Journal on
Internet of Things

Tech Science Press

DOI: 10.32604/jiot.2022.037416
Article

Intrusion Detection Method Based on Active Incremental Learning in Industrial Internet of Things Environment

Zeyong Sun[1], Guo Ran[2] and Zilong Jin[1,3,*]

[1]School of Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China
[2]Cyberspace Institute Advanced Technology, Guangzhou University, Guangzhou, 510006, China
[3]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing, 210044, China
*Corresponding Author: Zilong Jin. Email: zljin@nuist.edu.cn
Received: 03 November 2022; Accepted: 06 December 2022

Abstract: Intrusion detection is a hot field in the direction of network security. Classical intrusion detection systems are usually based on supervised machine learning models. These offline-trained models usually have better performance in the initial stages of system construction. However, due to the diversity and rapid development of intrusion techniques, the trained models are often difficult to detect new attacks. In addition, very little noisy data in the training process often has a considerable impact on the performance of the intrusion detection system. This paper proposes an intrusion detection system based on active incremental learning with the adaptive capability to solve these problems. IDS consists of two modules, namely the improved incremental stacking ensemble learning detection method called Multi-Stacking model and the active learning query module. The stacking model can cope well with concept drift due to the diversity and generalization selection of its base classifiers, but the accuracy does not meet the requirements. The Multi-Stacking model improves the accuracy of the model by adding a voting layer on the basis of the original stacking. The active learning query module improves the detection of known attacks through the committee algorithm, and the improved KNN algorithm can better help detect unknown attacks. We have tested the latest industrial IoT dataset with satisfactory results.

Keywords: Intrusion detection; IDS; active incremental learning; stacking ensemble learning; unknown attacks

1 Introduction

Under the background of intelligence, the industrial Internet of Things (IIot) is transforming industrial production environments by connecting sensors, brakes, and other devices to enable them to operate more efficiently and reliably. The rapid development of artificial intelligence technology has brought more new possibilities for the intelligence of Industrial Control Systems (ICS) [1]. Especially the development of various technologies such as sensors, 5g networks, cloud computing,

and edge computing, there will be more and more machines, equipment, and sensors connected to IIot gateways and local routers in the future, which may contain critical data in the industrial production process. which could be extremely costly for factories if stolen or tampered with by lawless. Network security is all about preventing network attacks. Intrusion detection systems (IDS) are currently one of the most effective and commonly used methods. Fig. 1 depicts all layers in the IIoT, including the perception layer, the networking layer, the application layer, and the solution based on the IDS network architecture. The process includes data collection, data preprocessing, concrete implementation, training, and validation. The security database stores the collected signatures of various attacks, and the data is preprocessed and sent to IDS for detection. IDS and Next Generation Firewall (NGFW) work together to maintain the security of IIoT [2].

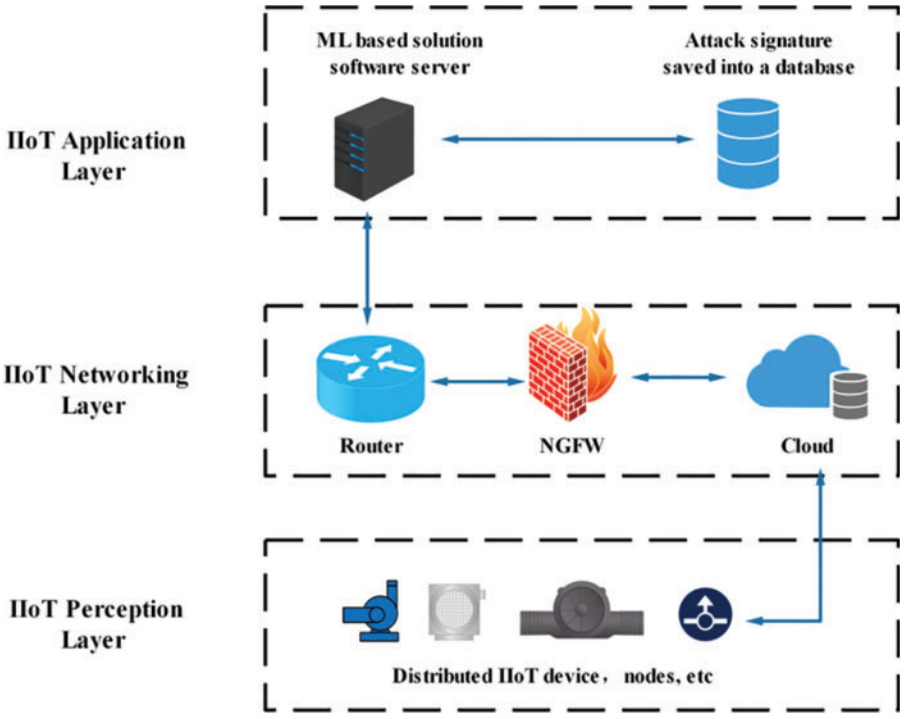

Figure 1: Solution based on IDS network architecture in IIOT environment

Network traffic as a stream of data comes from one or more data sources [3], with high frequency and large data volume, and many data will not be saved. It is important to note that with the advancement of artificial intelligence, a variety of attack generation tools came into being. Network criminals will often skillfully use these tools to generate various attacks, and some of these attacks can skillfully evade detection [4].

IDS based on batch learning has poor adaptability, because it cannot learn new attacks unless it is retrained, which causes a waste of time [5]. For huge data streams, it is not possible to load them into memory at once for training for batch learning. Incremental learning-based techniques are a good solution to these problems. During incremental learning, the models are updated accordingly to changes in the data [6], and there is no need to store all the training data at once.

Incremental learning also faces some challenges, the biggest of which is the problem of conceptual drift [7]. In the training prediction process, we often assume that the data has independent identical

distribution properties so that the machine learning algorithm can learn the distribution properties of the data in the training set. However, the joint probability distribution of real data streams tends to change, which may affect the classifier's performance [8,9].

To address these issues, this paper proposes a hybrid intrusion detection framework based on incremental ensemble learning and active learning. First, we propose an improved stacking ensemble learning detection method called Multi-Stacking. Stacking usually has two layers, the basic layer is composed of multiple heterogeneous classifiers, and the generalization layer is usually composed of one meta classifier. The diversity of base classifiers and generalization options in stacking can help solve concept drift. We further improve the model accuracy and generalization ability by adding a voting layer between the base layer and the generalization layer, which consists of multiple voting policies. In active learning, we first cull the noisy and hard-to-classify data by a committee algorithm and then continue to detect unknown attacks in the normal traffic obtained from the Multi-Stacking classification by an improved KNN algorithm. We store the culled data in markers and retrain the three-layer stacking model to improve its performance and ability to detect unknown attacks.

This paper is organized as follows: In Section 2, related work will be discussed. And our proposed method is introduced elaborately in Section 3. Section 4 gives the experimental results of the proposed scheme. Finally, a conclusion is given in Section 5.

2 Related Works

Traditional machine learning methods, despite their limitations, are simpler and more convenient in terms of optimizing model changes and adjusting hyperparameters while having better learning and detection capabilities. Anand Sukumar et al. [10] proposed an intrusion detection method, IGKM, which uses a genetic algorithm (GA) to obtain the optimal k value of the k-means algorithm. Kanimozhi et al. [11] proposed a logistic regression (LR) method based on the opposite truncated fuzzy mean, which uses logistic regression for feature selection, greatly improving the training efficiency, but it is easy to produce overfitting when the number of tuples is insufficient.

The popular deep learning method has strong detection ability and is the first choice for researchers to design intrusion detection systems. Amjad et al. [12] proposed a deep learning architecture combining auto coder and LSTM for detecting known and unknown attacks in networks. Singla et al. [13] proposed adversarial domain adaptation for intrusion detection in the absence of labeled data. Experiments show that this method can maintain good accuracy in the absence of labeled training data.

Ensemble learning models are also commonly used in intrusion detection by combining multiple weak classifiers into a single strong classifier. Miah et al. [14] use cluster-based undersampling and random forest classifier to classify a few types of network attacks/intrusions. This method is a multi-level classification method, which can deal with highly unbalanced big data to correctly identify minority/rare class intrusions. Zheng et al. [15] proposed a stacking-based intrusion detection method that greatly exceeds the detection capability of a single classifier by combining SVM, BPNN, K-Means, and XGBoost into a set of base classifiers.

Incremental learning is to learn samples one by one in chronological order to partially update the learning model. The Hoeffding tree (HT) is a type of decision tree (DT) that uses the Hoeffding bound to incrementally adapt to data streams [16]. Compared to a DT that chooses the best split, the HT uses the Hoeffding bound to calculate the number of necessary samples to select the split node. Thus, the HT can update its node to adapt to newly incoming samples. However, the HT does not

have mechanisms to address specific types of drift. The Extremely Fast Decision Tree (EFDT) [17], also named Hoeffding Anytime Tree (HATT), is an improved version of the HT that splits nodes as soon as it reaches the confidence level instead of detecting the best split in the HT. This splitting strategy makes the EFDT adapt to concept drifts more accurately than the HT, but its performance still needs improvement.

Due to the variable attack behavior, traditional IDS based on batch learning algorithms cannot be dynamically adjusted in the face of changes in the network environment and often require retraining of the model, which poses many problems and issues, for which incremental learning is a good solution. Xu et al. [18] proposed an incremental KNN-SVM method where the model is able to maintain a short prediction time during the update process. Wang et al. [19] proposed a small-sample class-incremental learning method to facilitate the class-incremental learning strategy of DNN through a meta-learning method, which is suitable for cases where there are few new class samples.

In the above research scheme, Few intrusion detection methods consider adapting a system to an attack that actually occurs in the environment. Some attack types, even if learned, may hardly occur in the environment in the future. At this point, our intrusion detection system does not need to learn. The IDS based on active incremental learning in this paper can quickly adapt and learn the types of attacks that exist in the current environment.

3 Proposed Methods

This section details the proposed intrusion detection method based on active incremental integrated learning. IDS mainly consists of two modules. Multi stacking method module and active learning module. Active learning is responsible for guiding and optimizing the multi stacking method module in the detection process. Fig. 2 describes the specific process.

3.1 Multi-Stacking Method

Stacking ensemble learning can effectively address the problem of concept drift due to the diversity of its base classifiers and generalization selection. We improve the input value of the meta classifier by adding combined voting to the output of the traditional base classifier to increase the accuracy of the model.

For the industrial IoT security dataset $D = \{(x_i, y_i)\}_{i=1}^{m}$, where $x_i \in R^d$ is the detection feature of the i-th instance. $y_i = Y \in \{0, 1\}$ is the corresponding label. For industrial IoT real data flow $S = \{(x'_j, y'_j)\}_{j=1}^{\infty}$, where $x'_j \in R^d$ is the detection feature of the j-th instance of the data stream. $y'_j = Y' \in \{0, 1, UA\}$, UA stands for unknown attack. We first use dataset D to train Multi-Stacking, and then use Multi-Stacking to detect known attacks in data stream S.

In our proposed Multi-Stacking method, the first layer is a base layer composed of a set of base learners, denoted as $BC_t (t = 1, 2, \ldots, M)$. Passive Aggregate Classifier (PAC), Gaussian Naïve Bayes (GNB), Stochastic Gradient Descent (SGD), Perceptron (Linear Model) and Multi Layer Perceptron (MLP) are selected as the base classifiers in this paper. For the training dataset, in order to increase the diversity of the base learner training data and thus improve the generalization ability of the intrusion detection model, we use the self-sampling method (Bootstrap) to generate multiple training subsets of equal size to the original dataset. The classification bias is reduced by k-fold cross validation, and k is taken as 5.

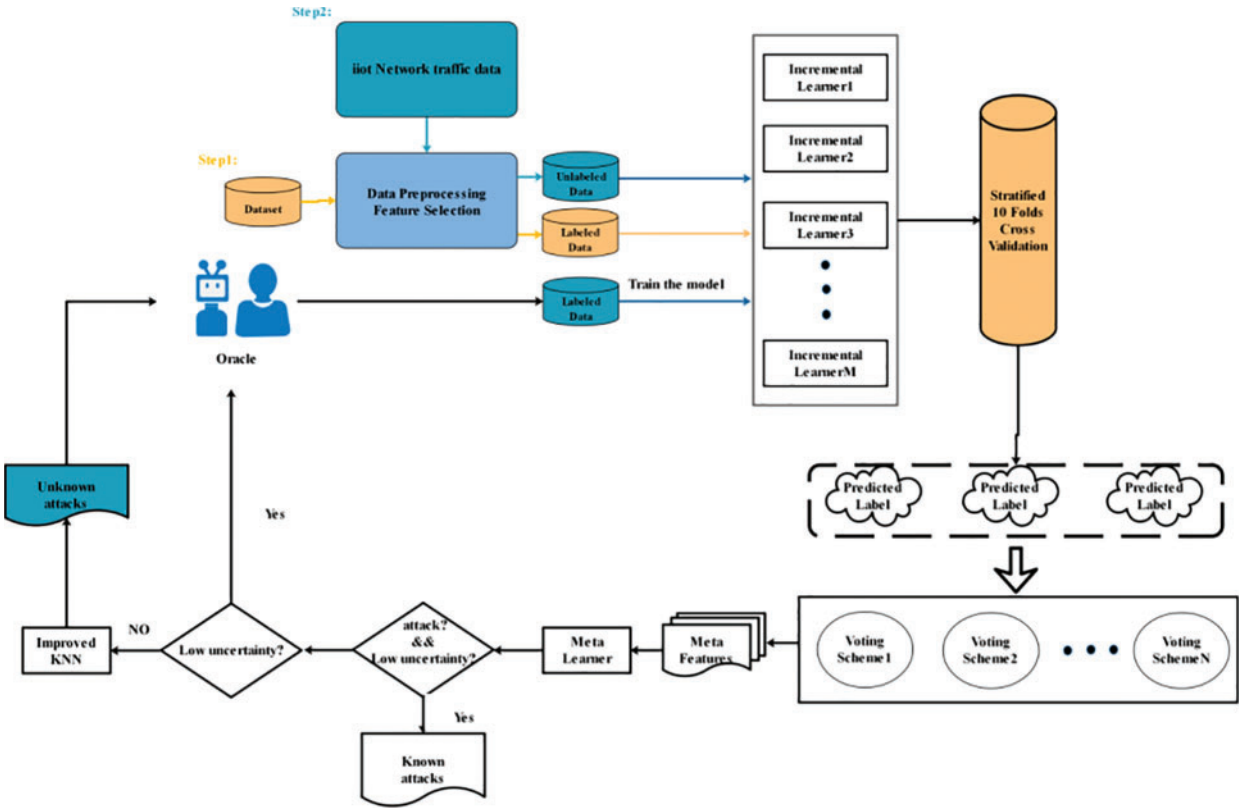

Figure 2: IDS detection framework based on active incremental learning

The industrial IoT attack dataset was divided into five parts, where $T_{train} = \{T_1, T_2, T_3, T_4\}$ was used as the training set, $T_{test} = \{T_5\}$ as the test set, and then T_1, T_2, T_3, and T_4 were used as the test set respectively, and the output of each base classifier was calculated five times to obtain the final result. As the base layer consists of multiple classifiers, the output will have a certain amount of variance and bias, and the model will inevitably have overfitting or underfitting. Therefore, a voting layer is added between the basic layer and the generalization layer, and the variance is reduced by comprehensive voting on the results of the base layer so as to optimize the input of the generalization layer and ultimately improve the accuracy of the model. The selected voting methods are plural voting, confidence-based voting and weighted majority voting, where plural voting is based on the number of base classifiers, and confidence-based voting, and weighted majority voting make decisions based on class probability. In the generalization layer, logical regression is selected as the meta classifier, and we use the output of the base layer optimized by combined voting and the top three features as the input of the logical regression to obtain the final predictive value.

Algorithm 1: Multi-Stacking algorithm

Input: Dataset $D = \{(x_i, y_i)\}_{i=1}^{m}$
Data stream S
Time instance T
Base-Learner: BC_1, BC_2, BC_3

(Continued)

Algorithm 1: Continued

Meta-learner: LR

Output: Network attack and normal traffic in real industrial Internet of things

1 Randomly Split D into five partitions of equal size to get $T_{train} = \{T_1, T_2, T_3, T_4\}$, $T_{test} = \{T_5\}$
2 for $t = 1,2,...,M$ do
3 for $n = 1,2,3,4,5$ do
4 $h_t = BC_t(T_t)$
//Training base classifier
5 if $n > 1$ Change T_{n-1} to be T_{test}
6 End
7 Plural Voting:
8 for $i = 1,2,...,m$ do
9 Plural Voting Label: $PVL = mode(h1(x_i), h2(x_i), h3(x_i))$
//*Find the mode of the output result*
10 End
11 Confidence-Based Voting:
12 For all the instances, store the class probabilities of all the classifiers for all the classes into an array
13 for $i = 1,2,...,m$ do
14 Max_Prob ← Max(class probabilities of all the classifiers for all the classes)
15 Confidence-Based Voting: CVL_i —> the class label corresponding to Max_Prob
End
16 Weighted Majority Voting:
17 P_t^k–> Accuracy of classifier t in the k-th round
18 $w_t = \dfrac{\frac{1}{K}\sum_{k=1}^{K} P_t^k}{\sum_{t=1}^{T} P_t}$
//Calculate classifier weights
19 Weighted Majority Voting:
$WMVL_i$—> Class label with the highest weighted average
20 F_i —> Selected top three features
21 $VL_i = \{PVL_i, CVL_i, WMVL_i\}$
22 $EL_i = F_i \cup VL_i$
//The input result is the set of the first three selected features and the output value of the voting layer
23 $h' = LR(EL_i)$
// Training meta learner
24 Detect attacks in data stream S with the trained ensemble learning model

3.2 Active Learning of Data Query Strategies

3.2.1 Committee Query Algorithm

The base classifier in Multi-Stacking, as a committee member in the committee query algorithm, is defined as $BC = \{BC_1, BC_2, \ldots, BC_i\}$, where BC_i represents the i-th base classifier model. If the predictions of a sample differ significantly between members in the committee, the current sample is considered to have a large disagreement and such data needs to be labeled. There are two types of samples with large disagreement, one is data with label noise, even a small amount of label noise in the

data can have a large impact on the final prediction result. The other is data that is difficult to classify, which can also be labeled to improve the model's performance. The KL divergence is used to measure the inconsistency of committee members. The calculation formula is as follows:

$$x^*_{KL} = argmax_x \frac{1}{C} \sum_{i=1}^{C} D(P_{BC_i} || P_{BC}) \tag{1}$$

where $D(P_{BC_i}||P_{BC}) = \sum_i P(y_i|x; BC_i) log \frac{P(y_i|x; BC_i)}{P(y_i|x; BC)}$, BC represents the whole committee, and C represents the number of members.

3.2.2 Improved KNN Algorithm

We improve the KNN algorithm to make it efficient for detecting unknown attacks. The proposed method censors the unknown attacks from the normal traffic obtained from the Multi-Stacking model classification by excluding the normal traffic by relative distance. In the industrial IoT dataset $D = \{(x_i, y_i)\}_{i=1}^{m}, y_i = Y \in \{0, 1\}$. For each known class, we draw a subset Q_i of data into Q by not putting back n instances, and the process is carried out c times, so Q can be expressed as $Q = \cup_{i=1}^{n} Q_i$. A hypersphere is defined according to the instances in each data subset Q_i, with the center of the sphere o. The radius is taken to be the mean of the L nearest neighbor points, expressed as follows:

$$r(o) = \frac{\sum_{l=1,}^{L} ||x_l - o||}{L} \tag{2}$$

Therefore, the hypersphere is defined as:

$$h(o, r(o)) = \{x: ||x - o|| \leq r(o)\} \tag{3}$$

For all hyperspheres in D_i, define H_i as the set of hyperspheres in Q_i:

$$H_i = \cup_{o \in D_i} h(o, r(o)) \tag{4}$$

Then we determine the category of the traffic by calculating the distance between the analysis traffic and the hypersphere. The discriminant formula is as follows:

$$f'_i(x) = \begin{cases} argmin_i ||x_l - o||, \frac{||x_l - o||}{r(o)} \leq t \\ UA, otherwise \end{cases} \tag{5}$$

where t is the set threshold, i $\in \{0, 1\}$.

Next, the discriminant values of the samples were voted on by Plural voting and the final flow class results were as follows:

$$f(x) = argmax_k \sum_{i=1}^{c} \rho(f'_i(x), c_k) \tag{6}$$

where $c_k \in \{0, 1, UA\}$, UA stands for unknown attack, $\rho(f'_i(x), c_k) = 1$, if $f'_i(x) = c_k$, otherwise $\rho(\mathrm{f'_i(x), c_k}) = 0$.

After the classified unknown attacks are labeled by active learning, the KNN detector can be further updated. The specific algorithm is shown in Algorithm 2.

Algorithm 2: Improved KNN algorithm

Input: Dataset $D = \{(x_i, y_i)\}_{i=1}^{m}$
Network data stream S
Output: Network attack and normal traffic in real industrial Internet of things
1 initialize $H = \cup_{i=1}^{b} H_i$, $H_i = \varnothing$
2 for $i = 1,2,...,c$ do
3 Sample for D to get Q_i containing all known classes
4 build a hypersphere ensemble Hi with Q_i using (2) and (3)
5 $H = H \cup H_i$
6 Calculate and analyze the distance between the Network traffic data and each hypersphere to obtain the classification output $f_i'(x)$
7 End
8 obtain the final result by majority voting using (6)
9 If the labeled data Q_{UA} of the new unknown attack class is obtained
10 Divide Q_{UA} into c parts: $Q_{UA} = \{Q_1^{UA}, Q_2^{UA},...,Q_c^{UA}\}$
11 for $i = 1,2,...,c$ do
12 $Q_i = Q_i \cup Q_i^{UA}$
13 do step 4–5
14 End

4 Experimental Analyses

In this section, we perform a series of experiments to evaluate the proposed IDS.

4.1 Datasets

We choose Edge-IIoTset, a comprehensive reality network security data set of the latest industrial IoT applications [20], in which the data is generated by a variety of IIoT devices, such as soil moisture sensors, pH sensors, flame sensors, and digital sensors used to sense temperature and humidity. The dataset was generated from 21 November 2021 to 10 January 2022 and overall contains a total of 20952,648 data and 1,176 features, 61 of which have high relevance. The dataset collects a total of fourteen attacks related to IoT and IIoT connectivity protocols, which are grouped into five threats, including DoS/DDoS attacks, information gathering, man-in-the-middle attacks, injection attacks, and malware attacks [21].

4.2 Evaluation Measure

The Area Under the ROC Curve (AUC) is plotted to validate the effectiveness of the proposed active incremental learning method based on Multi-Stacking, with the value of AUC ranging from 0 to 1, the closer to 1, the better the model validity. The Area Under the ROC Curve (ROC) is plotted by the false positive rate (TPR) and the false positive rate (FPR) for a given model. TPR and FPR are defined as follows:

$$TPR = \frac{TP}{TP + FN} \tag{7}$$

$$FPR = \frac{FP}{FP + FN} \tag{8}$$

Each component in the above formula is defined as follows:

- TP: Indicates traffic that is correctly classified as attack.
- TN: Indicates traffic that is correctly classified as normal.
- FP: Indicates traffic that is incorrectly as attack.
- FN: Indicates traffic that is incorrectly as normal.

4.3 Influences of Parameters

In this section, we study the influences of three parameters: the number of labeled data n, the number of nearest neighbors L, and the threshold t. We choose the appropriate parameters by fixing other parameters and evaluating the performance of the model for different parameters at different values.

Formula (1) expresses that if the threshold t is set too small, more normal traffic will be misjudged as attack traffic, which will add many unnecessary labels. If the value of t is too large, some attack traffic will be missed. Therefore, our primary goal is to ensure a higher TPR and then select a smaller t value as far as possible. Fig. 3a shows the effect of the t value on the TPR. When t is 0.98, it can not only achieve a higher TPR but also ensure that the value of t is as tiny as possible. Fig. 3b shows the influence of the number of nearest neighbors L on the TPR. When L is less than 7, the range of change is relatively large, and when L is greater than 7, it tends to be stable. The calculation cost can be reduced by adequately using a smaller number of nearest neighbors. Therefore, we choose seven as the number of nearest neighbors.

Fig. 3c shows the AUC of the active incremental learning method on the test set with the different number of labels n. It can be seen that the detection performance is low at the beginning because many attacks in the test set belong to unknown attacks and are difficult to detect. Still, over time, the AUC has significantly improved, indicating that active incremental learning has good adaptability in actual detection. Using the number of labels for n of 100 and 500, equivalent to 2% and 10% of the training set, and using the complete data for AUC comparisons at different time instances. The higher the labeling rate is, the better the effect will be. When n is 500, the final AUC can reach the AUC obtained by using the complete training set. Therefore, the same effect of all labels can be achieved by sampling labels, and the label cost can also be significantly reduced.

4.4 Results and Discussion

Fig. 4 mainly shows two groups of active incremental learning methods. The detection models are the Multi-Stacking method and the Stacking method of this paper and the change in AUC at different time instances for the removal of the committee algorithm. The addition of the combined voting layer brings a considerable improvement in the detection of the stacking model. There are few new attacks in 40 to 50 time instances. We can see that the detection of known attacks by the model can be improved by the committee algorithm by re-labeling noisy data and by labeling indistinguishable samples. The performance in 50 to 60 even slightly exceeds that of the model trained with all data, because some noise data in the data set are detected.

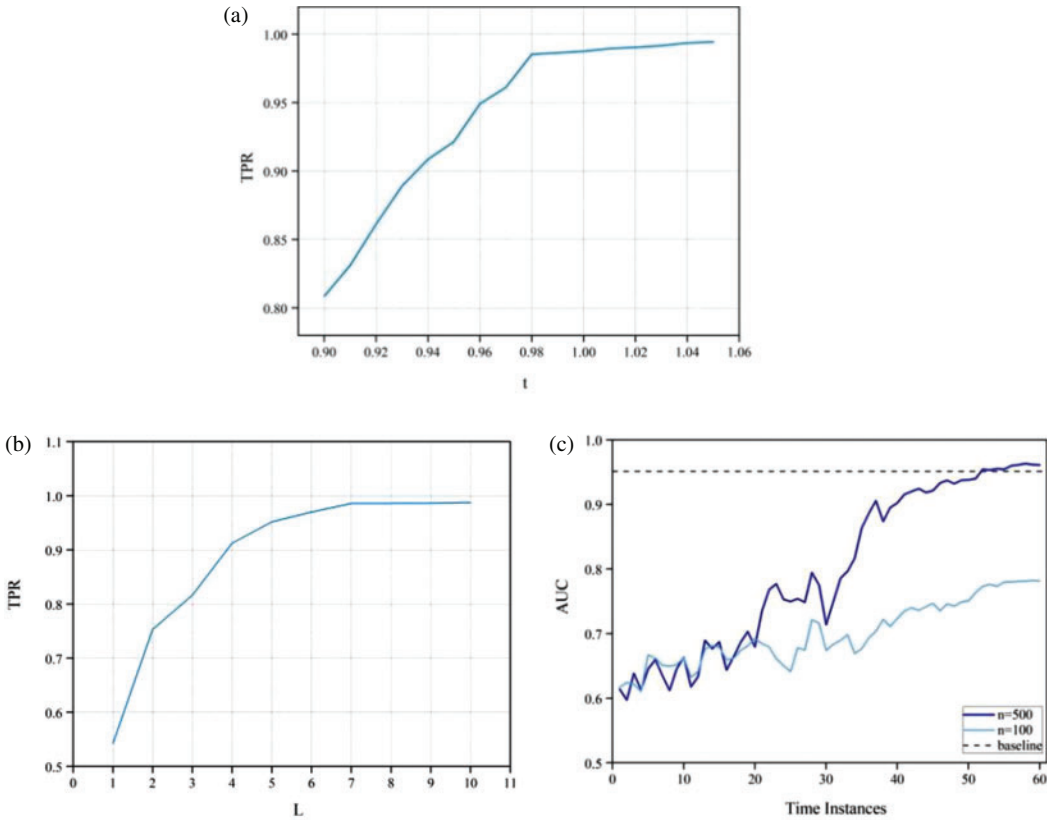

Figure 3: Influences of parameters: (a) Influence of threshold t. (b) Influence of number of nearest neighbors L. (c) Influences of number of labeled data n

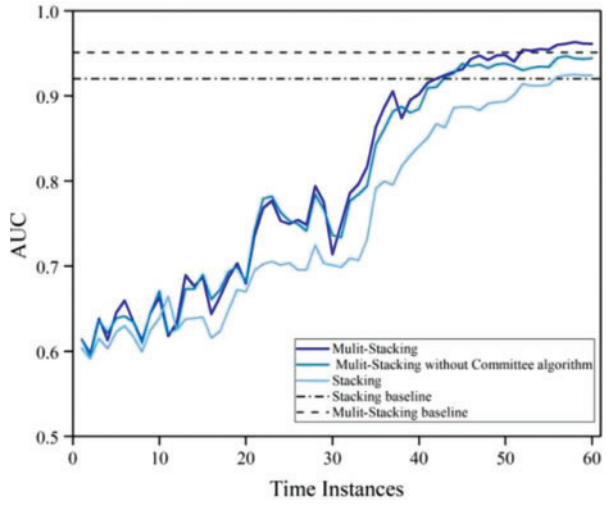

Figure 4: Performance of the Multi-Stacking, Stacking, and Multi-Stacking without Committee algorithm under different time instances

In Fig. 5, Multi-Stacking is replaced with other machine learning methods, and conducted several experiments to compare the performance of the final model in the test set for different numbers of time instances. We can see that the Multi-Stacking model is indeed a significant improvement in detection and outperforms several other standard machine learning methods selected.

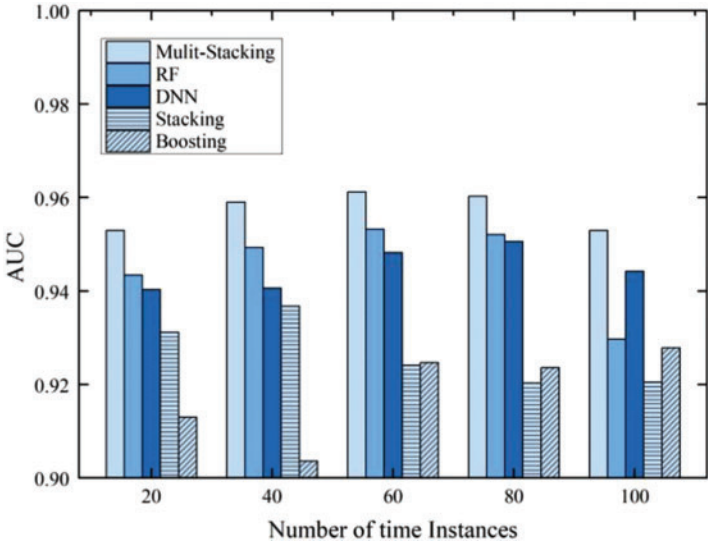

Figure 5: Performance of the final model under different number of time instances

5 Conclusion and Future Work

In this paper, An intrusion detection framework was proposed based on active ensemble learning, using an improved KNN and committee algorithm as the active learning query algorithm. The improved KNN is responsible for helping to find potential unknown attacks, and the committee algorithm is mainly responsible for finding some noisy data in the datasets and in the crowd sourcing process. Active learning is responsible for "teaching" the ensemble learning model to identify more attack types and improve the detection accuracy of known attacks, so the proposed IDS is adaptive. In order to fit this incremental learning model, the existing stacking model was chosen and improved, which copes well with conceptual drift in a realistic detection environment. By adding a voting layer between the base layer and the generalization layer of the stacking model, the Multi-Stacking model has higher accuracy and also copes well with concept drift.

In the future, we intend to further improve the Multi-Stacking method. The limitation of Multi-Stacking is that the added voting layer requires additional data processing and a small amount of extra execution time. Therefore, the following work considers the parallel processing of the integration layer and the voting layer to make up for this shortcoming. Feature selection is also one of the improvement directions we consider. Improving the existing feature selection method can help us perform real-time detection more efficiently and improve detection accuracy.

Funding Statement: This work was sponsored by the National Natural Science Foundation of China under Grants 62271264, 61972207, and 42175194, and the Project through the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institution.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Rodriguez-Diaz, "Industry 4.0: The industrial Internet of Things," *Computing Reviews*, vol. 58, no. 4, pp. 228–229, 2017.

[2] F. F. Alruwaili, "Intrusion detection and prevention in industrial IoT: A technological survey," in *Proc. of Int. Conf. on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Mauritius, pp. 1–5, 2021.

[3] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. of ICISSp*, Hyderabad, India, pp. 1–5, 2018.

[4] N. Moustafa, J. Slay and G. Creech, "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 481–494, 2019.

[5] Z. Xiaorong, W. Dianchun, L. Jin and T. Wei, "Incremental learning in the application of intrusion detection," in *Proc. of 2009 WRI Global Congress on Intelligent Systems*, Xiamen, China, pp. 549–553, 2009.

[6] S. Shalev-shwartz, "Online learning: Theory, algorithms, and applications," Ph.D. dissertation, University of Hebrew, Israel, 2007.

[7] F. Chu, Y. Wang and C. Zaniolo, "An adaptive learning approach for noisy data streams," in *Proc. of Fourth IEEE Int. Conf. on Data Mining (ICDM)*, Brighton, UK, pp. 351–354, 2004.

[8] J. Dromard, G. Roudiere and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34–47, 2016.

[9] L. Boukela, G. Zhang, S. Bouzefrane and J. Zhou, "An outlier ensemble for unsupervised anomaly detection in honeypots data," *Intelligent Data Analysis*, vol. 24, no. 4, pp. 743–758, 2020.

[10] J. V. Anand Sukumar, I. Pranav, M. Neetish and J. Narayanan, "Network intrusion detection using improved genetic K-means algorithm," in *Proc. of 2018 Int. Conf. on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, India, pp. 2441–2446, 2018.

[11] P. Kanimozhi and T. Aruldoss Albert Victoire, "Oppositional tunicate fuzzy C-means algorithm and logistic regression for intrusion detection on cloud," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 4, pp. 13, 2022.

[12] M. Amjad, H. Zahid, S. Zafar and T. Mahmood, "A novel deep learning framework for intrusion detection system," in *Proc. of 2019 Int. Conf. on Advances in the Emerging Computing Technologies (AECT)*, Al Madinah Al Munawwarah, Saudi Arabia, pp. 1–6, 2020.

[13] A. Singla, E. Bertino and D. Verma, "Preparing network intrusion detection deep learning models with minimal data using adversarial domain adaptation," in *Proc. of the 15th ACM Asia Conf. on Computer and Communications Security (ACM Asia CCS)*, Taipei, Taiwan, pp. 127–140, 2020.

[14] M. O. Miah, S. Shahriar Khan, S. Shatabda and D. M. Farid, "Improving detection accuracy for imbalanced network intrusion classification using cluster-based under-sampling with random forests," in *Proc. of 2019 1st Int. Confe. on Advances in Science, Engineering and Robotics Technology (ICASERT)*, Dhaka, Bangladesh, pp. 1–5, 2019.

[15] X. Zheng, Y. Wang, L. Jia, D. Xiong and J. Qiang, "Network intrusion detection model based on chi-square test and stacking approach," in *Proc. of 2020 7th Int. Conf. on Information Science and Control Engineering (ICISCE)*, Changsha, China, pp. 894–899, 2020.

[16] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. of the Sixth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Boston, MA, USA, pp. 71–80, 2000.

[17] C. Manapragada, G. I. Webb and M. Salehi, "Extremely fast decision tree," in *Proc. of the 24th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining*, London, UK, pp. 1953–1962, 2018.

[18] B. Xu, S. Chen, H. Zhang and T. Wu, "Incremental KNN-SVM method in intrusion detection," in *Proc. of 2017 8th IEEE Int. Conf. on Software Engineering and Service Science (ICSESS)*, Beijing, China, pp. 712–717, 2017.
[19] T. Wang, Q. Lv, B. Hu and D. Sun, "A few-shot class-incremental learning approach for intrusion detection," in *Proc. of 2021 Int. Conf. on Computer Communications and Networks (ICCCN)*, Athens, Greece, pp. 1–8, 2021.
[20] Edge-iiotset dataset, https://www.kaggle.com/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot, 2022.
[21] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, no. 1, pp. 40281–40306, 2022.