Sensors and Materials, Vol. 36, No. 3 (2024) 1035–1046
MYU Tokyo

S & M 3579

Feature Selection for Malicious Detection on Industrial IoT Using Machine Learning

Hong-Yu Chuang and Ruey-Maw Chen*

Computer Science and Information Engineering, National Chin-Yi University of Technology,
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 411030, Taiwan (R.O.C.)

(Received September 20, 2023; accepted March 12, 2024)

Keywords: industrial IoT (IIoT), intrusion detection system, network security, network features

The rapid deployment of IoT devices for enhanced convenience and increased production efficiency has resulted in a significant rise in the potential for cyberattacks. Consequently, the detection of malicious attacks has become a crucial concern in industrial IoT (IIoT) applications. Furthermore, IoT usage is continuously expanding, with new functional IoT devices connecting to the network daily, leading to a substantial increase in network traffic. To address the need for an intrusion detection system (IDS) to identify malicious attacks under a high-traffic condition, a highly efficient IDS is essential. In this study, an IDS based on machine learning (ML) with a reduced set of features on the TON_IoT dataset is employed. The TON_IoT includes telemetry data, operating systems data, and network data for an IoT network. A Pearson correlation coefficient (PCC) was applied to assess the correlations among packet features, and a filtering rule based on the Jamovi software's frequency table was used to identify the most essential features within the TON_IoT dataset. Finally, the 45 original features were narrowed down to 10 core features for the IDS to effectively detect intrusion activities. To evaluate the detection performance of malicious intrusion activities using the yielded set of 10 core features, we utilized evaluation metrics including accuracy, precision, recall, and $F1$-score. Four ML techniques, namely, K-Nearest Neighbors, Random Forest, Naïve Bayes, and eXtreme Gradient Boosting, were tested. The experimental results demonstrated that the four ML techniques could detect multiple types of attack with an accuracy exceeding 96% and with a recall rate over 97%, underscoring the effectiveness and efficiency of utilizing the reduced 10 core features for malicious attack detection while maintaining a high level of accuracy.

1. Introduction

A report entitled "Cyber Signals ISSUE 3" published by Microsoft at the end of 2022 highlighted the increasing trend among organizations to integrate IoT and operational technology[1,2] devices into their operations. While this integration has led to improved production efficiency, it has also exposed vulnerabilities to various forms of attack. Alarmingly, the report noted that as much as 75% of industrial devices remained unpatched for high-risk

*Corresponding author: e-mail: rmchen@ncut.edu.tw
https://doi.org/10.18494/SAM4666

ISSN 0914-4935
https://myukk.org/

vulnerabilities. The diverse devices have made it challenging to update the underlying equipment, posing a significant obstacle in defending against malicious attacks within IoT/industrial IoT (IIoT) environments. Among the various defense technologies, an intrusion detection system (IDS) based on machine learning (ML) has emerged as a crucial tool for early anomaly detection.

IDS technology based on ML is becoming increasingly widespread. For instance, Linear Discriminant Analysis (LDA), Random Forest (RF), and Classification and Regression Trees (CART) were used for intrusion classification.[3] Meanwhile, a hybrid learning model based on K-Nearest Neighbors (KNN) for intrusion detection was developed, evaluating its performance using the KDD-Cup 99 dataset.[4]

The effectiveness of ML-based IDSs hinges on the diversity of the dataset they are trained on. In this context, the TON_IoT dataset, proposed by UNSW in 2020, emerges as a valuable resource.[5] Given the continuous expansion of the IoT market and the daily influx of new IoT devices deployed, network traffic experiences rapid growth. To enable IDSs to efficiently identify malicious attacks under a high-traffic network condition, the need for highly efficient IDSs becomes evident. Restated, the required packet features have to be cut down to lower the computation complexity.[6] In a separate study, Guo *et al.* proposed using the Spearman rank correlation coefficient as a feature selection method to develop an ML-based IDS framework specifically for IoT systems.[7] Ultimately, 18 features were selected for the development of the stacking model. The performance of the stacking model on the TON_IoT network dataset was evaluated. The experimental results showed that multi class classification and binary classification achieved accuracies of 0.9949 and 0.9987, respectively.[7] Additionally, Telikani *et al.* used RF feature analysis for feature selection and ultimately selected 23 features. The performance of the hybrid model on the TON_IoT network dataset was evaluated. The experimental results showed that multi class classification and binary classification achieved precisions of 97.3 and 98.1%, respectively.[8]

To cut down the packet features used for ML, in this work, we applied the Pearson correlation coefficient (PCC) to calculate feature correlations among the dataset's attributes and utilized Jamovi analysis software to generate a frequency table for identifying the key features within the TON_IoT dataset. Finally, the original 45 features were reduced to a set of 10 core data features, which are subsequently employed to detect potential malicious attack activities.

The objective of this study revolves around the development of an ML-based IDS to detect malicious attacks in an IIoT circumstance using the 10 determined core data features. To verify the ML-based IDS detection performance of malicious intrusion behaviors using the yielded 10 core features, ML techniques such as KNN, RF, Naïve Bayes (NB), and eXtreme Gradient Boosting (XGB) are employed for the multiclass classification test, utilizing the TON_IoT dataset. To assess the model's reliability and performance, metrics such as accuracy, precision, recall, and *F*1-score are evaluated.

2. Datasets

To enhance the performance of ML techniques in detecting malicious activities, a statistical analysis and ML evaluation on the newly introduced ToN_IoT dataset were conducted by Booij *et al*.[9] They also compared the ToN_IoT dataset with other IoT datasets, highlighting the significance of dataset diversity and how differences between datasets can significantly affect detection performance. The ToN_IoT dataset contains recorded attack data in a real-world IIoT environment. Its distinctive advantage lies in its provision of up to ten distinct attack categories and diverse data types, including packet data, sensor data, and log data.

In this study, we developed a three-layered test platform, encompassing edge, fog, and cloud layers, to gather data that mirrors real data in IoT/IIoT network environments. The TON_IoT dataset comprises three distinct data types: raw, log, and sensor, and it incorporates nine different types of IoT device. It offers nine categories of attack methods: ransomware, XSS, backdoors, injections, DoS, DDoS, passwords, scanning, and MITM. The frequency of attacks in each category is documented as depicted in Table 1. The TON_IoT dataset encompasses 45 features, two of which are used for data labeling as illustrated in Table 2.

Table 1
Records on TON_IoT network dataset.

Type	No. of rows
normal	300000
backdoor	20000
injection	20000
password	20000
scanning	20000
ransomware	20000
xss	20000
ddos	20000
dos	20000
mtim	1043

Table 2
Features of TON_IoT dataset.

Activity	Features
Data labeling	label, type
Connection	ts, src_ip, src_port, src_btyes, dst_bytes, service, duration, dst_ip, dst_port, proto, conn_state, missed_bytes
HTTP	http_trans_depth, http_request_body_len, http_response_body_len, http_status_code, http_user_agent,http_method, http_uri, http_version, http_orig_mime_types, http_resp_mime_types
SSL	ssl_established, ssl_subject, ssl_version, ssl_cipher, ssl_resumed, ssl_issuer
DNS	dns_AA, dns_RD, dns_RA, dns_query, dns_qclass, dns_qtype, dns_rcode, dns_rejected
Statistical	src_pkts, dst_pkts, src_ip_bytes, dst_ip_bytes
Violation	weird_name, weird_notice, weird_addl

3. Methodology

3.1 Data cleaning

We utilized the Train_Test_Network feature subset part of the TON_IoT dataset. Two label features, "dns_query" and "http_uri," which represent DNS query responses and URL post parameters, respectively, are excluded since they contain less information to be effectively used in identifying the malicious attacks. Additionally, features marked with a "-" were replaced with a value of 0 during data preprocessing.

Moreover, non-numeric features were converted into numerical values. The label encoding scheme was employed for this purpose without increasing the dimensionality of the dataset.

3.2 Feature selection

The suggested process of feature selection is illustrated in Fig. 1. Initially, the PCC is employed to assess the relationships between features, and then the top 20 features with high PCCs are retained. Subsequently, a frequency table for each feature is generated through the Jamovi analysis indicating the occurrence frequency of attack types associated with each feature value. Finally, less significant features are filtered out from the dataset, and the selected core features used for ML are obtained.

The PCC is utilized to measure the strength and direction of the linear relationship between two variables. The correlation value (cv) range is $cv \in [-1, 1]$, where −1 indicates a perfect negative correlation, 1 indicates a perfect positive correlation, and 0 indicates no correlation. the PCC has been widely used in many applications, such as in checking whether signals and noise are corrected[10] and verifying whether rolling bearing vibration signals and bearing health are correlated.[11]

In this research, each feature was examined for its correlations with other features. A total correlation value (TCV) for each feature was calculated by adding those correction values that are greater than 0, i.e., $TCV = \sum_i cv_i, cv_i > 0$. These TCVs were then sorted in descending order. The first half of the TCV features, i.e., the top 20 TCV features, were retained. An example of the PCC using four randomly selected features in the TON_IoT dataset and the corresponding sorted TCV are shown in Table 3.

Figures 2 and 3 show the counts of different attack-type occurrences on different "ssl_cipher" and "dns_rejected" feature values. Figure 2 highlights that the packet distribution is unevenly associated with the feature values, which could potentially introduce classification bias towards

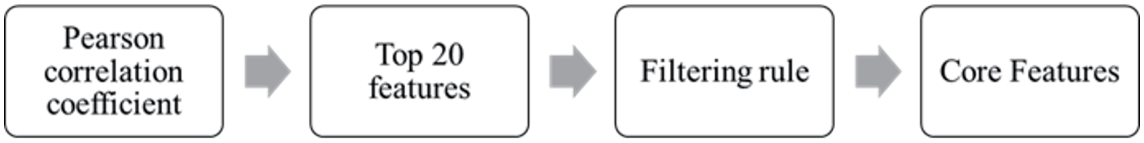

Fig. 1. Feature selection.

Table 3
Four features of PCC and *TCV* example.

		Features				
		1	2	3	4	*TCV*
Features	3	0.8890	0.9374	1	−0.1980	1.8265
	1	1	0.8398	0.8890	−0.2290	1.7289
	2	0.8398	1	0.9374	−0.1771	1.7773
	4	−0.2290	−0.1771	−0.1980	1	0

Frequencies of ssl_cipher

ssl_cipher	type	Counts	ssl_cipher	type	Counts
0	normal	299696	4	normal	11
	backdoor	20000		backdoor	0
	injection	20000		injection	0
	password	20000		password	0
	scanning	20000		scanning	0
	ransomware	20000		ransomware	0
	xss	20000		xss	0
	ddos	19998		ddos	0
	dos	20000		dos	0
	mitm	1043		mitm	0
⋮	⋮	⋮		normal	0
3	normal	1	5	backdoor	0
	backdoor	0		injection	0
	injection	0		password	0
	password	0		scanning	0
	scanning	0		ransomware	0
	ransomware	0		xss	0
	xss	0		ddos	2
	ddos	0		dos	0
	dos	0		mitm	0
	mitm	0			

Fig. 2. Frequency table of ssl_cipher.

the dominant class.[12] To address this issue, a frequency table filtering rule was introduced; if the count of at least 8 different packet types within a specific feature value is 0, then the particular feature is excluded from consideration for malicious detection.

Finally, 10 features are selected for use in ML after the above-stated PCC and frequency table processing. The 10 determined core features are shown in Table 4.

3.3 ML methods

Before applying ML, features have to be processed to prevent feature skewing. Hence, a normalization method is employed to standardize attributes to a uniform scale. A min–max scaler, as defined in Eq. (1), is utilized for adjusting feature values. In Eq. (1), "x" represents a

Frequencies of dns_rejected

dns_rejected	type	Counts	dns_rejected	type	Counts
0	normal	214273	2	normal	371
	backdoor	20000		backdoor	0
	injection	19295		injection	141
	password	19823		password	2
	scanning	19424		scanning	509
	ransomware	20000		ransomware	0
	xss	18600		xss	262
	ddos	14730		ddos	4930
	dos	18522		dos	276
	mitm	491		mitm	2
1	normal	85356			
	backdoor	0			
	injection	564			
	password	175			
	scanning	67			
	ransomware	0			
	xss	1138			
	ddos	340			
	dos	1202			
	mitm	550			

Fig. 3. Frequency table of dns_rejected.

Table 4
Ten selected 10 core features.

No.	Feature
1	http_version
2	proto
3	conn_state
4	weird_notice
5	dns_rejected
6	dns_RA
7	dns_AA
8	dns_RD
9	ts
10	type

specific feature value, while "x_{max}" and "x_{min}" denote the maximum and minimum feature values, respectively. Accordingly, the resulting feature value "x'" is rescaled to fall within the range of 0 to 1.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

There are four supervised ML techniques, KNN, RF, NB, and XGB, which have been well applied in various multi class classification applications with good performance. KNN[13] is a supervised ML algorithm that can be employed to tackle both classification and regression problems. It does not necessitate any preprocessing of all labeled samples before its application. To predict the value of new data points, the algorithm utilizes the concept of "feature similarity".

RF[14] is an ensemble learning method that operates by constructing a multitude of decision trees. Each tree serves as a fundamental classifier, and the outcomes are derived from the analysis of these decision trees.

XGB[15] is derived from the Gradient Boosting Decision Tree (GBDT).[16] Compared with GBDT, the advantage of XGB lies in its support for linear classifiers and its use of second-order derivatives to perform Taylor expansion on the cost function, resulting in more accurate outcomes.

NB[17] is a simple yet powerful probability estimator that operates on the application of Bayes' theorem, with the assumption that the attributes under consideration are independent among all attributes. This implies that each feature independently influences the outcome.

4. Results and Discussion

4.1 Split dataset

The TON_IoT dataset is divided in such a way that 70% of the data is utilized for training, while the remaining 30% is set aside for testing.

4.2 Performance metrics

To evaluate the performance of ML models on the reduced 10 core features, four metrics were used, namely, accuracy, precision, recall, and *F*1-score. These measures are calculated using true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as defined in Eqs. (2)–(5).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

4.3 Analysis and evaluation

To verify the ML performance of applying the proposed scheme, various combinations with PCC treatment and frequency table filtering rule were tested, as shown in Table 5. When the top 20 features are retained after PCC processing and the 10 core features are obtained after using frequency table filtering rules, the maximum average accuracy of the four models can be obtained at 0.9904. Additionally, even if only the top 20 features are retained after PCC processing without using frequency table filtering rules, the maximum average accuracy can be obtained at 0.9903, as indicated in the table.

Figures 4 and 5 show the attack classification test performances, namely, precision, recall, and *F*1-score using both the KNN and RF models, respectively, including using the reduced 10 core features with the original 44 features. The KNN classification results using the 10 core features outperform those using the 44 features across most attack types except for the "Passwd" type, and with a notable improvement in the 'mitm' attack category, the recall rate is over 0.98. Meanwhile, the classification results of RF using the 10 core features exhibit strong consistency with those of RF using the original 44 features across the most attack categories. Similarly, in the classification results of RF using 10 core features, the performances for "Passwd" and "mitm" attack types are similar to the effects of using KNN.

The NB and XGB models' test performances are depicted in Figs. 6 and 7, respectively, also using the reduced 10 core features and the original 44 features. The XGB classification results show a precision of over 0.98 across all data types, whether using 10 core features or using the original 44 features. Furthermore, Table 6 provides the averaged precision, recall, and *F*1-score for all data types using both the 10 core features and 44 original features. The maximum and minimum recall rates obtained by applying 10 core features are 0.99885 and 0.97800, which were used in the XGB and NB models, respectively.

Table 5
Classification results of models.

PCC	Filtering rule	Core Features	Model Accuracy				SUM AVG.
			KNN	NB	RF	XGB	
w/o	w/o	44	0.9973	0.9731	0.9995	0.9999	0.9924
top 15	w/o	16	0.7392	0.0894	0.7687	0.7687	0.5915
top 15	>0 8	8	0.69	0.1814	0.7689	0.7689	0.6023
top 15	>0 7	8	0.69	0.1814	0.7689	0.7689	0.6023
top 15	>0 6	8	0.69	0.1814	0.7689	0.7689	0.6023
top 20	w/o	21	0.9977	0.9675	0.998	0.9981	0.9903
top 20	**>0 8**	**10**	**0.9985**	**0.9657**	**0.9988**	**0.9989**	**0.9904**
top 20	>0 7	7	0.9974	0.9657	0.9968	0.9968	0.9891
top 20	>0 6	7	0.9974	0.9657	0.9968	0.9968	0.9891
top 50%	w/o	23	0.998	0.9636	0.9976	0.9977	0.9892
top 50%	>0 8	10	0.9985	0.9657	0.9988	0.9989	0.9904
top 50%	>0 7	7	0.9974	0.9657	0.9968	0.9968	0.9891
top 50%	>0 6	7	0.9974	0.9657	0.9968	0.9968	0.9891
top 60%	w/o	27	0.9964	0.9656	0.9984	0.9984	0.9897
top 60%	>0 8	11	0.9976	0.9636	0.9983	0.9984	0.9894
top 60%	>0 7	8	0.9975	0.9642	0.9977	0.9977	0.9892
top 60%	>0 6	8	0.9975	0.9642	0.9977	0.9977	0.9892

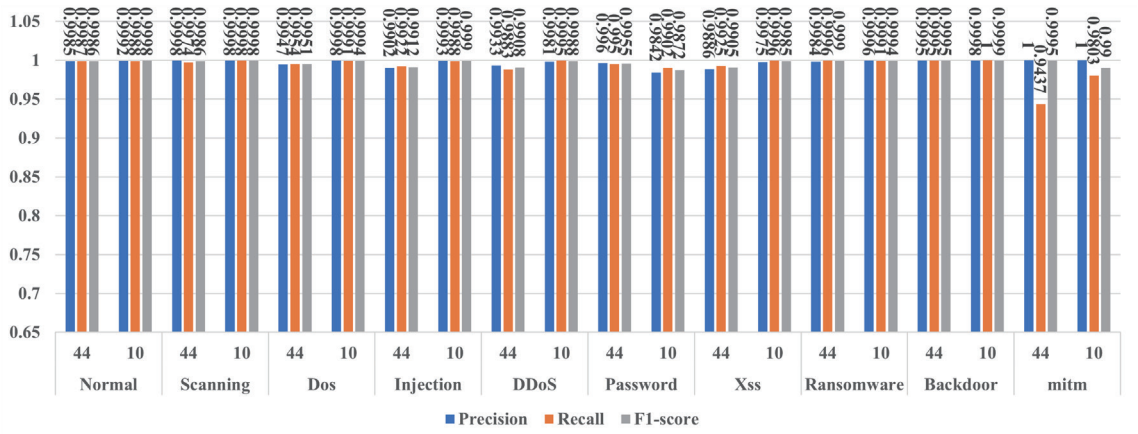

Fig. 4. (Color online) KNN model classification results.

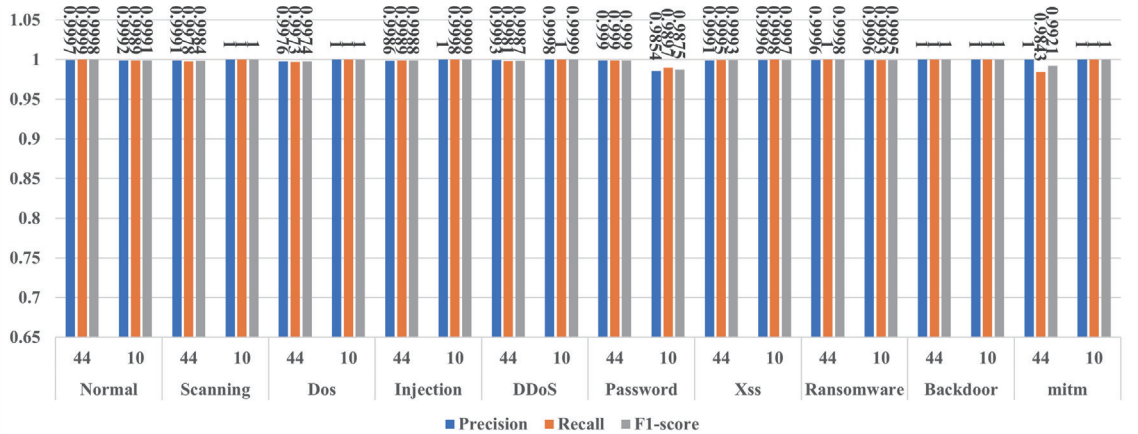

Fig. 5. (Color online) RF model classification results.

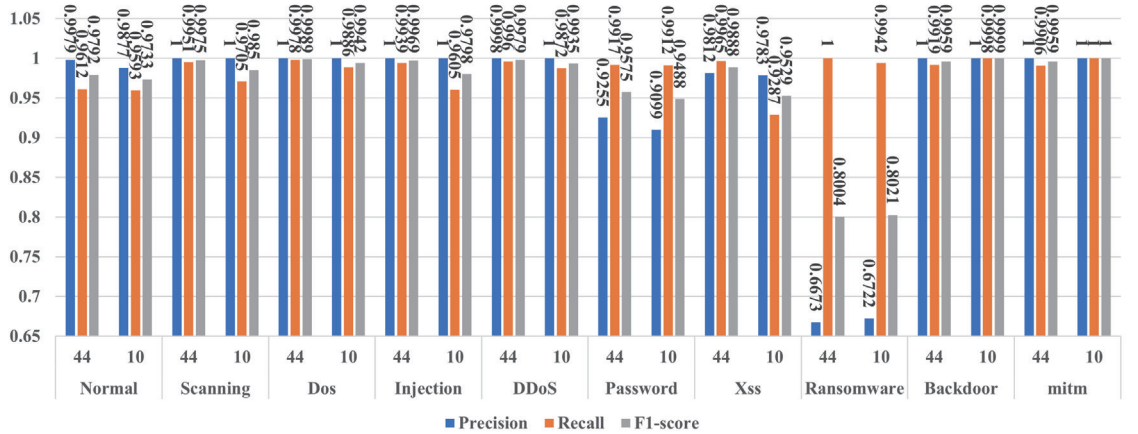

Fig. 6. (Color online) NB model classification results.

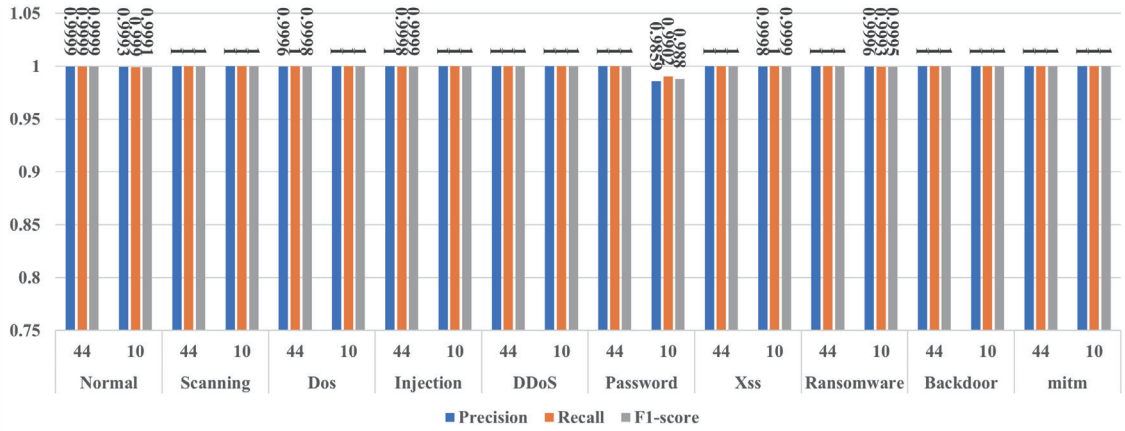

Fig. 7. (Color online) XGB model classification results.

Table 6
Averaged performance metrics.

Model	Precision	Recall	*F*1-score
KNN (44 features)	0.9959	0.99023	0.99583
KNN (10 features)	0.99992	0.99748	0.99833
RF (44 features)	0.9992	0.99748	0.99833
RF (10 features)	0.99773	0.99652	0.99718
NB (44 features)	0.95717	0.99147	0.97089
NB (10 features)	0.95481	0.97800	0.96295
XGB (44 features)	0.99995	0.99997	0.99996
XGB (10 features)	0.99846	0.99885	0.99865

5. Conclusions

We proposed a scheme combining PCC and a frequency filtering rule to select the core data features to enhance the ML model's efficiency. The PCC was calculated between features, and 20 features were chosen on the basis of *TCV*. A filtering rule associated with frequency analysis was used to select the core features on the TON_IoT network dataset, and the 45 original features were cut down to 10 core features. Four ML models (KNN, RF, NB, and XGB) were trained and tested using the 10 selected core features (as listed in Table 4) to confirm the efficiency of malicious attack detection efficiency in an IIoT environment. The recall rate indicates how many of the data samples that are true are predicted correctly. Hence, the most important performance index in the detection of malicious attacks in an IIoT environment is the recall rate. The experimental results indicate a recall rate of using 10 core data features exceeding 0.97 for all four ML models, with the KNN, RF, and XGB models achieving a recall of over 0.996, as shown in Table 6. Therefore, the 10 selected core features proposed in this study remain effective and efficient in detecting malicious attack activities and classifying various attack types in an IIoT environment.

References

1 Mircosoft Security Reports Cyber signals Issue 3: The convergence of IT and OT. https://www.microsoft.com/en-us/security/business/security-insider/reports/cyber-signals/cyber-signals-issue-3-the-convergence-of-it-and-ot/ (accessed August 2023).
2 B. Cunha and C. Sousa: 2021 16th Iberian Conf. Information Systems and Technologies (CISTI) (2021) 1. https://doi.org/10.23919/CISTI52073.2021.9476342
3 T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan: Procedia Comput. Sci. **171** (2020) 1251. https://doi.org/10.1016/j.procs.2020.04.133
4 L. Li, H. Zhang, H. Peng, and Y. Yang: Chaos Solitons Fractals **110** (2018) 33. https://doi.org/10.1016/j.chaos.2018.03.010
5 A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar: IEEE Access **8** (2020) 165130. https://doi.org/10.1109/ACCESS.2020.3022862
6 H. N. Saha, A. Mandal, and A. Sinha: 2017 IEEE 7th Annual Computing and Communication Workshop and Conf. (CCWC) (Las Vegas, NV, USA, 2017) 1. https://doi.org/10.1109/CCWC.2017.7868439
7 G. Guo, X. Pan, H. Liu, F. Li, L. Pei, and K. Hu: 2023 IEEE 13th Annual Computing and Communication Workshop and Conf. (CCWC) (Las Vegas, NV, USA, 2023) 0333. https://doi.org/10.1109/CCWC57344.2023.10099144
8 A. Telikani, J. Shen, J. Yang, and P. Wang: IEEE Internet Things J. **9** (2022) 23260. https://doi.org/10.1109/JIOT.2022.3188224
9 T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. d. Hartog: IEEE Internet Things J. **9** (2022) 485. https://doi.org/10.1109/JIOT.2021.3085194
10 J. Benesty, J. Chen, and Y. Huang: IEEE Trans. Audio Speech Lang. Process. **16** (2008) 757. https://doi.org/10.1109/TASL.2008.919072
11 W. Mao, J. He, and M. J. Zuo: IEEE Trans. Instrum. Meas. **69** (2020) 1594. https://doi.org/10.1109/TIM.2019.2917735
12 G. M. Weiss and F. Provost: J. Artif. Intell. Res. **19** (2003) 315. https://doi.org/10.1613/jair.1199
13 F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. Mohamed Chaabani, and A. Taleb-Ahmed: 2020 IEEE Int. Conf. Internet of Things and Intelligence System (IoTaIS) (2021) 23. https://doi.org/10.1109/IoTaIS50849.2021.9359689
14 L. Breiman: Mach. Learn. **45** (2001) 5. https://doi.org/10.1023/A:1010933404324
15 H. Jiang, Z. He, G. Ye, and H. Zhang: IEEE Access **8** (2020) 58392. https://doi.org/10.1109/ACCESS.2020.2982418
16 X. Ma, J. Sha, D. Wang, Y. Yu, Q. Yang, and X. Niu: Electron. Commer. Res. Appl. **31** (2018) 24. https://doi.org/10.1016/j.elerap.2018.08.002
17 T. Wisanwanichthan and M. Thammawichai: IEEE Access **9** (2021) 138432. https://doi.org/10.1109/ACCESS.2021.3118573

About the Authors

Hong-Yu Chuang is currently a graduate student in the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taiwan, R.O.C. His research interests focus on network security, intrusion detection systems, and machine learning techniques. (sapang0626@gmail.com)

Ruey-Maw Chen received his M.S. and Ph.D. degrees in engineering science from National Cheng Kung University, Taiwan, R.O.C., in 1985 and 2000, respectively. From 1985 to 1994, he was a senior engineer in avionics system design at Chung Shan Institute of Science and Technology. From 1994 to 2001, he was an engineer at the Computer Center, Chin-Yi Institute of Technology. Since 2002, he has been with the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, where he is a full professor. His research interests include scheduling, neural networks, meta-heuristic algorithms, image processing, and computer networks. (rmchen@ncut.edu.tw)