Hybrid cyber threats detection using explainable AI in Industrial IoT

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

Abstract—The rise of new technologies has 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 threat detection system using explainable artificial intelligence techniques, preventing the growing number of combined cyber-attacks in IoT environments. A multilayer hybrid cyber threats model was proposed to analyse increasingly complicated and capable cyber attacks. In addition, an explainable artificial intelligence (XAI) model was designed to create an understandable interpretation for the detection model. The experimental results demonstrate the effectiveness of the proposed hybrid cyber threats analysis model and XAI model.

Index Terms—IoT, 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, widely regarded as the pioneering malware against IIoT, attacked to minimize the losses in cyber security accidents.

This research mainly aims to propose a robust IDS 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 attacks;
- Using the 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, and 1D-CNN, were designed to extract contextual information from continuous data, have obtained exceptional performance in threat detection tasks [9]–[11]. However, previous sequential models have some limitations. For example, gradient vanishing problem [12] and forgetting long-term information [13]. The structure of recurrent models led to relatively lower training efficiency [14]. Transformer model relies on attention mechanism that enables paralleled computation in long sequences analysis [13]. It has achieved dramatic performance in natural language processing and gradually interested cyber security researchers. Wu et al. proposed a method called RTIDS to classify network traffic based on transformer model [13]. The proposed method achieved 99.17% and 98.48% F1-Score.
on CICIDS2017 and CIC-DDoS2019 datasets respectively. A transformer-based IIoT network anomaly detection model has been proposed in [15]. 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 model threats using a single classifier. Many researchers have utilised ensemble learning to enhance the performance of intrusion detection. Belarbi et al. proposed a federated DL method to mitigate the negative impact of heterogeneous data in IIoT environment. This method achieved a 20% improvement in F1-score compared to centralised model [16]. 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 were deployed to reduce the dimension of features. LSTM layers were 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 HCT detection in IIoT environment. Several problems need to be addressed:

- Datasource: Most research relies on a single data source 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 and regarded 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).

\[
\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right)V
\]

(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.

\[
Q = W^Q \cdot x
\]

(2)

\[
K = W^K \cdot x
\]

(3)

\[
V = W^V \cdot x
\]

(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 \text{concat} to the sample, in which Eq. (5) will encode the \text{odd} position, (6) encode the \text{even} position.

\[
P E_{(\text{pos},2i)} = \sin \left( \frac{\text{pos} \cdot 10000^d_{\text{model}}} {10000^d_{\text{model}}} \right)
\]

(5)

\[
P E_{(\text{pos},2i+1)} = \cos \left( \frac{\text{pos} \cdot 10000^d_{\text{model}}} {10000^d_{\text{model}}} \right)
\]

(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 \( N \) fold 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).

\[
\text{Singlehead}(x) = \text{Linear}(\text{Attention}(x))
\]

(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.
\[ Attention(Q, K, V) = \text{softmax}(M + \frac{QK^T}{\sqrt{d_k}})V \]  
\[ M_{ij} = \begin{cases} 0 & \text{allow to attended} \\ -\infty & \text{prevent from attending} \end{cases} \]  

Inputs of the decoder are shifted outputs of its previous prediction. The masked single-head attention will give a prediction according to former outputs. In this work, we set the number of encoder and decoder layers \( N \) to 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

\[ \text{Sublayer}(x) = \text{Norm}(\text{SubLayer}(x) + x) \]  

where \( \text{Sublayer} \) refers to a single encoder or decoder layer, \( x \) is the input of a layer.

---

4) Decision Tree: A random forest (RF) combines multiple decision trees (DTs) independent of the others. The classification and regression tree (CART) algorithm is employed in this paper. 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)

\[ Gini(D) = 1 - \sum_{i=1}^{n} p(x_i)^2 \]  

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 with inconsistent class labels from the dataset \( D \).

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 \( Gini_{index} \) is as

\[ Gini_{index}(D, a) = \sum_{v=1}^{V} \left| \frac{D_v}{D} \right| Gini(D_v) \]  

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

\[ a_* = \arg_{a \in A} \min Gini_{index}(D, a) \]  

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

\[ \text{Algorithm 1 Bagging Algorithm} \]

**Input** Dataset \( D = (x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m) \);
Basic model \( DT \);
Training epoch \( T \);

**Output** \( H(x) = \arg_{y \in Y} \max \sum_{t=1}^{T} I(h_t(x) = y) \);

1: for \( t \leq T \) do
2: \( h_t = DT(D, D_{bs}) \);
3: 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, indicating the impact on model decision of a feature. The colour 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.

\[
    z_i = \frac{(x_i - \mu)}{\sigma} \quad (14)
\]

\[
    \sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{d_{size}}} \quad (15)
\]

\[
    \mu = \frac{\sum (x_i)}{d_{size}} \quad (16)
\]

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 of the 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 $D_2$:

**Output** ensemble dataset $D_{ensemble}$:

1. Sort $D$ by Timestamp;
2. Init empty array $arr$;
3. for sample $x_i$ in $D_2$ do
4.  $ano \_score_i = \varphi(x_i)$;
5.  $arr$ append $ano \_score_i$;
6. end for
7. $D_2$ 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 the 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 an RF classifier and a SHAP-based explainer that can provide detailed explanations to highlight the causes and 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 include the following three main stages.

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.

![Fig. 2. Attention Modeling Training Process](image)

**Table I** SUMMARY OF THE X-IIoTID DATASET

<table>
<thead>
<tr>
<th>Label</th>
<th>Count</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>421417</td>
<td>51.34%</td>
</tr>
<tr>
<td>Reconnaissance</td>
<td>127590</td>
<td>17.20%</td>
</tr>
<tr>
<td>RDoS</td>
<td>141261</td>
<td>15.54%</td>
</tr>
<tr>
<td>Weaponization</td>
<td>67260</td>
<td>8.19%</td>
</tr>
<tr>
<td>Lateral Movement</td>
<td>31596</td>
<td>3.85%</td>
</tr>
<tr>
<td>Exfiltration</td>
<td>22134</td>
<td>2.70%</td>
</tr>
<tr>
<td>Exploitation</td>
<td>1133</td>
<td>0.62%</td>
</tr>
<tr>
<td>Tampering</td>
<td>5122</td>
<td>0.35%</td>
</tr>
<tr>
<td>C &amp; C</td>
<td>2863</td>
<td>0.14%</td>
</tr>
<tr>
<td>Crypto Ransomware</td>
<td>458</td>
<td>0.06%</td>
</tr>
</tbody>
</table>

Total 820834 100%

In the attention model training process, invalid data were removed in the X-IIoTID dataset and then sorted by timestamp to match the time-sequential attention model well. Two columns Date, Scr_IP and Des_IP were dropped to avoid overfitting. Since the 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

<table>
<thead>
<tr>
<th>Source</th>
<th>Type</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Continuous</td>
<td>Duration, Scr_bytes, Des_bytes, Missed_byt, 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_packets_ratio, resp_packets_ratio</td>
</tr>
<tr>
<td></td>
<td>Discrete</td>
<td>Scr_Port, Des_Port, Protocol, Conn_state, SYN, SYN-ACK, Pure ACK, Packet with payload, FIN or RST, Bad checksum, SYN with RST</td>
</tr>
<tr>
<td>Host</td>
<td>Continuous</td>
<td>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_cpu, Std_cpu, Avg_rtps, Std_rtps, Avg_wtps, Std_wtps, Avg_lavg_1, Std_lavg_1, Avg_Kmemused, Std_Kmemused, Avg_num_proc, Std_num_proc, Avg_num_swap, Std_num_swap,</td>
</tr>
<tr>
<td></td>
<td>Discrete</td>
<td>Anomaly_alert, osSec_alert, Alert_level, R_W_physical, File_act, Proc_act, Is_priviliged, Login_attempts, Succ_login</td>
</tr>
<tr>
<td>Generated</td>
<td>Continuous</td>
<td>ano_score</td>
</tr>
</tbody>
</table>

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

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 the following Eqs. (17) - (20), in which $TP$ denotes true positive, $FP$ denotes false positive, $FP$ denotes false positive, and $TN$ denotes true negative, respectively.

$$Pre = \frac{TP}{TP + FP}$$ (17)
Regarding the anomaly detection in X-IoTID set, 500 samples were selected, and SHAP explainer was used to generate a causes analysis report. SHAP values of labels were used to indicate the impact on model output. As an example, Fig. 3 shows results of three labels (normal, reconnaissance and weaponisation) as examples, and we found that the new feature ano_core has 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 the 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, focuses on the 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


(continued...)

TABLE III

MODEL EVALUATION

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Accuracy</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [19]</td>
<td>97.3300%</td>
<td>99.4900%</td>
<td>97.2000%</td>
<td>97.2700%</td>
</tr>
<tr>
<td>RF</td>
<td>99.7219%</td>
<td>99.7213%</td>
<td>99.7213%</td>
<td>99.7207%</td>
</tr>
<tr>
<td>Attention + RF</td>
<td>99.7477%</td>
<td>99.7472%</td>
<td>99.7472%</td>
<td>99.7467%</td>
</tr>
</tbody>
</table>

F1 = 2 × Precision × Recall / Precision + Recall

Acc = (TP + TN) / (TP + FP + TN + FN)

Recall = TP / (TP + FN)

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 F1 score. It is noticeable that the proposed scheme achieved a certain improvement compared to the RF and was significantly better than the baseline.