrt{2/\Pi}(x + 0.0444715x^3))) \quad (7)$$

2.5. BiGRU-DNN Module

The Gated Recurrent Unit network (GRU) is a cyclic neural network RNN that solves the loss of long-distance information [30–33]. It is mostly used to process timing information and solve the problems of gradient explosion and gradient disappearance in the RNN structure. The most important thing is that it can memorize the information effectively and give up the memory of redundant information. The Bidirectional Gated Recurrent Unit network (BiGRU) is composed of a forward GRU and a reverse GRU, which contains all the information of the forward and backward directions, and its structure is shown in Figure 1.

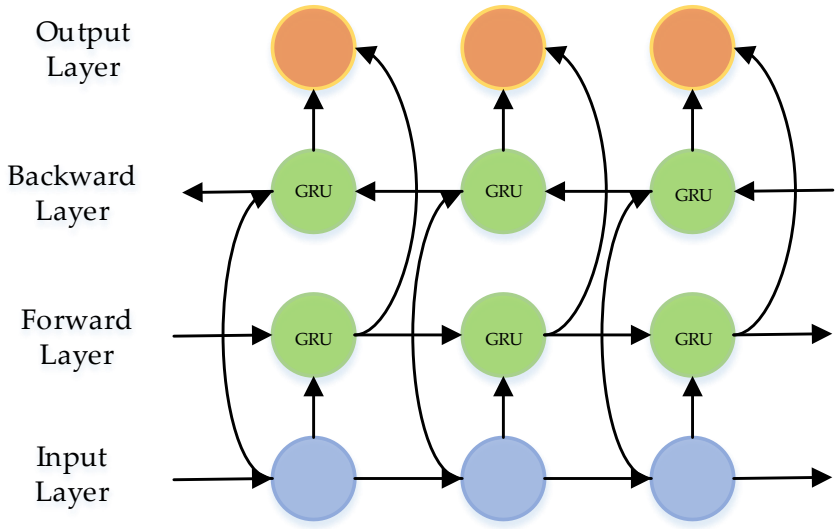

Figure 1. BiGRU structure.

Firstly, the data is fed into the Forward Network and the Backward Network through the Input Layer. After being processed by the Forward Network and the Backward Network,

the data is spliced in the Output Layer. The results after the Forward Network and the Backward Network processing will be used as inputs to the next layer of the proposed model. In this paper, the output dimension of the BiGRU model will be set as two times the input dimension. This can effectively reduce the complexity of the model. Its output is shown in Formula (8):

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \tag{8}$$

The $\overrightarrow{h_i}, \overleftarrow{h_i}$ in Formula (8) denote the results after processing of the Forward Network and the Backward Network, respectively.

Deep neural network (DNN) model is an ordered combination of multiple perceptual machines and is a commonly used deep learning network, which is also called Multilayer perceptron. In this paper, the DNN model is used to fit the output of the previous layer. In this model, DNN has two hidden layers and uses RELU as the activation function. The general Formula (9) for DNN calculation is as follows:

$$f(x) = \sigma(WX + b) \tag{9}$$

3. Proposed Model

3.1. Algorithm Overview

In the field of industrial control security, machine learning can effectively process a large number of network data packets and analyze the massive data, which in turn can be mined for potential correlations and patterns between network data. However, machine learning also has obvious limitations that it requires a large amount of data for training and optimization. If the amount of data is insufficient or the quality is not good, it may lead to a decrease in the accuracy and generalization ability of the model. The Transformer module and the introduced BiGRU and DNN models will solve the problem of the weak generalization ability of machine learning when processing a small number of samples. They can highlight the attention between different features and between local and global features and tap into the intrinsic connections between features.

Therefore, this paper proposes a hybrid machine learning and neural network model for an ETM-TBD intrusion detection model. In the following subsections, we will describe the model in detail.

3.2. Model Structure and Algorithm Flow

In this paper, we propose an ETM-TBD intrusion detection model, shown in Figure 2. The model mainly consists of four parts: an Ensemble Tree Models module, a Transformer module, a BiGRU module and a DNN module. The Ensemble Tree Models module adopts the integrated learning model that stacks four basic machine learning trees. In this paper, four tree-based machine learners, namely SVM, RF, ET and XGBoost, are first trained and the individual machine learners are optimized with the BO-GP method. Then, the outputs of four basic machine learning trees for learning are integrated using a stacked integration model, and the results are then optimized by the BO-GP method, and the stacked integrated model is used as the first layer of the ETM-TBD model. We then input the results of the integrated machine learning tree as new features into the processed data. The Transformer module is trained on the newly constructed data to establish relationships between the different features. Afterward, the BiGRU neural network is used in conjunction with the Transformer model to obtain the connection between the front and rear features. The proposed DNN model further extracts the features, and the last layer of the model uses a SoftMax classifier to identify and classify the features to obtain the results.

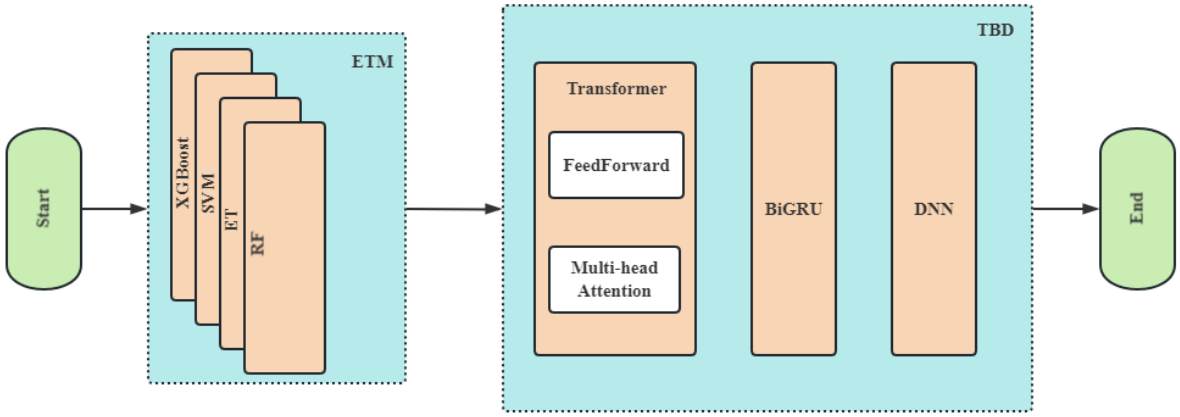

Figure 2. ETM-TBD model.

The overall framework of the proposed ETM-TBD model is shown in Figure 3, and the algorithm flow is as follows. The first step is the data preprocessing work, where the partitioning of the training and test sets is performed. Next comes the training module, which is in the lower left position in the figure. The detailed steps of the ETM module and TBD module are shown in the upper right and lower right corners of the figure. The results of the ETM model will be parameter optimized using the BO algorithm, and in the next step, it will be trained with a mixture of TBD models. And the BO module in the top-right corner of the figure is used to interpret the Bayesian Optimization model (BO) in the ETM module.

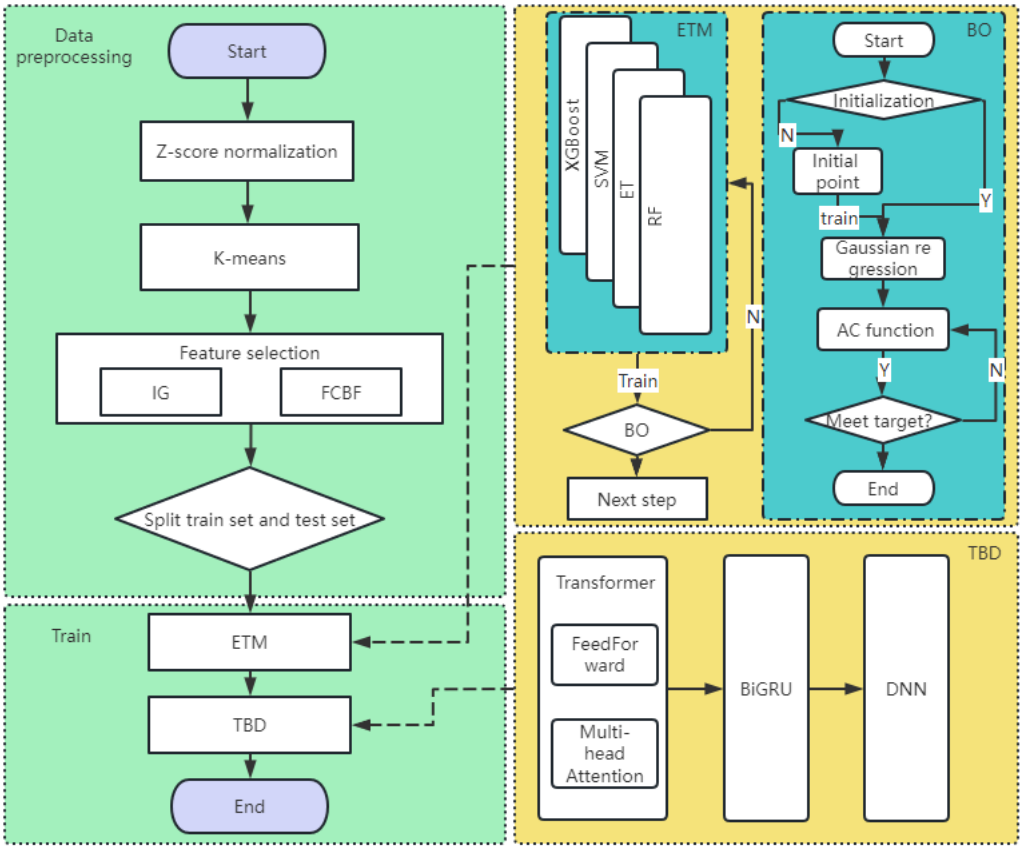

Figure 3. Algorithm flowchart.

The modules in the flowchart will be described in detail next.

(1) Z-score normalization: The range of values of the raw data collected through the industrial control system is very large. We cannot directly process the raw data directly because it may trigger the problem of gradient explosion. Therefore, we will use normalization to scale each feature dimension of the raw data to between -1 and 1.

(2) K-means: The K-means clustering algorithm is used for data sampling. In order to improve the efficiency of model training, data sampling is a commonly used technique, which can generate a subset of the original data. It will randomly select a portion of the data in each category labeled data and then uniformly integrate it into a new subdataset. The k-means algorithm can well reduce the time complexity of training.

(3) Feature selection: Obtaining an optimal feature list through proper feature engineering can also improve the quality of the dataset. Then, we can remove some redundant features from the dataset by appropriate feature engineering, which can make the model training more accurate and efficient. In this paper, we propose the method based on IG-FCBF to extract effective features.

(4) ETM module: After the data preprocessing work and feature engineering work, we can obtain a highly representative subdataset. An integrated machine learning tree model proposed in this paper will train the obtained dataset to train a signature-based intrusion detection model which is called ETM model. Using the output labels of four basic learners (SVM, RF, ET and XGBoost) as input features, a powerful metalearner is trained for further prediction. Using stacking, the information of four base learners can be learned simultaneously, thereby reducing the error of a single learner and obtaining a more reliable and robust metaclassifier.

(5) BO module: Hyperparameter optimization (HPO) is the process of using an optimization algorithm to build an optimized machine learning model for a specific problem or dataset. Bayesian Optimization (BO) algorithms are a set of efficient HPO algorithms that determine the next hyperparameter value based on previous evaluation results. Therefore, in this step we optimize the parameters of each single module of the ETM model as well as the integrated ETM model using the BO algorithm.

(6) Transformer module: The results of the ETM model are input into the processed data as a new feature. The Transformer module is trained on the newly constructed data to establish relationships between the different features.

(7) BiGRU-DNN module: We will propose a hybrid model called a TBD model, which is an integration of the Transformer model, BiGRU model and DNN model. The BiGRU neural network is used in conjunction with the Transformer model to obtain the connection between the front and rear features. The proposed DNN model further extracts the features and the last layer of the model uses a SoftMax classifier to identify and classify the features to obtain the results.

4. Experiment

4.1. Experiment Settings

Experiments are coded based on the PyTorch framework. The platform used is PyCharm 2022.1 $\times$ 64, the operating system is Windows 11, and the hardware device used is NVIDIA GeForce RTX 3050.

4.2. Data Preprocessing

(1) Numerical processing: In this paper, the numerical processing of data uses the currently more commonly used label encoding. Label encoding is an encoding method that maps categorical variables to integer labels. Its basic idea is to assign a distinct integer label to each distinct categorical variable, and these labels are usually continuous integers from 0 to $n - 1$, where n is the number of distinct values of the categorical variable.

(2) Data standardization: In this paper, we use Z-score standardization, which is the process of dividing the difference between each raw value and the mean by the standard deviation. This method can make the data distribution near the mean and at the same time

unify the scale of the data for better comparison and analysis. And it prevents the raw data from being too explosive. Its mathematical Formula (10) is as follows:

$$x' = \frac{x - mean}{std} \tag{10}$$

where mean represents the mean of the original data, and std represents the standard deviation of the original data.

(3) Data sampling: Due to the large amount of data in the dataset, this paper uses the K-means cluster sampling method to sample the dataset. The basic idea of K-means sampling is to use the K-means algorithm to cluster the dataset, and then randomly select a small number of samples from each cluster as representative samples, and the final representative sample set is the sampling result. The K-means can reduce the size of the dataset to reduce the amount of calculation and storage space and can remove some redundant information and noise.

(4) SMOTE method: There is a clear class imbalance in the collected dataset, which is the over-representation of normal samples in the dataset. This can lead to biased results of model training as well as reduced accuracy. In this paper, we propose the SMOTE method to deal with the class imbalance problem which is called Synthetic Minority Oversampling Technique. It creates new instances for a small number of classes to balance the dataset. The principle of SMOTE is based on the concept of KNN to synthesize high-quality instances and it does not simply replicate the instances as in random sampling which leads to overfitting. For each instance, X in the minority class, assuming that X_i is a random sample drawn from the k nearest neighbors of X; a new synthetic instance X_{new} can be expressed by Formula (11) as:

$$X_{new} = X + rand(0,1) \times (X_i - X), i = 1, 2, \ldots, k \tag{11}$$

where $rand(0,1)$ denotes a random number in the range (0,1), X_i is a randomly selected sample from the k nearest neighbors of X and X_{new} is a newly synthesized instance.

So far, we have carried out data preprocessing work as well as feature selection work based on IG-FCBF. Finally we selected 22-dimensional feature data among the massive dimensional data. In order to reduce the complexity of the intrusion detection model, this paper adopts the Pearson correlation coefficient to analyze the influencing factors of the flow of robotic arm system. The Pearson correlation coefficient is a measure of the degree of linear correlation between two feature variables [34]. The exact principle is obtained by calculating the ratio between the covariance and the standard deviation of the two variables. If the correlation coefficient is closer to the value 1, it means that the two variables are closer to being perfectly positively correlated. Conversely, the closer the correlation is to a perfect negative correlation. If the correlation coefficient is 0, then there is no correlation between the two variables. As shown in Figure 4, darker colors indicate stronger correlations. We put the 22-dimensional feature data into the heat map display. The last column in the figure is class features, i.e., labeled data. The numbers in the figure illustrate the degree of association of each feature with the labeled data. For example, the value of 0.63 in the 22 rows and 20 columns illustrates that the correlation value between the Init_Win_bytes_backward feature and the class labeling data is 0.63, which means that this feature has a strong correlation with the class labeling data. Therefore, we tend to focus more on these feature dimensions with a strong correlation to train the model. We can find that some core features have a high decision-making effect on the accuracy of classification. These data features mainly include core features such as Bwd Packet Length Min, Min Packet Length, PSH Flag Count, Down/Up Ratio, Init_Win_bytes_forward and Init_Win_bytes_backward. Therefore, we will highlight the importance scores of these features for the classification task. Also, other features can have an impact on the results, and we will use all the features in the figure as inputs to the model.

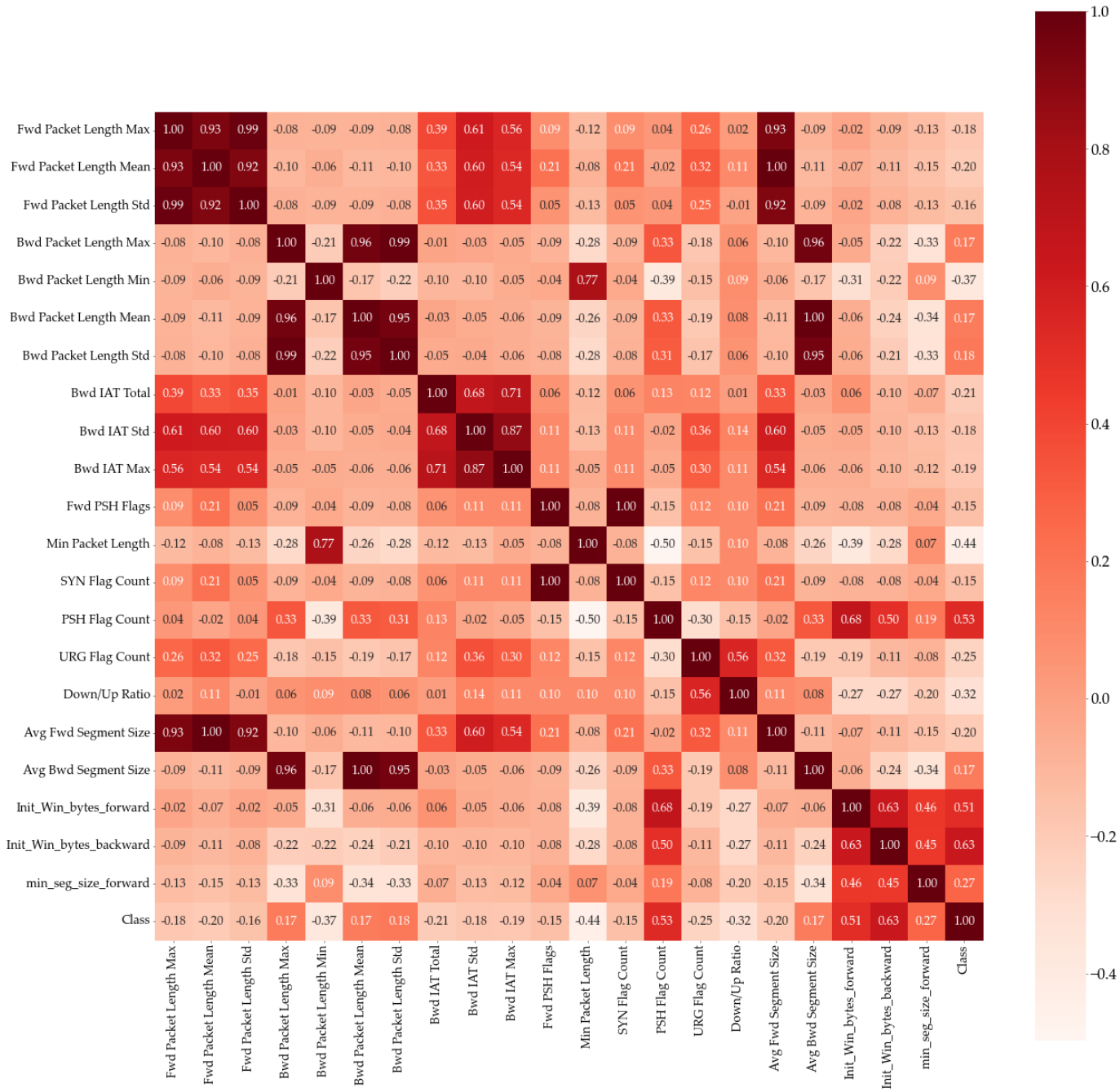

Figure 4. The heatmap of the correlation coefficients of input variables.

4.3. Performance Metrics

The purpose of this article is to detect various attacks and anomalies based on industrial control data. To measure the performance of the architecture, we will focus on the metrics of accuracy, recall, precision, F1-score, FNR and FPR.

The accuracy represents the proportion of the number of samples correctly predicted by the classifier out of all samples:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{12}$$

The recall rate indicates the proportion of the number of samples that the classifier correctly predicts as positive examples to the number of samples of all true examples:

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

The precision represents the ratio of the number of samples that the classifier correctly predicts as positive to the number of all samples that are predicted to be positive:

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

The F1 score represents the harmonic mean of precision and recall:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{15}$$

The FNR represents the ratio of the number of positive samples that the classifier incorrectly predicts as negative samples to the sum of the number of positive samples that are correctly predicted as positive samples and the number of positive samples that are incorrectly predicted as negative samples:

$$FNR = \frac{FN}{FN + TP} \tag{16}$$

The FPR represents the proportion of the number of negative samples that the classifier incorrectly predicts as positive samples to the sum of the number of negative samples that are incorrectly predicted as positive samples and the number of negative samples that correctly predict negative samples:

$$FPR = \frac{FP}{FP + TN} \tag{17}$$

where TP is the number of positive samples correctly predicted as positive samples, FP is the number of negative samples incorrectly predicted as positive samples, TN is the number of negative samples correctly predicted as negative samples and FN is the number of positive samples incorrectly predicted as negative samples.

4.4. Parameter Settings

In order to improve the effectiveness of the ETM-TBD model, this paper sets appropriate hyperparameters. In the training process, the experiment uses the Adam optimizer to optimize the model parameters. The specific parameters are shown in Table 2. Table 2 describes the detailed parameters of the Transformer model and BiGRU model. For example, the BatchSize is set to 512, the Dropout is set to 0.5, the Input Dimension is set to 32, the Embedding Layer Size is set to 32, the Number of Attention Heads is set to 4, the BiGRU Hidden Layer is set to 64 and the DNN Input Dimension is also 64.

Table 2. Experimental hyperparameters.

Parameter	Value
BatchSize	512
Dropout	0.5
EmbSize	32
InputDim	32
AttentionHeadNum	4
BiGRUHiddenSize	64
DNNInputSize	64

4.5. Related Experiments

This article uses the application of common machine learning algorithms on industrial control network traffic as comparative experiments, such as support vector machine (SVM), extra tree (ET), random forest (RF) and extreme gradient boosting (XGBoost).

The evaluation indicators selected in this paper are accuracy, recall, precision, F1-score, FNR and FPR mentioned in Section 4.3. This article first compares with the traditional machine learning methods, and also compares the stacked tree of four machine learning trees (stacking). It can be seen that the model in this paper has certain optimization compared with the traditional machine learning method. The strength of this paper's model on a range of metrics is shown more specifically in Table 3. Among them, FNR and FPR refer to the probability of false negatives and false positives in the training process. The lower the index, the better it reflects the model classification.

Table 3. Comparison of the results of the model in this paper and traditional machine learning methods.

Method	Accuracy	Precision	Recall	F1-score	FNR	FPR
XGBoost	0.9089	0.9156	0.9054	0.9104	0.0871	0.0217
RF	0.9173	0.9207	0.9103	0.9154	0.1053	0.0263
SVM	0.9099	0.9104	0.9015	0.9059	0.0907	0.0206
ET	0.9124	0.9083	0.9079	0.9081	0.0713	0.0178
Stacking	0.9507	0.9446	0.9472	0.9459	0.0408	0.0102
ETM-TBD	0.9724	0.9658	0.9672	0.9665	0.0263	0.0081

From the metrics in Table 3, it can be seen that the Acc, Pre, Rec and F1-score scores of the stacking model are 0.9507, 0.9446, 0.9472 and 0.9459, respectively. These metrics are all improved by 3.1–4.5% compared to the single machine learning model. Meanwhile, the stacking model has lower FNR and FPR scores of 0.0408 and 0.0102, respectively. The stacking model also performs a bit better in terms of false negatives and false positives. Thus, we can conclude that a single common machine learning algorithm tends to be less effective than multiple machine learning stacking integration models in handling intrusion detection tasks. It can be proved that the stacked ensemble model (ETM) is reasonable as the first layer of the model in this paper. The last column in Table 3 shows the performance results of the ETM-TBD model in this paper. From the results of the six metrics, the metrics results of the ETM-TBD model are improved by about 2.1% compared to the stacking model. The F1-score of the ETM-TBD model can reach 0.9665. Thus, it can be proved that the ETM-TBD model proposed in this paper is successful in the idea of hybrid machine learning and neural network models. As we all know, the radar chart can display the information of multiple variables at the same time, and it is convenient for comparison. Therefore, in order to show the effect of the algorithms more intuitively, we decided to use the radar chart to display the data, as shown in Figure 5.

At the same time, this paper also conducts comparative experiments with the following neural network models or machine learning integration models.

(1) ETM model: The ETM model is the first layer of the ETM-TBD model, which retains only the integrated machine learning module. We consider it as one of the comparison experiments for demonstrating the advantages of hybrid machine learning and neural network models.

(2) ETM-MBD model: Compared with the model in this paper, the ETM-MBD model replaces the Transformer module with the multi-head-attention mechanism to prove whether the skip connect in the Transformer module is more effective for classification.

(3) ETM-TD model: Compared with the model in this paper, the ETM-TD model lacks the BiGRU module. It can be used to explore the advantages of the BiGRU module in the case of retaining timing features and long-distance dependencies.

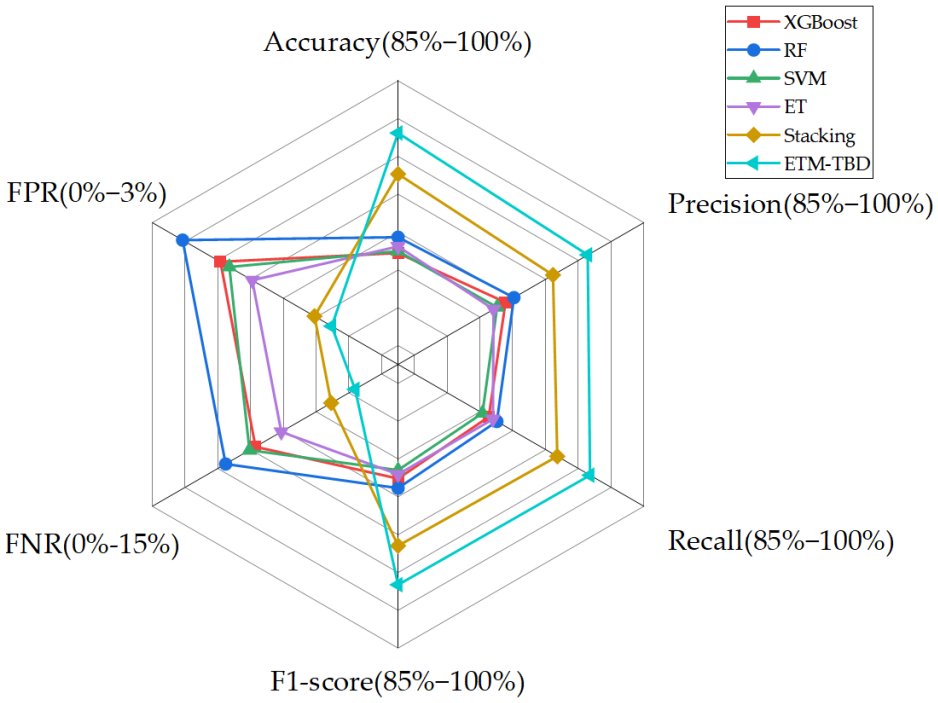

Figure 5. Comparison experiment radar chart.

As shown in Figure 6, different models also have different values for different labels. The labels in the figure are the string type labels of the original data encoded and processed. Label 1 corresponds to BENIGN, label 2 corresponds to Brute Force, label 3 corresponds to DDos, label 4 corresponds to Sql Injection and label 5 corresponds to XSS. It is not difficult to find that the model in this paper has a more significant effect than the compared models. The F1-score for each label of the model in this paper is 0.9607, 0.9710, 0.9667, 0.9580 and 0.9640, in that order. The single machine learning model ETM is obviously not as good as the results obtained by the model in this paper without the optimization of the neural network model. The average F1-score of the ETM-TBD model in this paper is 2.9% higher than that of the ETM model. The ETM-MBD model adds the multi-head-attention mechanism and lacks the Transformer module compared with the model in this paper. From the resulting graph, the average f1 score of the ETM-TBD model in this paper is 1.2% higher than that of the ETM-MBD model. It can be seen that the help of the Transformer for multiclassification tasks is very obvious, and it can be proved that the skip connect in the Transformer is important for extracting features and classification effects. Moreover, the ETM-TD model obtained from this paper's model after removing the Bidirectional Gated Recurrent Unit is compared with this paper's model. The average f1 score of the ETM-TBD model in this paper is 2.0% higher than that of the ETM-TD model. It can be proved that the preservation of time series features has more obvious advantages for multiclassification tasks. And the ETM-TD model often cannot optimize the results of the model according to the long-distance features of the context.

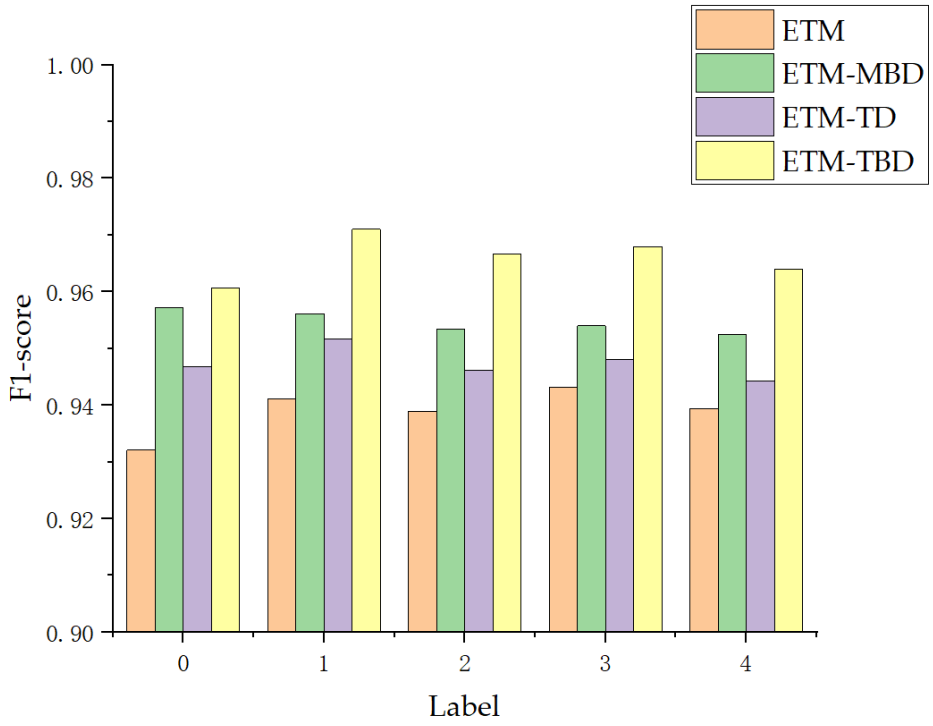

Figure 6. F1 scores of different models under different labels.

As shown in Figure 7, the vertical axis y_true is the true label of the test set and the horizontal axis y_pred is the predicted label of the test set predicted by the ETM-TBD model. We can see the specific prediction rate of each label from the confusion matrix; that is, we can see that there is a clear detection rate regardless of the specific attack type.

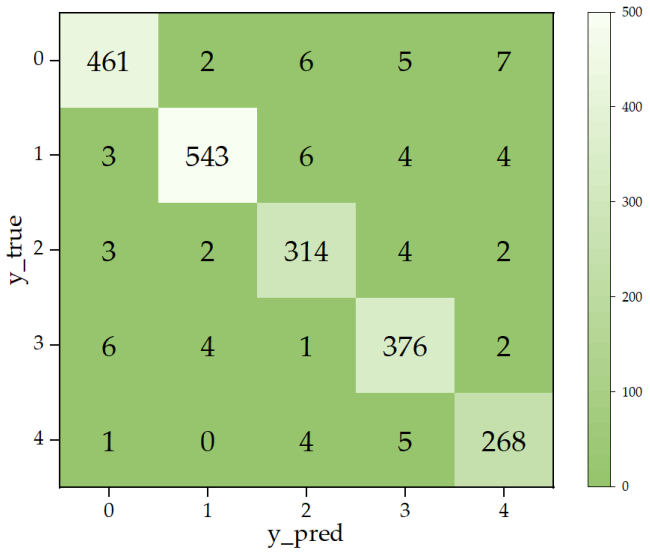

Figure 7. Confusion matrix of the ETM-TBD model.

Overall, the ETM-TBD model performs the best. Because compared to the ETM model, it has the ability to search for the connection between local and global features as well as the ability to search for the connection between features before and after the temporal sequence through the hybrid neural network model. It is obvious that the ETM-MBD model, compared to the model in this paper, lacks the skip connect feature of the Transformer model. The skip connect can result in the problem of gradient vanishing and the problem

of weight matrix degradation. Therefore the ETM-MBD model will be slightly worse in the results. The ETM-TD model is less considering the relationship of continuous features on the time series, which affects the accuracy of the model training. Therefore, through comparative experiments with the ETM model, ETM-MBD model and ETM-TD model, this paper proves that the combination of machine learning and neural network models has significant advantages in multiclassification tasks.

5. Conclusions

In order to ensure the security of industrial control networks and the accuracy of detecting network traffic, this paper proposes a hybrid machine learning and neural network model for a ETM-TBD intrusion detection model. First of all, the quality of the dataset can be significantly improved through preprocessing and feature engineering, which can be used for model training to make the results obtained after model training more accurate. The machine learning stacking tree model proposed in this paper can better complete the identification of intrusion attacks in industrial control networks. But the stacking machine learning tree model has similar shortcomings compared with traditional machine learning; that is, machine learning cannot adaptively learn features in data well and cannot be better adapted to complex datasets and tasks. In this regard, the integrated model of hybrid machine learning and neural network models is relatively better. It not only adapts itself to more complex tasks but also represents the data in a distributed manner, thus better handling high dimensional data. The Transformer module proposed in this paper can better handle classification tasks by establishing the connection between different features. And the BiGRU module can better capture the correlation and timing in the data by learning context information when dealing with high dimensions. The BiGRU module is very suitable for the situation, which can handle high dimensions. It is very suitable for the situation where the data volume of the industrial control network is relatively large. The DNN module adopted in the end supplements the first two modules to a certain extent. By weighing and nonlinearly transforming each layer of the model, higher-level features can be better learned, thereby achieving more accurate classification.

Through the performance evaluation of an industrial robotic arm control laboratory dataset, the ETM-TBD intrusion detection model proposed in this paper can effectively detect various known attack patterns. The detection accuracy rate of the dataset is 97.24%, and the average F1 score is 0.9665. Therefore, the feasibility of this model in the industrial control network is realized.

At the same time, the model in this paper has some drawbacks. For example, when we extract the more important features, the method based on IG-FCBF is not the best. Sometimes, some relatively unimportant features will be selected as well. In addition, we cannot handle unknown network traffic attacks well. As of now, if we are attacked by unknown types of attacks, we generally categorize them into the class of attack types that are most similar to them. Therefore, in our future work, we will focus on these two aspects, especially on the detection algorithm for open-set identification. We hope to mix the algorithms of open-set recognition into our intrusion detection system to face a variety of attacks in the industrial control field.

Author Contributions: Conceptualization, methodology, validation, writing, D.S.; Conceptualization, Data curation, Formal analysis, D.S., K.J. and J.L.; Supervision, funding acquisition and review, D.S., L.Z. and X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The authors are not allowed to publish the data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, F.; Kodituwakku, H.A.D.E.; Hines, J.W.; Tan, C.W.; Yu, W. Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4362–4369. [CrossRef]
2. Zhang, X.M.; Han, Q.L.; Ge, X.; Wang, Z. Networked control systems: A survey of trends and techniques. *IEEE/CAA J. Autom. Sin.* **2019**, *7*, 1–17. [CrossRef]
3. Farivar, F.; Haghighi, M.S.; Jolfaei, A.; Zomorodi-Moghadam, M. Artificial intelligence for detection, estimation, and compensation of malicious attacks in nonlinear cyber-physical systems and industrial IoT. *IEEE Trans. Ind. Inform.* **2019**, *16*, 2716–2725. [CrossRef]
4. Jiang, X.; Lora, M.; Chattopadhyay, S. An experimental analysis of security vulnerabilities in industrial IoT devices. *ACM Trans. Internet Technol. (TOIT)* **2020**, *20*, 1–24. [CrossRef]
5. Shafiq, M.; Tian, Z.; Bashir, A.K.; Chen, Y. IoT malicious traffic identification using wrapper-based feature selection mechanisms. *Comput. Secur.* **2020**, *94*, 101863. [CrossRef]
6. Fu, C.; Li, Q.; Shen, M.; Li, W.; Zhang, Y. Realtime robust malicious traffic detection via frequency domain analysis. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, 15–19 November 2021; pp. 3431–3446.
7. Wang, Z.; Fok, K.W.; Thing, V.L.L. Machine learning for encrypted malicious traffic detection: Approaches, datasets and comparative study. *Comput. Secur.* **2022**, *113*, 102542. [CrossRef]
8. Alshammari, A.; Aldribi, A. Apply machine learning techniques to detect malicious network traffic in cloud computing. *J. Big Data* **2021**, *8*, 90. [CrossRef]
9. Li, Z.; Liu, F.; Yang, W.; Li, Y.; Li, S. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [CrossRef]
10. Kanna, P.R.; Santhi, P. Hybrid Intrusion Detection using MapReduce based Black Widow Optimized Convolutional Long Short-Term Memory Neural Networks. *Expert Syst. Appl.* **2022**, *194*, 116545. [CrossRef]
11. Foley, J.; Moradpoor, N.; Ochenyi, H. Employing a Machine Learning Approach to Detect Combined Internet of Things Attacks against Two Objective Functions Using a Novel Dataset. *Secur. Commun. Netw.* **2020**, *2020*, 2804291. [CrossRef]
12. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
13. Peng, W.; Kong, X.; Peng, G.; Li, X.; Wang, Z. Network intrusion detection based on deep learning. In Proceedings of the 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), Haikou, China, 5–9 July 2019.
14. Othman, S.M.; Ba-Alwi, F.M.; Alsohybe, N.T.; Al-Hashida, A.Y. Intrusion detection model using machine learning algorithm on Big Data environment. *J. Big Data* **2018**, *5*, 34. [CrossRef]
15. Lv, H.; Du, Y.; Zhou, X.; Ni, W.; Ma, X. A Data Enhancement Algorithm for DDoS Attacks Using IoT. *Sensors* **2023**, *23*, 7496. [CrossRef] [PubMed]
16. Chen, C.; Guo, X.; Zhang, W.; Zhao, Y.; Wang, B.; Ma, B.; Wei, D. Application of GA-WELM Model Based on Stratified Cross-Validation in Intrusion Detection. *Symmetry* **2023**, *15*, 1719. [CrossRef]
17. Kilichev, D.; Kim, W. Hyperparameter Optimization for 1D-CNN-Based Network Intrusion Detection Using GA and PSO. *Mathematics* **2023**, *11*, 3724. [CrossRef]
18. Yang, H.; Xu, J.; Xiao, Y.; Hu, L. SPE-ACGAN: A Resampling Approach for Class Imbalance Problem in Network Intrusion Detection Systems. *Electronics* **2023**, *12*, 3323. [CrossRef]
19. Saba, T.; Rehman, A.; Sadad, T.; Shahid, W.; Shah, S.A. Anomaly-based intrusion detection system for IoT networks through deep learning model. *Comput. Electr. Eng.* **2022**, *99*, 107810. [CrossRef]
20. Dablain, D.; Krawczyk, B.; Chawla, N.V. Deep SMOTE: Fusing deep learning and SMOTE for imbalanced data. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 6390–6404. [CrossRef]
21. Gao, Z.; Li, Z.; Luo, J.; Li, Y.; Li, S. Short text aspect-based sentiment analysis based on CNN+BiGRU. *Appl. Sci.* **2022**, *12*, 2707. [CrossRef]
22. Bhat, P.; Dutta, K. A multi-tiered feature selection model for android malware detection based on feature discrimination and information gain. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 9464–9477. [CrossRef]
23. Chen, B.; Chen, X.; Chen, F.; Gao, R.X.; Lei, Y. Integrated early fault diagnosis method based on direct fast iterative filtering decomposition and effective weighted sparseness kurtosis to rolling bearings. *Mech. Syst. Signal Process.* **2022**, *171*, 108897. [CrossRef]
24. Injadat, M.N.; Moubayed, A.; Shami, A. Detecting botnet attacks in IoT environments: An optimized machine learning approach. In Proceedings of the 2020 32nd International Conference on Microelectronics (ICM), Aqaba, Jordan, 14–17 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–4.
25. Bansal, M.; Goyal, A.; Choudhary, A. A comparative analysis of K-Nearest Neighbour, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning. *Decis. Anal. J.* **2022**, *3*, 100071. [CrossRef]
26. Balyan, A.K.; Ahuja, S.; Lilhore, U.K.; Singh, S.P.; Kumar, A. A hybrid intrusion detection model using EGA-PSO and improved random forest method. *Sensors* **2022**, *22*, 5986. [CrossRef]

27. Kavzoglu, T.; Teke, A. Predictive performances of ensemble machine learning algorithms in landslide susceptibility mapping using random forest, extreme gradient boosting (XGBoost) and natural gradient boosting (NGBoost). *Arab. J. Sci. Eng.* **2022**, *47*, 7367–7385. [CrossRef]
28. Chen, L.; You, Z.; Zhang, N.; Zhang, Y.; Huang, H. Utrad: Anomaly detection and localization with U-Transformer. *Neural Netw.* **2022**, *147*, 53–62. [CrossRef] [PubMed]
29. Zhang, W.; Li, H.; Tang, L.; Wang, Z.; Li, J. Displacement prediction of Jiuxianping landslide using gated recurrent unit (GRU) networks. *Acta Geotech.* **2022**, *17*, 1367–1382. [CrossRef]
30. AlHaddad, U.; Basuhail, A.; Khemakhem, M.; Eassa, F.E.; Jambi, K. Ensemble Model Based on Hybrid Deep Learning for Intrusion Detection in Smart Grid Networks. *Sensors* **2023**, *23*, 7464. [CrossRef]
31. Xiang, G.; Shi, C.; Zhang, Y. An APT Event Extraction Method Based on BERT-BiGRU-CRF for APT Attack Detection. *Electronics* **2023**, *12*, 3349. [CrossRef]
32. Yang, T.; Li, G.; Wang, T.; Yuan, S.; Yang, X.; Yu, X.; Han, Q. A Novel 1D-Convolutional Spatial-Time Fusion Strategy for Data-Driven Fault Diagnosis of Aero-Hydraulic Pipeline Systems. *Mathematics* **2023**, *11*, 3113. [CrossRef]
33. Cao, W.; Zhang, Y.; Gao, J.; Cheng, A.; Cheng, K.; Cheng, J. Pkd: General distillation framework for object detectors via pearson correlation coefficient. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 15394–15406.
34. Baak, M.; Koopman, R.; Snoek, H.; Klous, S. A new correlation coefficient between categorical, ordinal and interval variables with Pearson characteristics. *Comput. Stat. Data Anal.* **2020**, *152*, 107043. [CrossRef]