An IoT Intrusion Detection System Based on TON_IoT Network Dataset

Ge Guo
School of Computer Science
Wuhan Qingchuan University
Wuhan, China
guoge52501@gmail.com

Xuefeng Pan
School of Computer Science
Wuhan Qingchuan University
Wuhan, China
57336206@qq.com

He Liu
School of Computer Science
Wuhan Qingchuan University
Wuhan, China
11833855@qq.com

Fen Li
School of Computer Science
Wuhan Qingchuan University
Wuhan, China
25879030@qq.com

Lang Pei
School of Computer Science
Wuhan Qingchuan University
Wuhan, China
11286978@qq.com

Kewei Hu
School of Computer Science
Wuhan Qingchuan University
Wuhan, China
hukewei.yx@qq.com

2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC) | 979-8-3503-3286-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/CCWC57344.2023.10099144

***Abstract*—As the Internet of Things (IoT) rapidly proliferate in the world, new attacks exploiting the weaknesses of the unfledged IoT technologies are emerging constantly. An Intrusion Detection System (IDS) is a powerful tool to defend IoT systems against security threats by monitoring abnormal activities on networks. As an effective approach to detecting malicious behaviors, Machine Learning (ML) has gained substantial interest from researchers. An ML-based IDS framework for IoT systems is proposed in this study and ten learning methods are applied for performance evaluation based on a recently published dataset, the TON_IoT network dataset. Experimental results show that the stacking-ensemble model is the most optimal classifier, obtaining Matthews correlation coefficient (MCC) scores of 0.9971 and 0.9909 in the binary classification and the multiclass classification, respectively.**

Keywords—Machine learning, Intrusion Detection Systems, Feature selection, TON_IoT

I. INTRODUCTION

In recent years, new computing and communication technologies have been dazzlingly emerging and developing: cloud computing, smartphones, 5G, and various other wireless communication technologies, sensors, and artificial intelligence (AI). They enable much more human-to-machine and machine-to-machine interactions which are termed the Internet of Things [1]. The pervasive use of IoT devices raises many security issues in various IoT applications. The inherent vulnerabilities of IoT systems, including limited resource, heterogeneous components, unattended deploying environments, and large scale, make maintaining their security a challenging task.

As a paradigm-shifting technology, AI has obtained significant success in a wide range of fields such as fraud detection, spam filtering, and image recognition. To meet the particular security requirements of IoT, researchers have delved into employing machine learning, a subfield of AI, to enhance the defenses of IoT systems. One of the applications of ML in IoT security is IDS.

The detection ability of an ML-based IDS substantially relies on the quality of the dataset. A data-driven IDS tailored to IoT systems should be trained and assessed by IoT-related datasets which are collected on real-world testbeds that embrace IoT devices and realistic attacking traffic. Many datasets which were employed as the benchmark data in the works of this subject do not contain any properties of IoT applications, e.g., NSL-KDD [2], UNSW-NB15 [3], and CIC-IDS2017 [4]. Hence, we select a lately released IoT dataset, the TON_IoT network dataset [5], to act as the benchmark dataset. A framework for building an ML-based IoT IDS is suggested in this work and an IDS model is obtained by comparing the performance results of 10 ML methods. The novelty of the framework is that it integrates Spearman rank correlation coefficient as the feature selection method and builds a stacking-ensemble model by combining CatBoost, Extra Tree, and Extreme Gradient Boosting algorithms.

The remainder of this paper is organized as follows: Section II presents a brief survey of related study. Section III describes the framework for building the IoT IDS model. The evaluation results and discussion are presented in Section IV. Section V delivers the concluding remarks and provides the future research directions.

II. RELATED WORKS

IDS has constantly earned researchers' attention and applying ML in this field became a hot topic in recent years. A novel imputation approach based on feature transformation and incremental clustering was proposed in [6] for replacing missing values in datasets of IoT networks. The validating results confirmed the efficiency of the approach. In the study [7], the authors developed a hybrid dimension reduction scheme and investigated it on NSL-KDD, BoT-IoT, and DS2OS datasets. The evaluation results revealed that the proposed scheme performed very well with a detection rate above 90%.

Song et al. [8] use a generative deep learning algorithm, autoencoder, to detect intrusion in IoT networks. The authors conduct extensive experiments to find the optimal hyperparameter settings of the classifier using 3 datasets, NSL-KDD, IoTID20, and N-BaIoT. A hybrid approach combining

979-8-3503-3286-5/23/$31.00 ©2023 IEEE

two deep learning algorithms, convolution neural network (CNN) and long short-term memory (LSTM), is proposed in [9] for building an IoT IDS. The authors also use the particle swarm optimization technique (PSO) to reduce dimensionality for improving efficiency and obtain an accuracy of 99.82% on the IoTID20 dataset. Qaddoura et al. [10] suggest a two-stage scheme for IoT attack detection. The first stage utilizes Single-hidden Layer Feed-forward Neural Network (SLFN) to detect attacks and the second stage employs LSTM to identify the category of attack. The Synthetic Minority Oversampling Technique (SMOTE) is also used to solve the imbalanced issue of the benchmark dataset IoTID20. The fine-tuned approach achieves the performance with a G-mean of 78%.

An ensemble model named Extra Boosting Forest (EBF) is proposed in [11] to detect intrusion from multi-domain systems having both local network and IoT-based traffic. To evaluate the performance of the approach, the authors combine two datasets UNSW-NB15 and IoTID20 with the help of principal component analysis (PCA) to reduce both dimensionalities to 30. Experimental results show that the approach delivers accuracy scores of 98.5% and 98.4% for two and four classes, respectively. The researchers in [12] developed an IDS framework with two-stage processes for IoT systems. Firstly, the authors employed a Deep Sparse AutoEncoder (DSAE) method to conduct feature engineering. Secondly, an ensemble model combining LSTM and Deep Neural Network (DNN) was leveraged to identify attacks. Assessing results on IoT-23, LITNET-2020, and NetML-2020 datasets validate that the framework can provide adequate accuracy for the anomaly detection task.

An IDS named AIEMLA was implemented in [13] to counter attacks against an IoT routing protocol, RPL. To generate the dataset, the authors utilized Cooja to simulate 3 RPL attacks, namely Hello Flooding, Decreased Rank, and Increased Version. The effectiveness of AIEMLA was confirmed by using a model based on artificial neural network (ANN) algorithm. Osman et al. [14] simulated a dataset containing Version Number Attacks (VNA) in RPL-based IoT networks and proposed a lightweight VNA detection model by using Light Gradient Boosting Machine (LGBM) algorithm. Results demonstrated the proposed model could achieve an accuracy of 99.6%. The research in [15] focused on building an IDS against the combination of attacks against two common objective functions (OF) of RPL, namely OF0 and MRHOF. The authors applied multiple ML methods on the generated dataset and found MLP and RF are the best-performing classifiers for identifying such attacks.

III. Methodology

Our proposed framework of the IoT IDS is illustrated in Fig. 1. The framework consists of seven steps for building the model of attack recognition: data cleansing, feature encoding, data splitting, normalization, feature selection, classifier training, performance assessment. The latter five steps are implemented by 5-fold cross-validation.

A. Data

The TON_IoT network dataset is a recently published dataset aiming to provide a representative benchmark containing modern and realistic IoT cyberattacks. The testbed was constructed on an orchestrated architecture consisting of Edge, Fog, and Cloud layers, which were implemented based on NSX-VMware platform, Kali Linux, Node-RED Server, Hive-MQTT broker, and Cloud centers [16]. The dataset is composed of nine categories of cyberattacks, involving backdoor, DoS, DDoS, injection, Man-In-The-Middle (MITM), password cracking, ransomware, Scanning, and cross-site scripting (XSS) attacks. 43 features were extracted from the network-traffic files and 2 new features, namely 'label' and 'type', were added to tag the flow. The 'label' feature is for binary classification and the 'type' feature is for the multi-class recognition of attacking categories. Table I shows the distribution of the records in the dataset. One can observe that the dataset is imbalanced with normal records occupying the major portion.

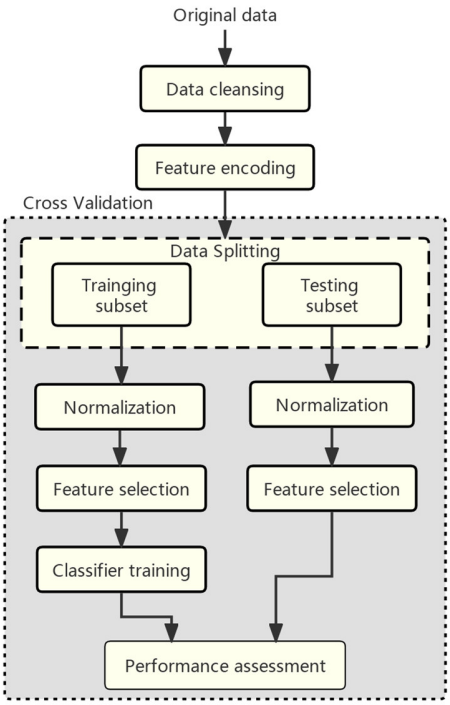


Fig. 1. The framework of the attack classification

TABLE I. Distribution of Normal and Attacking Records

Category of attack	Percentage
Backdoor	4.3%
DoS	4.3%
DDoS	4.3%
Injection	4.3%
MITM	0.2%
Normal	65.1%
Password	4.3%
Ransomware	4.3%
Scanning	4.3%
XSS	4.3%

B. Data cleansing

In the given dataset, 3 attributes, namely "timestamp", "src_ip", and "dest_ip", are removed because they are unrelated to the target variables. Containing discrete values, the "timestamp" feature is incapable to contribute to the attack recognition. Meanwhile, malicious hackers can launch attacks from legitimate users' computers, which demonstrates the "src_ip" and "dest_ip" features do not help for identifying intrusions and could incur overfitting issues for the classification. Moreover, there exists an unexpected character ('-') in the numeric "http_trans_depth" feature so they are replaced with the median value of this feature in all the available records. The application of the median imputation is due to its merit of being less sensitive to outlier errors than the mean imputation.

C. Feature encoding

The TON_IoT network dataset contains 23 nominal attributes, such as "conn_state", "service", and "proto", etc. For instance, the "http_method" feature has 3 values, namely 'GET', 'POST', and 'HEAD'. It is vital to transform these nominal attributes into the numeric form so that they are compatible with the application of many learning algorithms. Several techniques can be used for this conversion, including One Hot Encoding Scheme, Label Encoding, Feature Hashing Scheme, Bin-counting Scheme, Effect Coding Scheme, and Dummy Coding Scheme. We choose Label Encoding in this study because it does not change the dimensionality of the dataset and does not incur more time costs to the classifier training.

D. Data Splitting

To achieve a better bias-variance tradeoff, we choose a 5-fold cross-validation approach to divide the dataset. In each fold, 80% of the data is used for training and the rest 20% is used for testing. It is worth mentioning that the splitting of data should be performed before normalization and feature selection, ensuring data leakage can be prevented. The final results are the average values of 5 folds yielding a more accurate estimate. In this study, the cross-validation approach is implemented based on pipeline and grid search techniques, involving this step and the following four steps.

E. Normalization

It is common that the scales of numeric attributes are relatively different. For instance, in the given dataset, the range of the "dst_bytes" feature is much larger than the scale of the "http_request_body_len" feature. In such a case, training a distance-based classifier with the raw data will produce results being biased toward the "dst_bytes" feature. To address this issue, features should be transformed into a similar scale by using normalization techniques. The min-max scaler is used in our framework, which converts each attribute of the dataset into the range between 0 and 1 by applying the formula:

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

where x is the raw value of a feature, x_{max} and x_{min} refer to the maximum and the minimum values of the feature to which x belongs.

F. Feature Selection

Due to the resource-constrained trait of IoT systems, the computational complexity of IDSs for detecting IoT attacks should be lightweight. Feature selection is one of the crucial techniques for tackling this issue by removing redundant and unimportant features from datasets.

To obtain an optimal subset of features, we compare two feature selection methods, i.e., Spearman rank correlation coefficient and Chi-Squared statistic, by using Decision Tree as the assessing classifier. The feature subset attained by the better-performing method is selected as the input data fed into the training process.

Spearman rank correlation coefficient evaluates the monotonic relationship between two features in a non-parametric way. The coefficient takes values in the range of [-1,1], in which 1 and -1 denote perfect positive and negative correlations, respectively. The nearer it is to 0, the less associated between the features. Spearman rank correlation coefficient is simpler and more versatile than the conventional Pearson correlation coefficient. The two features that have the absolute values of such coefficients greater than a threshold are regarded as highly associated, and one of them will be discarded.

Chi-Squared statistic calculates the dependent weight of a feature with regard to response variables. Features are ranked according to their Chi-Squared scores concerning a class and a certain number of top features will remain as the subset of selection.

G. Model training and performance assessment

To attain an effective IoT IDS for monitoring malicious behaviors, we first train the processed dataset with 8 base models, namely Adaptive Boosting (AB), CatBoost (CB), Decision Tree (DT), Extra Tree (ET), Gradient Boosting (GB), k-Nearest Neighbor (kNN), Random Forest (RF), and Extreme Gradient Boosting (XGB). For the sake of a fair comparison, all the models use the default hyperparameters. Then, three best-performing classifiers are integrated by applying two ensemble learning techniques, namely soft voting (En_V) and stacking (En_S), to build two new ensemble models. Eventually, all ten models are compared according to seven performance metrics to obtain the most effective classifier for attack detection in IoT environments.

IV. Results and Discussion

A. Experimental settings

The experiments are carried out on a PC operated on 64-bit Windows 7, Intel(R) Xeon® CPU E5-2650 v3 2.30 GHz, and 16GB RAM. Both binary classification (normal or attack) and multiclass classification (the category of attacks) are performed. The implementation is based on multiple python libraries, including Numpy, Pandas, Scikit-learn, Seaborn, and Matplotlib.

B. Performance metrics

The ten classification models are evaluated with seven metrics: accuracy, precision, recall, F1-score, Matthews correlation coefficient (MCC), Area Under the Receiver

Operating Characteristic Curve (AUC), and time consumption. From the aforementioned description, we observe that the TON_IoT network dataset is imbalanced. Since taking all four parts (True Positive, True Negative, False Positive, False Negative) of the confusion matrix into account, MCC is a better metric than F1-score to evaluate the performance of such imbalanced datasets. Therefore, we use MCC as the primary metric in this study.

C. *Feature-selection results*

To attain the optimal subset of features for attack detection, we compare the two feature selection techniques. First, the threshold of Spearman rank correlation coefficient is set to be 0.9. As shown in Table II, 18 features are found to have high relevance with some other features so they are eliminated from the dataset. Since the total number of predictive features in the cleaned datasets is 40, we choose 22 features having top-ranked Chi-Squared scores for a fair comparison. Then, we apply DT algorithm to both selected subsets and compare their values of MCC. The results show Spearman rank correlation coefficient technique achieves a better performance. Therefore, it is used as the feature selection method, and the heatmap illustrating the correlation of remaining features is shown in Fig. 2.

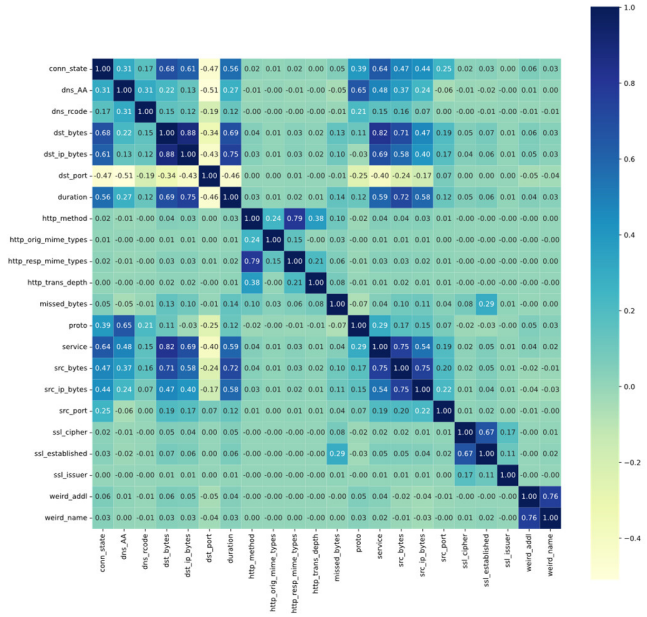


Fig. 2. The correlation heatmap of the 22 selected features

TABLE II. ELIMINATED REDUNDANT FEATURES

No	Feature Name	No	Feature Name
1	dns_RA	10	http_status_code
2	dns_RD	11	http_uri
3	dns_qclass	12	http_user_agent
4	dns_qtype	13	http_version
5	dns_query	14	src_pkts
6	dns_rejected	15	ssl_resumed
7	dst_pkts	16	ssl_subject
8	http_request_body_len	17	ssl_version
9	http_response_body_len	18	weird_notice

D. *Results and analysis*

At the first stage of assessment, we obtain three best-performing models, namely CB, ET, and XGB. Therefore, they are used as the base learners to form the voting model and the stacking model. The meta classifier of the stacking model is Logistic Regression. The two ensemble models are assessed at the second stage. The evaluation results of the binary classification and the multiclass classification are presented in Table III and Table IV, with the maximum values of the performance metrics emphasized in bold type. The result of comparing MCC among the models is depicted in Fig. 3 and the computational costs of the models are illustrated in Fig. 4.

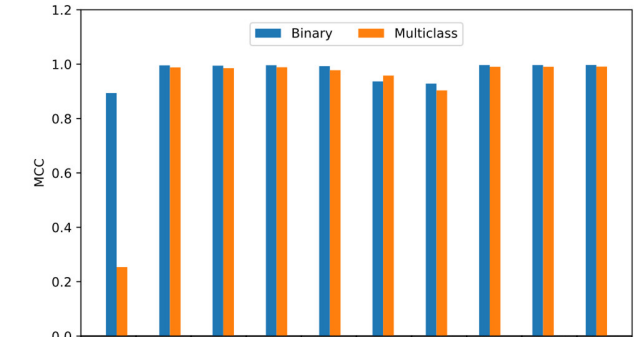
From Table III we can observe that the stacking model achieves the highest MCC score with a value of 0.9971. In the multiclass classification for identifying the categories of attacks, the stacking model outperforms other models with an MCC score of 0.9909. We can also find that the stacking model attains the highest scores in terms of other metrics except AUC in the multiclass prediction, which corroborate its better performance than other models. However, as shown in Fig. 4, its achievement is at the cost of lengthy processing time.

TABLE III. RESULTS OF PREDICTING THE BINARY CATEGORIES

	Acc	Prec	Rec	F1	MCC	AUC
AB	0.9516	0.9517	0.9516	0.9517	0.8937	0.9472
CB	0.998	0.998	0.998	0.998	0.9956	0.9978
DT	0.9976	0.9976	0.9976	0.9976	0.9946	0.9974
ET	0.9983	0.9983	0.9983	0.9983	0.9963	0.9981
RF	0.9967	0.9968	0.9967	0.9967	0.9928	0.9957
GB	0.971	0.9711	0.971	0.971	0.9364	0.9694
kNN	0.9675	0.9676	0.9675	0.9675	0.9286	0.9647
XGB	0.9985	0.9985	0.9985	0.9985	0.9967	0.9983
En_V	0.9986	0.9986	0.9986	0.9986	0.9969	0.9983
En_S	**0.9987**	**0.9987**	**0.9987**	**0.9987**	**0.9971**	**0.9984**

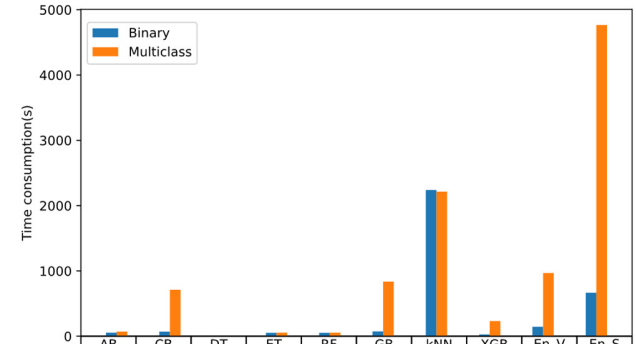
TABLE IV. RESULTS OF PREDICTING THE MULTICLASS CATEGORIES

	Acc	Prec	Rec	F1	MCC	AUC
AB	0.5604	0.5976	0.5604	0.5646	0.2539	0.6139
CB	0.9934	0.9936	0.9934	0.9935	0.9883	0.9999
DT	0.9917	0.9917	0.9917	0.9917	0.9852	0.9947
ET	0.9936	0.9936	0.9936	0.9936	0.9887	0.9991
RF	0.9875	0.9874	0.9875	0.9872	0.9777	0.9997
GB	0.9766	0.9761	0.9766	0.9757	0.958	0.998
kNN	0.9459	0.9457	0.9459	0.9456	0.9036	0.9849
XGB	0.9946	0.9947	0.9946	0.9947	0.9905	**1**
En_V	0.9947	0.9948	0.9947	0.9947	0.9905	**1**
En_S	**0.9949**	**0.9949**	**0.9949**	**0.9949**	**0.9909**	0.9999



	AB	CB	DT	ET	RF	GB	kNN	XGB	En_V	En_S
Binary	0.8937	0.9956	0.9946	0.9963	0.9928	0.9364	0.9286	0.9967	0.9969	0.9971
Multiclass	0.2539	0.9883	0.9852	0.9887	0.9777	0.9580	0.9036	0.9905	0.9905	0.9909

Fig. 3. The comparison of MCC among the models



	AB	CB	DT	ET	RF	GB	kNN	XGB	En_V	En_S
Binary	54.2	69.9	3.3	51.8	51.8	73.8	2239.7	27.5	143.6	665.9
Multiclass	71.6	711.5	3.4	54.6	54.1	836.2	2213.8	231.7	967.5	4766.7

Fig. 4. The comparison of time consumption among the models

Fig. 5 and Fig. 6 present the normalized confusion matrixes of the binary and multiclass classification by using the stacking model combining CB, ET, and XGB. We can see from Fig.5 that the stacking model can perfectly predict whether a record is benign or malicious. In Fig. 6, only the "MITM" attacks are predicted slightly poorer than other categories of attacks. This is because of the smaller number of records with the "MITM" label, which only occupies 0.2% of the total samples in the given dataset.

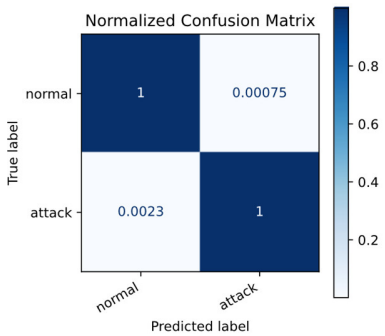


Fig. 5. The confusion matrix of En_S in the binary classification

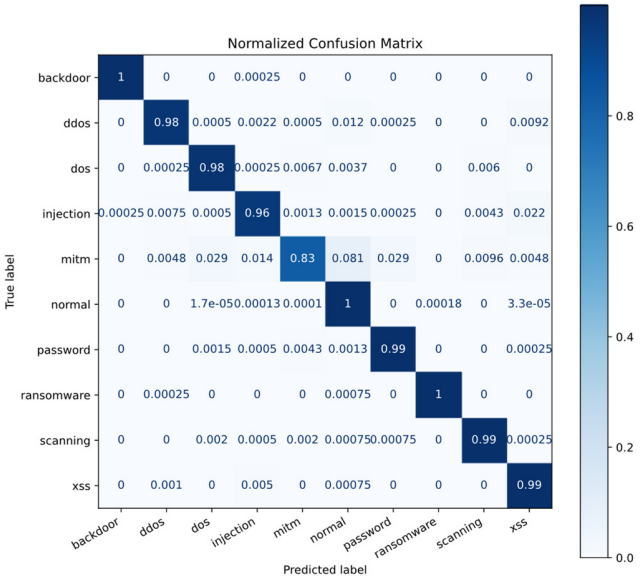


Fig. 6. The confusion matrix of En_S in the multiclass classification

Table V shows the comparison of our proposed model with other existing models that are evaluated under the TON_IoT network dataset. Since the other existing models were not assessed in term of MCC, we choose the F1 score as the metric of comparison due to the imbalanced characteristic of the dataset. We can observe that our model attains superior performances in both binary and multiclass classifications.

TABLE V. COMPARISON WITH EXISTING WORKS ASSESSED ON THE TON_IOT NETWORK DATASET

IDS Models	ML Methods	Task	F1
Alsaedi et al. [17] (2020)	CART	Binary	0.88
		Multiclass	0.75
Kumar et al. [18] (2021)	Stacking ensemble	Binary	0.9503
Kumar et al. [19] (2021)	TP2SF	Multiclass	0.9528
Disha et al. [20] (2022)	GIWRF-DT	Binary	0.9985
Our model (2022)	Stacking ensemble	Binary	0.9987
		Multiclass	0.9949

V. CONCLUSION AND FUTURE WORKS

In this study, we propose a data-driven IDS for IoT networks based on the TON_IoT network dataset, which contains realistic IoT attacks and heterogeneous network components. After comparing 8 base classifiers and 2 ensemble classifiers, the stacking model integrating CatBoost, Extra Tree, and XGBoost is finally chosen as the method for classification due to its outstanding performance. The evaluation results show that our proposed model outperforms other existing machine learning models based on the same dataset, by recording MCC scores of 0.9971 and 0.9909 in the binary and multiclass classification, respectively.

As future works, we will explore other advanced feature selection methods and deep learning algorithms for attack detection in IoT environments based on more IoT datasets.

REFERENCES

[1] S. Ray, Y. Jin, and A. Raychowdhury, "The changing computing paradigm with internet of things: A tutorial introduction," IEEE Design & Test, vol. 33, no. 2, pp. 76-96, 2016.

[2] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE symposium on computational intelligence for security and defense applications, 2009: Ieee, pp. 1-6.

[3] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in 2015 military communications and information systems conference (MilCIS), 2015: IEEE, pp. 1-6.

[4] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," ICISSp, vol. 1, pp. 108-116, 2018.

[5] N. Moustafa. TON_IoT Datasets. [Online]. Available: https://cloudstor.aarnet.edu.au/plus/s/ds5zW91vdgjEj9i

[6] R. Vangipuram, R. K. Gunupudi, V. K. Puligadda, and J. Vinjamuri, "A machine learning approach for imputation and anomaly detection in IoT environment," Expert Systems, vol. 37, no. 5, p. e12556, 2020.

[7] P. Kumar, G. P. Gupta, and R. Tripathi, "Toward design of an intelligent cyber attack detection system using hybrid feature reduced approach for iot networks," Arabian Journal for Science and Engineering, vol. 46, no. 4, pp. 3749-3778, 2021.

[8] Y. Song, S. Hyun, and Y.-G. Cheong, "Analysis of Autoencoders for Network Intrusion Detection," Sensors, vol. 21, no. 13, p. 4294, 2021.

[9] H. Alkahtani and T. H. Aldhyani, "Intrusion detection system to advance internet of things infrastructure-based deep learning algorithms," Complexity, vol. 2021, 2021.

[10] R. Qaddoura, M. Al-Zoubi, H. Faris, and I. Almomani, "A multi-layer classification approach for intrusion detection in iot networks based on deep learning," Sensors, vol. 21, no. 9, p. 2987, 2021.

[11] P. L. Indrasiri, E. Lee, V. Rupapara, F. Rustam, and I. Ashraf, "Malicious traffic detection in iot and local networks using stacked ensemble classifier," Computers, Materials and Continua, vol. 71, no. 1, pp. 489-515, 2022.

[12] V. Dutta, M. Choraś, M. Pawlicki, and R. Kozik, "A deep learning ensemble for network anomaly and cyber-attack detection," Sensors, vol. 20, no. 16, p. 4583, 2020.

[13] S. Sharma and V. K. Verma, "AIEMLA: artificial intelligence enabled machine learning approach for routing attacks on internet of things," The Journal of Supercomputing, vol. 77, no. 12, pp. 13757-13787, 2021.

[14] M. Osman, J. He, F. M. M. Mokbal, N. Zhu, and S. Qureshi, "ML-LGBM: A Machine Learning Model based on Light Gradient Boosting Machine for the Detection of Version Number Attacks in RPL-Based Networks," IEEE Access, vol. 9, pp. 83654-83665, 2021.

[15] J. Foley, N. Moradpoor, and H. Ochenyi, "Employing a machine learning approach to detect combined internet of things attacks against two objective functions using a novel dataset," Security and Communication Networks, vol. 2020, 2020.

[16] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," Sustainable Cities and Society, vol. 72, p. 102994, 2021.

[17] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," IEEE Access, vol. 8, pp. 165130-165150, 2020.

[18] P. Kumar, G. P. Gupta, and R. Tripathi, "An ensemble learning and fog-cloud architecture-driven cyber-attack detection framework for IoMT networks," Computer Communications, vol. 166, pp. 110-124, 2021.

[19] P. Kumar, G. P. Gupta, and R. Tripathi, "TP2SF: A Trustworthy Privacy-Preserving Secured Framework for sustainable smart cities by leveraging blockchain and machine learning," Journal of Systems Architecture, vol. 115, p. 101954, 2021.

[20] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," Cybersecurity, vol. 5, no. 1, pp. 1-22, 2022.