Hybrid cyber threats detection using explainable AI in Industrial IoT

Yifan Liu, Shancang Li
School of Computer Science and Informatics
Cardiff University
Cardiff, UK

2023 International Conference on Human-Centered Cognitive Systems (HCCS) | 979-8-3503-5918-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/HCCS59561.2023.10452621

***Abstract*—The rise of new technologies have enabled the combination of multiple cyber threats to become more sophisticated and capable cyber attacks, which can result in significant losses and consequences. This work aims to develop a hybrid threats detection systems using explainable artificial intelligence techniques, preventing the growing number of combined cyber attacks in IoT environment. A multilayer hybrid cyber threats model was proposed to analyse increasingly complicated and capable cyber attacks. Meanwhile, an explainable artificial intelligence (XAI) model was designed to create understanble interpretation for the detection model. The experimental results demonstrate the effectiveness of proposed hybrid cyber threats analysis model and XAI model.**

***Index Terms*—IIoT, Deep learning, Hybrid cyber threats**

I. Introduction

Industrial Internet of Things (IIoT) takes advantage of interactive smart devices, which enable massive data transfer, real-time control, monitoring and analysis. The IIoT technology promoted the automation of manufacturing and significantly boosted production efficiency. While IIoT has brought tremendous benefits to the industrial sectors, it is facing a challenge from hybrid cyber threats (HCTs).

HCTs refer to sophisticated cyber-attacks that combine various technologies, which capitalise on the nature of IIoT system. New attack surfaces were introduced because of the interconnectivity and heterogeneity nature of IIoT [1]. For instance, attackers could compromise the IIoT system by utilizing the vulnerability of network protocols [2] and software [3]. The interconnected objects in IIoT allow attackers to invade internal networks more stealthily. For example, Stuxnet, which is widely regarded as the pioneering malware against IIoT to damage facilities physically, garnered considerable attention due to its sophistication. The attacker exploited four zero-day vulnerabilities, employing various techniques to evade detection. Researchers claimed this attack specifically targeted Siemens PLC devices and may aim to damage Iran's nuclear facilities [4].

Attacks against IIoT may cause serious ramifications such as data breach, facility damage and even casualties. Therefore, accurate threat diagnostic approach could be critical for IIoT system. During the last decades, machine learning (ML)-powered schemes have shown a compelling performance in threat detection tasks [5]–[7]. Sequence models, such as Recurrent neural network (RNN), Long-short-term memory (LSTM) and transformer, have been widely utilized in threat detection because they can extract context information from continuous data, which enables the model to detect long-term attacks [1]. Since IIoT components are designed for specific production tasks, their statements are relatively regular compared to consumption IoT [8]. The data generated by IIoT system in a normal status could have stable patterns, anomaly could be detected if the data sequence cannot match the patterns of normal status. Therefore, sequential models could be suitable for anomalies detection in IIoT.

While ML enabled threat detection mehtods shows a powerful capacity, they are usually regarded as black box approaches, which lack interpretability for security analysts. Explainable IDSs methods could be essential to boost the investigation and decision making process after the IIoT was attacked, to minimize the losses in cyber security accident.

This research mainly aims to propose a robust IDS that is empowered to recognize HCTs in IIoT. The proposed method is able to analyze heterogeneous data, recognize and classify anomalies from sequential data and provide interpretability of results. The main contributions of this work include:

- An attention mechanism-based model was proposed to improve the datasets for XAI threat detection framework;
- An SHAP (Shapley Additive Explanations) enabled explainable random forest (RF) model was developed to analyse the causes of cyber threats and attack;
- Using X-IIoT dataset, we tested the proposed models and the experimental results demonstrated the effectiveness of SHAP-RF.

II. Related Works

Sequential models, such as RNN, LSTM, 1D-CNN was designed to extract contextual information from continous data, has been obtained exceptional performance in threat detection tasks [9]–[11]. However, previous sequential modal have some limitations. For example, gradient vanishing problem [12] and forgetting long-term information [13]. The structure nature of recurrent models led to relatively lower training efficiency [14]. Transformer model relies on attention mechanism enables paralleled computation in long sequences analysis [13]. It has achieved dramatic performance in NLP domain, and gradually interested cyber security researchers. Wu *et al.* proposed a method called RTIDS to classify network traffic, which is based on transformer model [13]. The proposed method achieved 99.17% and 98.48% F1-Score on

CICIDS2017 and CIC-DDoS2019 dataset respectively. [15] proposed a transformer-based IIoT network anomaly detection model. This work was based on the concept that a pre-trained transformer model on normal traffic data tends to have higher loss values when confronted with abnormal traffic samples. This model achieved 97.4% accuracy on the WUSTL-IIoT dataset.

The increasingly complex network attack-generated traffic has become challenging to accurately model threats using a single classifier. Many researchers have utilised ensemble learning to enhance the performance of intrusion detection. [16] proposed a federated DL method to mitigate the negative impact of heterogeneous data in IIoT environment. This method achieved 20% improvement of F1-score compared to centralised model. Yazdinejad *et al.* constructed an ensemble model which combined LSTM with auto-encoders to identify the anomalies in imbalanced data, and obtained an accuracy of 99.7% [17]. Ullah *et al.* proposed a transformer based IDS to analyse network traffic data in detail from PCAP files [18]. BERT was used to extract features from the decoded PCAP files. CNN layers was deployed to reduce the dimension of features. LSTM layers was functioned for analyzing the semantic information of the sequence data. The result shows that the IDS-INT achieved 99% precision on NSL-KDD, CIC-IDS2017 and UNSW-NB15 datasets.

Although the current IDSs achieved dramatic performance, the existing approaches may be inadequate for HCTs detection in IIoT environment. There are several problems that need to be addressed:

- Datasource: Most research generally relies on a single data source, either based on network flow or host logs. Nevertheless, in real IIoT environments, the lifecycle of attacks usually involves a spectrum of techniques and diverse targets from different IIoT components [19]. Single data source lacks a comprehensive perspective and may omit potential attack patterns [20].
- Adaptability: Current studies usually recognize attacks according to patterns of existing datasets. However, some new attacks may be misclassified, they may be simply treated as false positives in investigations.
- Interpretability: Previous works have tended to focus on sample classification, which lacks semantic interpretability [21].

III. Methodology of Proposed Scheme

This section will introduce models employed in this scheme. Then briefly explain the basic concept of the proposed method.

A. Modelling

1) Attention Mechanism: The attention model can help neural networks focus on different parts of input data with varying levels of importance [14], which allows the model to focus more on specific threats or attacks in cyber security. The definition of attention is given by Eq. (1).

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

in which three matrix, Q, K, and V, denote *query, key* and *values* respectively, are generated by linear transformations of input sequence x, namely, matrix multiplication with weight matrices W^Q, W^K, W^V separately. The weight matrices are initialized randomly and are updated by gradient descent algorithm in training process.

$$Q = W^Q \cdot x \tag{2}$$

$$K = W^K \cdot x \tag{3}$$

$$V = W^V \cdot x \tag{4}$$

2) Positional Embedding: Compared to RNNs, attention mechanism itself is not able to extract positional information of sequences. Therefore, positional embedding is required to indicate the sequential difference of input samples. In this paper Eqs. (5) and (6) are used to encode position and **concat** to the sample, in which Eq. (5) will encode the **odd** position, (6) encode the **even** position.

$$PE_{(pos,2i)} = \sin(\frac{posision}{1000^{\frac{2i}{d_{model}}}}) \tag{5}$$

$$PE_{(pos,2i+1)} = \cos(\frac{posision}{1000^{\frac{2i}{d_{model}}}}) \tag{6}$$

3) Single-Head Attention Model: The attention model utilized in this paper is based on the transformer, which is a multi-head attention model proposed by Google [14]. Compared to the NLP problem, the feature dimension in anomaly detection tasks is significantly lower [15]. Therefore, single-head attention was utilized in this study. The constructed attention-based anomaly recognition model is shown in Fig. 1. The attention model uses Encoder-Decoder architecture, both Encoder and Decoder include N folded encoders and decoders. The *Encoder* includes a *single-head attention* and a *feed-forward network.* Single-head attention is attention combined with a fully connected layer. It is represented by Eq. (7).

$$Singlehead(x) = Linear(Attention(x)) \tag{7}$$

The *single-head attention* can identify the relationships of the input and the *feed-forward network* is employed to improve the model capacity of feature abstraction.

Encoder layers are used to convert the input sequence to a matrix which includes context information. The result of the encoder will multiply the W^Q and W^K and be employed by the decoder. The decoder in this scheme includes two single-head attention and a Feed Forward module. It should be noted that the first attention layer employs a masked process, which is used to prevent the decoder from obtaining the information it needs to predict, thereby avoiding over-fitting. The masked attention is calculated with Eq. (8), where M is a mask matrix.

$$Attention(Q,K,V) = softmax(M + \frac{QK^T}{\sqrt{d_k}})V \quad (8)$$

$$M_{ij} = \begin{cases} 0 & allow\ to\ attened \\ -\infty & prevent\ from\ attending \end{cases} \quad (9)$$

Inputs of the decoder are shifted outputs of its previous prediction. The masked single-head attention will give a prediction according to former outputs. The single-head attention will consider the context information from the results of encoder, and then predict the next output. In this work, we set the number of encoder and decoder layers N is 6. Residual addition and batch normalization are carried out following each step in both the encoding and decoding layers. The output of each operation is

$$Sublayer(x) = Norm(SubLayer(x) + x) \quad (10)$$

where $Sublayer$ refers to a single encoder or decoder layer, x is the input of a layer.

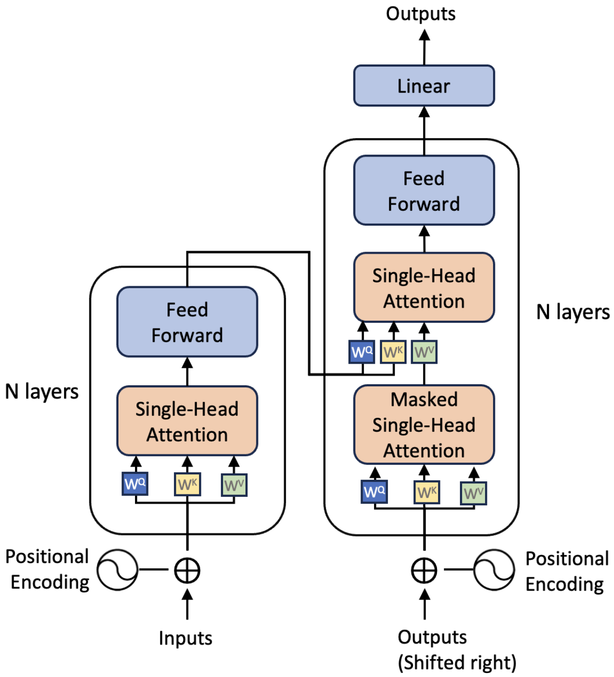

Fig. 1. Attention Model

4) Decision Tree: A random forest (RF) combines multiple decision trees (DTs) that are independent of the others.

The classification and regression tree (CART) employed in this paper, is one of the algorithms of decision tree. Its feature selection is based on Gini impurity, which represents the probability that a randomly chosen sample from a dataset will be incorrectly classified. It is a metric that guides tree establishment by dividing nodes according to features. The calculation of Gini impurity can be obtained by Eq. (11)

$$Gini(D) = 1 - \sum_{i=1}^{n} p(x_i)^2 \quad (11)$$

where $p(x_i)$ indicates the probability of the occurrence of class x_i, n refers to the number of classes. $Gini(D)$ represents the probability of randomly selecting two samples from the dataset D that have inconsistent class labels.

For a feature a, which may have V different values $a_1, a_2, \ldots, a_V$, there will be V nodes if employ a to split D. The branch node v involves samples D_v whose $a = a_v$. The definition of its $Giniindex$ is as

$$Gini_{index}(D,a) = \sum_{v=1}^{V} \frac{|D_v|}{D} Gini(D_v) \quad (12)$$

For feature selection in feature set A, the selected a will be

$$a_* = \arg_{a \in A} \min Gini_{index}(D,a) \quad (13)$$

5) Random Forest: RF is an improvement based on bagged DT, which is assembled by bagging algorithm, to accomplish tasks by combining multiple models. The bagging algorithm is as follows:

Algorithm 1 Bagging Algorithm

```
Input Dataset $D = (x_1,y_1),(x_2,y_2),\ldots,(x_m,ym)$;
      Basic model $DT$;
      Training epoch $T$;
Output $H(x) = \arg_{y \in Y} max \sum_{t=1}^{T} \mathbb{I}(h_t(x) = y)$;
for $t \le T$ do
    $h_t = DT(D, D_{bs})$;
end for
```

RF is employed in this project because of the following benefits:

- It is less prone to overfitting and learns correlations between features.
- It can process high-dimensional data without feature selection.
- It can mitigate the impact of imbalanced data.

B. SHAP

SHAP is an explanation method based on game theory to describe machine learning model. It utilizes Shapley values to combine local explanations and credit allocation [22]. This project employed SHAP to describe how the features impact the result of the model. As shown in Fig. 3, a single picture is the heat map for feature contributions of a specific class in model classification. The $x-axis$ is SHAP value, it indicates the impact on model decision of a feature. The color represents the numerical scales of features.

C. Proposed Solution

To address above problem, we proposed a new three-phase solution: data processing, anomaly detection, and attack classification.

1) Data processing: For a given dataset D, it was sorted by timestamp and partitioned into two parts: The first part D_1 contains continuous normal samples, and the second part D_2 is remaining data which includes samples of both normal and various attacks.

The data used for the single-head attention model is standardized using the method given in Eq. (14), where x_i is the sample vector, d_{size} is the size of processed dataset.

$$z_i = \frac{(x_i - \mu)}{\sigma} \tag{14}$$

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{d_{size}}} \tag{15}$$

$$\mu = \frac{\sum(x_i)}{d_{size}} \tag{16}$$

2) Anomaly detection: There are two steps for anomaly detection. Firstly, The single-head attention model was trained using D_1 to learn data patterns of normal status. Then, the pre-trained model was employed to process the second part data to predict the anomalies. The result was correlated to the D_2 by sample and assembled as a new feature. The data assembling process is as follows:

Algorithm 2 Anomalies Prediction and Data Assembling

```
Input Dataset $D_2$;
      Pre-trained Attention Model $\varphi_{attention}$;
Output ensemble dataset $D_{ensemble}$;
Sort $D$ by $Timestamp$;
Init empty array $arr$;
for sample $x_i$ in $D_2$ do
    $ano\_score_i = \varphi(x_i)$;
    $arr$ append $ano\_score_i$;
end for
$D_2$ concat $arr$ with column name $ano\_score$;
```

3) Attack classification and model explanation: The assembled data was further split into training and testing sets, which are prepared for training and evaluating RF model. Additionally, SHAP is deployed to provide interpretability.

The structure of proposed scheme is shown in Fig. 2, which includes two main phases: *anomaly detection* and *XAI enabled threat classifier*. In anomaly detection phases, a pre-trained attention model is used to detect the anomaly possibilities in pre-processed data, which involve information from network traffic and system logs. The detection results will be fed into the threat classification in the second phase, which includes a RF classifier and a SHAP-based explainer that can provdie detailed explanation to highlight the causes, contributed features, to a specific threat.

IV. Experiment Validation

A. Experiment Settings

The proposed method was deployed on a server with *Intel Core i7-12700 processor*, *32GB RAM*, *Nvidia RTX 3080 GPU* and *Ubuntu 22.04*. Both the attention model and RF classification model depend on *python 3.9.0*, *PyTorch 1.12.0*, and *CUDA 12*. The detailed validation procedures inlcude following three main stages.

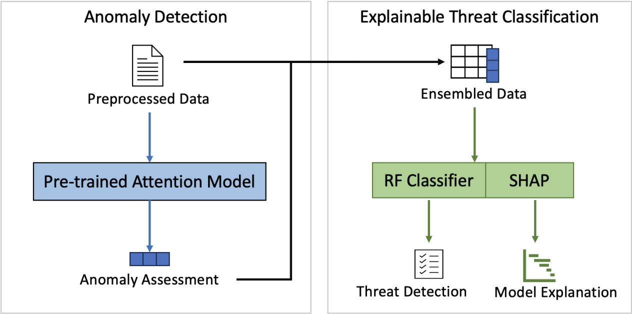

Fig. 2. Attention Modeling Training Process

1) Data Processing: To validate the proposed scheme, we use the X-IIoTID dataset to test the proposed model. The X-IIoTID consists of features extracted from network traffic, system logs, and rule-based IDS alerts. A summary of the X-IIoT dataset is shown in Table I.

TABLE I
Summery of the X-IIoTID Dataset

Label	Count	Ratio
Normal	421417	51.34%
Reconnaissance	127590	17.20%
RDoS	141261	15.54%
Weaponization	67260	8.19%
Lateral Movement	31596	3.85%
Exfiltration	22134	2.70%
Exploitation	1133	0.62%
Tampering	5122	0.35%
C & C	2863	0.14%
Crypto Ransomware	458	0.06%
Total	820834	100%

In the attention model training process, we first removed invalid data in the X-IIoTID dataset and then sorted by timestamp to well match the time-sequential attention model. The two columns *Date, Scr_IP and Des_IP* were dropped to avoid overfitting. Since port number can indicate the service, the feature *Service* may be redundant. Table II shows the selected features as sourced with *Network* and *Host*.

2) Anomalies detection training: In this work, we selected part of normal samples for Single-Head Attention model training, and the model is trained with 1000 epochs, 1024 batch size. An adam optimizer is applied with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 10^{-9}$, and the learning rate is $l = 0.0001$.

The trained Single-Head Attention model can be used to predict the anomalies of the dataset, and the prediction result ano_score is presented as a new feature in Table. II.

3) Threat classification training: The ensemble data $D_{ensemble}$ was divided into 80% training set and 20% test

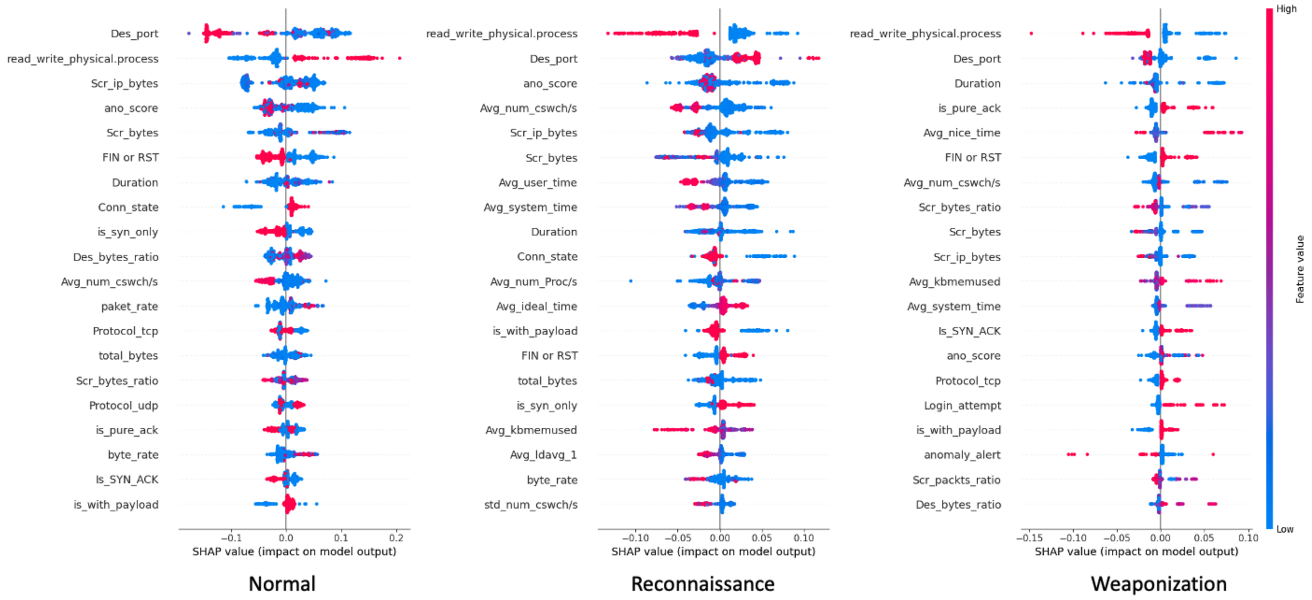

Fig. 3. RF Explanation with SHAP

TABLE II
SELECTED FEATURES

Source	Type	Features
Network	Continuous	Duration, Scr_bytes, Des_bytes, Missed_btye, Scr_pkts, Des_pkts, Scr_IP_bytes, Dec_IP_bytes, Total_bytes, Byte_rate, Total_Pkts, Pkts_rate, orig_bytes_ratio, resp_bytes_ratio, orig_packts_ratio, resp_pkts_ratio
	Discrete	Scr_Port, Des_Port, Protocol, Conn_state, SYN, SYN-ACK, Pure ACK, Packet with payload, FIN or RST, Bad checksum, SYN with RST
Host	Continuous	Avg_user_time, Std_user_time, Avg_nice_time, Std_nice_time, Avg_system_time, Std_system_time, Avg_IO_wait_time, Std_IO_wait_time, Avg_idle_time, Std_idle_time, Avg_tps, Std_tips, Avg_rtps, Std_rtps, Avg_wtps, Std_wtps, Avg_ldavg_1, Std_ldavg_1, Avg_Kbmemused, Std_Kbmemused, Avg_num_proc/s, Std_num_proc/s, Avg_num_swch/s, Std_num_swch/s
	Discrete	Anomaly_Alert, OSSEC_alert, Alert_level, R_W_physical, File_act, Proc_act, Is_priviliged, Login_attemp, Succ_login
Generated	Continuous	ano_score

set. The features summary of $D_{ensemble}$ are shown in Table II.

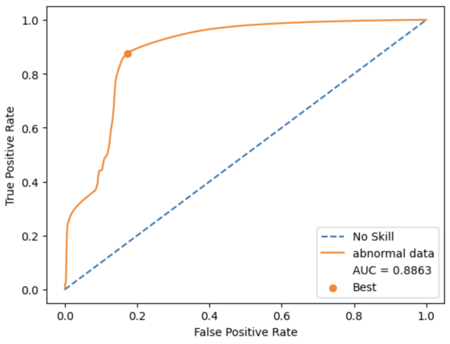

Fig. 4. ROC of Attention Model

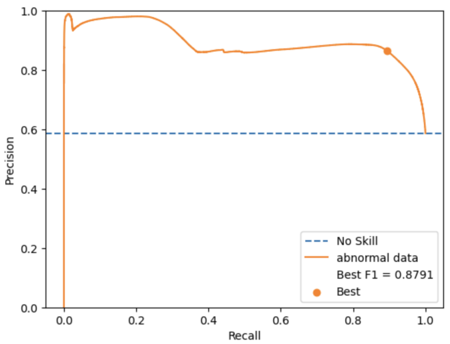

Fig. 5. PR of Attention Model

4) Evaluation Metrics: We use *Precision, Accuracy, Recall and F1-score* to evaluate the performance of the proposed scheme and compare it with the based-line, as described in following Eqs. (17) - (20), in which TP denotes true positive, FP denotes false positive, FP denotes false positive, and TN denotes true negative, respectively.

$$Pre = \frac{TP}{TP + FP} \tag{17}$$

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{18}$$

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{20}$$

B. Results

Fig. 4 and Fig. 5 show the ROC and PR curves of the trained attention model. Table. III presents the performance of the proposed classifier in terms of precision, accuracy, recall rate, and F_1 score. It is noticed that the proposed scheme achieved a certain improvement compared to the RF, and significantly better than baseline.

TABLE III
MODEL EVALUATION

Model	Precision	Accuracy	Recall	F1-score
Baseline [19]	97.3300%	99.4900%	97.2000%	97.2700%
RF	99.7219%	99.7213%	99.7213%	99.7207%
Attention + RF	99.7477%	99.7472%	99.7472%	99.7467%

Regarding the anomaly detection in X-IIoTID set, 500 samples were selected and SHAP explainer was used to generate causes analysis report. SHAP values of labels were used to indicate the impact on model output. As an example, Fig. 3 shows three $\{$*normal, reconnaissance, weaponisation*$\}$ of ten as example and we found that the new feature ano_score have a significant contribution to RF classification.

V. CONCLUSION

In this paper, a robust HCT detection framework has been proposed to recognize and classify malicious activities in IIoT environment. This scheme utilizes heterogeneous data sources including network flow and system logs to construct an attention mechanism-based model to assess the data status in the system. An RF-based classifier is deployed to detect the anomalies from data with the bias of the former assessment. In addition, SHAP was deployed to improve interpretability of the detection framework by visualizing the feature importance of the RF model. However, there are some limitations in this work. The dataset employed in this work is preprocessed, it focuses on heterogeneity of data sources and may lack sufficient consideration of temporality. Since multi-head attention models allow parallel computation and accommodate higher data dimensions, future works may involve preserving more original information when converting raw time sequence data into sample vectors, and employing a multi-head attention mechanism for anomaly detection.

REFERENCES

[1] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for internet of things (iot) security," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1646–1685, 2020.

[2] A. S. Mohammed, E. Anthi, O. Rana, N. Saxena, and P. Burnap, "Detection and mitigation of field flooding attacks on oil and gas critical infrastructure communication," *Computers & Security*, vol. 124, p. 103007, 2023.

[3] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine learning-based network vulnerability analysis of industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6822–6834, 2019.

[4] T. M. Chen and S. Abu-Nimeh, "Lessons from stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, 2011.

[5] S. Latif, Z. Idrees, Z. Zou, and J. Ahmad, "Drann: A deep random neural network model for intrusion detection in industrial iot," in *2020 International Conference on UK-China Emerging Technologies (UCET)*, 2020, pp. 1–4.

[6] S. Aftab, Z. S. Shah, S. A. Memon, and Q. Shaikh, "A machine-learning-based intrusion detection for iiot infrastructure," in *2023 7th International Multi-Topic ICT Conference (IMTIC)*, 2023, pp. 1–6.

[7] S. Li, G. Chai, Y. Wang, G. Zhou, Z. Li, D. Yu, and R. Gao, "Crsf: An intrusion detection framework for industrial internet of things based on pretrained cnn2d-rnn and svm," *IEEE Access*, vol. 11, pp. 92 041–92 054, 2023.

[8] S. V. B. Rakas, M. D. Stojanović, and J. D. Marković-Petrović, "A review of research work on network-based scada intrusion detection systems," *IEEE Access*, vol. 8, pp. 93 083–93 108, 2020.

[9] T.-T.-H. Le, J. Kim, and H. Kim, "An effective intrusion detection classifier using long short-term memory with gradient descent optimization," in *2017 International Conference on Platform Technology and Service (PlatCon)*, 2017, pp. 1–6.

[10] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1222–1228.

[11] S. A. Althubiti, E. M. Jones, and K. Roy, "Lstm for anomaly-based network intrusion detection," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*, 2018, pp. 1–3.

[12] T. Pooja and P. Shrinivasacharya, "Evaluating neural networks using bi-directional lstm for network ids (intrusion detection systems) in cyber security," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 448–454, 2021.

[13] Z. Wu, H. Zhang, P. Wang, and Z. Sun, "Rtids: A robust transformer-based approach for intrusion detection system," *IEEE Access*, vol. 10, pp. 64 375–64 387, 2022.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] J. Casajús-Setién, C. Bielza, and P. Larrañaga, "Anomaly-based intrusion detection in iiot networks using transformer models," in *2023 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2023, pp. 72–77.

[16] O. Belarbi, T. Spyridopoulos, E. Anthi, I. Mavromatis, P. Carnelli, and A. Khan, "Federated intrusion detection system based on deep belief networks," *arXiv preprint arXiv:2306.02715*, 2023.

[17] A. Yazdinejad, M. Kazemi, R. M. Parizi, A. Dehghantanha, and H. Karimipour, "An ensemble deep learning model for cyber threat hunting in industrial internet of things," *Digital Communications and Networks*, vol. 9, no. 1, pp. 101–110, 2023.

[18] F. Ullah, S. Ullah, G. Srivastava, and J. C.-W. Lin, "Ids-int: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," *Digital Communications and Networks*, 2023.

[19] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-iiotid: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3962–3977, 2022.

[20] 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. 165 130–165 150, 2020.

[21] D. L. Marino, C. S. Wickramasinghe, C. Rieger, and M. Manic, "Self-supervised and interpretable anomaly detection using network transformers," *arXiv preprint arXiv:2202.12997*, 2022.

[22] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature Machine Intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.