Empowering Adaptive Endogenous Security Trend Prediction Detection for IoT Sensor Nodes

1st Lihu Zhou
School of Computer Science and Software Engineering
Southwest Petroleum University
Chengdu, China
zhoulihu2000@163.com

2nd Xiuying Dong
School of Computer Science and Software Engineering
Southwest Petroleum University
Chengdu, China
dongxiuying20020306@outlook.com

3rd Enli Zhang
Exploration and Development Research
Institute of PetroChina Southwest Oil & Gasfield Company
Chengdu, China
zhangenli@petrochina.com.cn

4th Ting Wang
School of Computer Science
Chengdu University of Information Technology
Chengdu, China
wangting@cuit.edu.cn

5th Xiao Zhang
School of Computer Science and Software Engineering
Southwest Petroleum University
Chengdu, China
17781537802@163.com

6th Chong Zhang
School of Computer Science and Software Engineering
Southwest Petroleum University
Chengdu, China
zhangchong92@swpu.edu.cn

***Abstract*—Massively deployed IoT devices require a lightweight design to reduce costs. However, this architecture inherently limits their security, increasing the risk of data breaches and tampering on a large scale. In this paper, we propose CGAD, a model that leverages one-dimensional convolution (Conv1D), gated recurrent units (GRU), and the attention mechanism to analyze multi-sensor time series data from sensing terminals at the data gateway. To achieve this, we generate time series predictions and detect anomalies by comparing predicted values with real data using a threshold-based approach, thereby enhancing data security. To address the labeling problem of massive data, we employ unsupervised learning and design a dynamic sliding window threshold to improve the judgment of time series with varying characteristics. We also evaluate our system using the KW51 railway bridge dataset and a self-collected temperature and humidity dataset. The results demonstrate that our CGAD model achieves recall rates of 98% and 99% for anomaly detection on the two datasets, respectively. Additionally, the precision and F1 scores surpass those of other methods, ensuring cost-effectiveness for massive terminals and addressing the security issues of sensitive data. These advantages contribute to realizing the Internet of Everything.**

***Index Terms*—Internet of Things, anomaly detection, Gate Recurrent Unit, time series forecast**

I. Introduction

The Internet of Everything (IoE), an extension of the Internet of Things (IoT) paradigm, holds immense potential in driving the future of Industry 4.0, smart homes, cities, agriculture, and more [1]. To enable seamless interconnectivity and advance IoT applications, it is crucial to gather comprehensive environmental sensing data. Deploying a large network of devices helps gain a deeper, more holistic understanding of the physical world. Furthermore,addressing data security concerns for the many distributed sensing nodes is crucial to maintaining the reliability and smooth operation of IoT applications.

However, to realize the vision of the Internet of Everything (IoE), reducing post-maintenance costs for the vast number of sensing nodes is essential. Many devices operate for long periods of time on just a small battery (avoiding frequent battery replacements and increased maintenance costs later on). Consequently, certain devices have integrated ultra-low-power backscatter communication technologies [2], which minimize computational tasks at the terminal level or maintain sensing nodes in a dormant state for extended periods. In such scenarios, it is unaffordable for them to do high performance security detection. In the end, they can only sacrifice data security to reduce the power consumption of the sensing nodes [3], which in turn reduces the maintenance cost at a later stage.

To reduce post-maintenance costs of sensing nodes while ensuring data security, various strategies have been proposed. One approach integrates security policies directly at the terminal level, relying on microprocessors for tasks like encryption and decryption, but it fails to significantly lower power consumption and maintenance costs [4]. Another approach focuses on unsupervised machine learning [5] for the detection of sensing time series, and although they work well in unlabelled multiple time series anomaly detection, they are very weak against noise and have a single threshold for judging anomalies, which does not allow for true and accurate anomaly judgments. Although the above methods have made

some progress in terms of low cost and data security of IoT, they still cannot meet the demand of low maintenance cost and high data security of the Internet of Everything.

In this paper, we propose CGAD (Conv1D, GRU, Attention, Anomaly Detector), a model architecture for near-zero power consumption on the node side and intelligent on the gateway side for endogenous security detection based on the adaptive future trend of sensor data change. The architecture is illustrated in Fig. 1. Sensor terminals communicate with the gateway using backscattering technology [2], enabling near-zero power consumption at the micro-watt level. The gateway processes sensor node data using one-dimensional convolution and GRU to analyze future trends. It then generates predictions of data trends and compares the actual data with these predicted values. This comparison allows the system to automatically detect anomalies, thereby achieving endogenous security detection. CGAD significantly reduces the maintenance costs of sensor terminals while ensuring the security of sensor data.

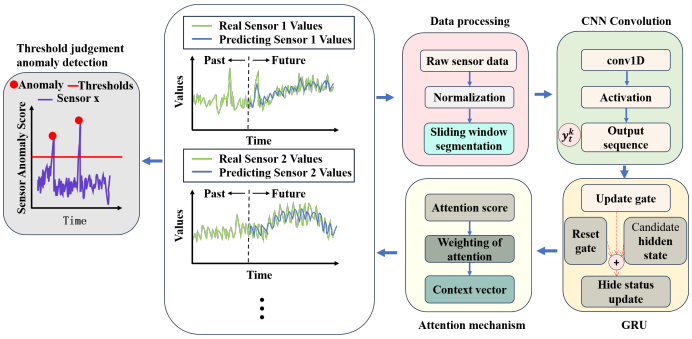

Fig. 1. The CGAD composition model network architecture.

Challenges and contributions. Our system design faces three main challenges, as follows:

1) In complex natural environments, sensor data is often subject to various forms of noise, which can degrade the accuracy of subsequent predictions. This degradation increases the risk of misjudgment in anomaly detection.

To address this issue, we introduce an attention mechanism that allows the model to focus on relevant features, mitigating noise and enhancing robustness. We also integrate an early stopping mechanism to further reduce the risk of overfitting.

2) Different sensing terminal time series require consideration of sensor characteristics and application scenarios. Temperature and humidity sensors may show gradual changes, while acceleration sensors can fluctuate rapidly. Therefore, a single average anomaly threshold is inadequate for accurate anomaly detection across these diverse time series.

To address this issue, we propose employing a sliding window threshold for anomaly detection. By adjusting the window size to accommodate the specific characteristics of different sensing terminal time series, we can establish a more accurate threshold for anomaly discrimination. This approach enhances the precision of anomaly detection.

3) Since this paper uses an unsupervised model, the final anomaly detection judgement needs to be achieved by certain criteria or mechanisms.

To address this issue, we designed an anomaly detector that calculates the anomaly score between predicted and actual values, enabling adaptive detection via a dynamic threshold.

We implemented CGAD on a personal computer and tested it on various datasets, including the KW51 railway bridge dataset and a self-collected temperature and humidity dataset. Comprehensive comparison and ablation experiments were conducted, employing precision, recall, and F1 score as evaluation metrics. The results demonstrate that CGAD achieves anomaly detection rates of 98% to 99%, with precision and F1 scores exceeding those of competing methods. These findings indicate that CGAD effectively reduces deployment and maintenance costs of sensing terminals while ensuring data security, thereby advancing the Internet of Everything. The main contributions of this work are as follows:

- We propose the CGAD model, which can reduce the post-maintenance cost of sensing terminals, while improving data security and facilitating the large-scale application of sensing IoT.
- We enhance system robustness by integrating attention and early stopping, effectively addressing noise and overfitting in complex sensor data.
- We designed a sliding-window threshold discrimination mechanism for endogenous safety monitoring of time series prediction, which can effectively solve the problem of misjudgement of sensor time series with different characteristics due to the threshold problem.
- We designed an anomaly detector for unsupervised learning models to calculate the anomaly score between the predicted and true values and to self determine if there are anomalies in the sensed data.

II. Related Work

In order to reduce the cost of IoT post maintenance while ensuring data security, researchers have proposed rich research in two technical directions as shown below:

A. Integrated Deployment of Security Policies on the Sensing End-Side

Power consumption and security pose significant challenges in sensor networks, often conflicting with each other. As security measures become more complex, battery consumption increases. Data encryption and related operations, such as key exchange, place demands on power that lightweight devices may struggle to meet.

To mitigate these issues, OA Khashan and Wanli Xue et al. have proposed lightweight low-power encryption algorithms [6] and [7], which aim to ensure data security while reducing power consumption. Nevertheless, these algorithms fall short of achieving near-zero power consumption, resulting in sustained maintenance costs.

Furthermore, the inherent characteristics of sensor communication and dynamic network topologies make traditional encryption protocols insufficient. In response, researchers have

explored security authentication and trust mechanisms [8], which improve data security but significantly increase power consumption in sensing terminals.

B. Data Anomaly Detection Based on Machine/Deep Learning Models

In recent years, machine learning and deep learning techniques have gained traction in IoT data anomaly detection, broadly classified into supervised and unsupervised learning.

In supervised learning, Munwar Ali et al. employed the k-Nearest Neighbor (k-NN) method for anomaly detection [9]. Despite enhancements to the traditional k-NN algorithm, it remains burdened by high computational complexity and insensitivity to outliers, especially with large training datasets. Similarly, Alan Colman et al. utilized the Decision Tree (DT) method [10], a predictive modeling approach that combines statistics, data mining, and machine learning. Their experiments showed significant improvements in detection accuracy; however, the training process requires substantial computational resources, complicating real-time applications.

Conversely, unsupervised learning effectively addresses the labeling challenges associated with large volumes of sensing data in supervised approaches. Chunyong Yin proposed a convolutional recurrent autoencoder for anomaly detection [11], but this method has been criticized for its relatively low detection accuracy. To enhance accuracy, some researchers introduced Unsupervised Deep Anomaly Detection for Multi-Sensor Time-Series Signals [12], yet this approach relies on a single threshold for anomaly detection, which may prove inadequate for time series with rapid fluctuations.

In conclusion, while existing solutions have advanced IoT multi-sensor time-series anomaly detection and achieved improvements in metrics such as Precision, Recall, and F1 Score, they still struggle to significantly reduce power consumption in sensing terminals while ensuring data security. Additionally, their accuracy can vary depending on the characteristics of different sensors. In contrast, our design achieves near-zero power consumption in sensing terminals while enhancing anomaly detection accuracy by adaptively analyzing multi-sensor time series, generating predictions, and identifying anomalies through thresholding against actual values.

III. One-Dimensional Convolution

Given that the data in this study consists of multi-sensor terminal time series, we first perform local feature extraction using a one-dimensional convolutional layer. This enhances the abstraction of feature representations, allowing for more effective capture and learning of interrelationships among the gates. The refined features are then input into a Gated Recurrent Unit (GRU) layer, which enhances prediction accuracy and anomaly detection robustness. The network architecture is illustrated in Fig. 2.

In order to speed up convergence, the input normalised multi-sensor terminal time series, one sensor time series corresponds to one channel, assuming that the input sequence data is $x = [x_1, x_2, \ldots, x_T]$, where T is the length of the sequence, and each x_t is a vector containing C_{in} channels. Each element y_t of the output sequence $y = [y_1, y_2, ..., y_{T'}]$ of the convolution operation is calculated by the following equation:

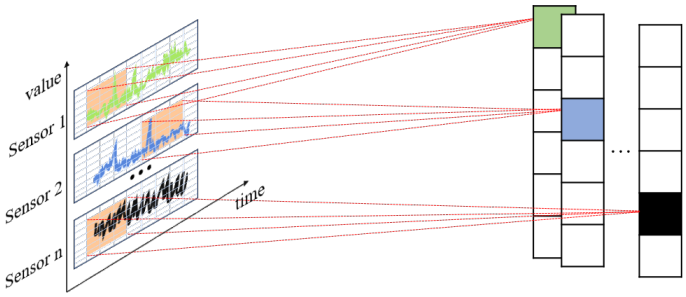

Fig. 2. One-dimensional convolutional network architecture.

$$y_t^k = b^k + \sum_{c=1}^{C_{\text{in}}} \sum_{i=1}^{K} w_{i,c}^k \cdot x_{t+i-1,c} \tag{1}$$

In the Eq.1, y_t^k is the output value of output channel k at time step t, b^k is the bias term of output channel k, $w_{i,c}^k$ is the weight of the convolution kernel denoting the ith weight from the input channel c to the output channel k. The size of the convolution kernel is k. $x_{t+i-1,c}$ is the value of the input sequence x at the time step $t+i-1$ at channel c. T' is the length of the output sequence after the convolution operation, which is calculated by the input length T, convolution kernel size K, step size S, and padding P. The formula is as follows:

$$T' = \frac{T + 2P - K}{S} + 1 \tag{2}$$

By incorporating a one-dimensional convolutional layer, the model's predictive performance and overall effectiveness are enhanced, effectively addressing the challenge of capturing multi-sensor features across multiple time series.

IV. GRU

Capturing long-term dependencies in multi-sensor terminal time series for accurate predictions poses significant challenges, particularly as Recurrent Neural Networks (RNNs) struggle with long-term memory and gradient propagation. While alternatives such as Long Short-Term Memory (LSTM), Convolutional LSTM (ConvLSTM), and Bidirectional LSTM (Bi-LSTM) present potential solutions, they often involve complex architectures, numerous parameters, and extended training times. To overcome these limitations, this paper employs Gated Recurrent Units (GRUs), providing a more efficient approach to modeling long-term dependencies.

The Gated Recurrent Unit (GRU), an evolution of the Long Short-Term Memory Network (LSTM), simplifies the gating mechanism to two gates: the reset gate and the update gate. This design enables the GRU to selectively retain and discard information at various time steps, effectively capturing both long-term and short-term dependencies in sequence data. Furthermore, the GRU extracts pertinent features from the input

multi-sensor terminal time series, which are subsequently fed into an attention mechanism. Compared to traditional LSTM, the GRU's streamlined architecture reduces the number of parameters, enhances training and inference speed, and alleviates the vanishing gradient problem, all while maintaining lower computational complexity. These attributes render the GRU particularly suitable for processing longer sequences. The network structure is illustrated in Fig. 3.

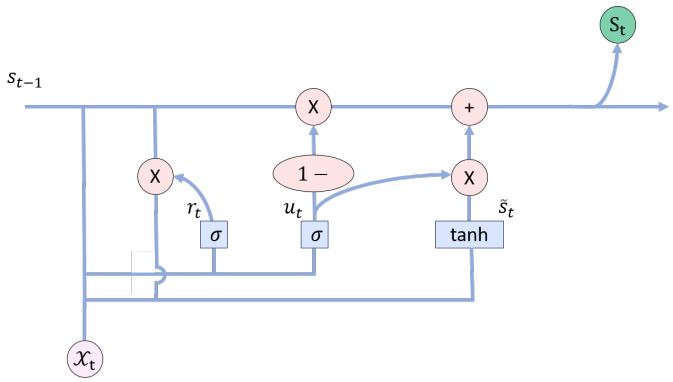

Fig. 3. GRU network structure.

1) Update gate: it controls the extent to which the hidden state (memory) from the previous time step is retained and passed to the current step, enabling it to effectively capture long-term dependencies while adapting to variations in the input data during time series processing. When the update gate value u_t approaches 1, it signifies a preference for retaining the hidden state s_{t-1} from the previous moment. Conversely, when u_t is close to 0, it favors updating the current hidden state with the candidate hidden state. This relationship is governed by the following formula:

$$u_t = \sigma(W_u \cdot [s_{t-1}, x_t]) \tag{3}$$

In the Eq. 3, u_t is the output of the update gate at time step t, is the Sigmoid activation function with output in the range of [0, 1] to control the information flow, W_u is the weight matrix of the update gate, s_{t-1} is the hidden state at the previous moment, and x_t is the input at the current moment.

2) Reset gate: control how much of the previous hidden state should be forgotten or ignored, so that the model can effectively deal with the short-term dependence of information, when r_t is close to 0, the reset gate will be 'closed', i.e., the previous memory is forgotten, and more dependent on the current input x_t; when r_t is close to 1, the gate will be 'open', allowing more information of the previous hidden state s_{t-1} to flow into the current candidate hidden state $\tilde{s}_t$. When r_t approaches 1, the reset gate 'opens', allowing more information from the previous hidden state s_{t-1} to flow into the current candidate hidden state $\tilde{s}_t$, as follows:

$$r_t = \sigma(W_r \cdot [s_{t-1}, x_t]) \tag{4}$$

In the Eq. 4, r_t is the output of the reset gate at time step t, is the Sigmoid activation function, W_r is the weight matrix of the reset gate, s_{t-1} is the hidden state at the previous moment, and x_t is the input at the current moment.

3) Candidate hidden state: generates a possible hidden state for updating the final hidden state at the current moment. This candidate hidden state combines the information of the current input and the hidden state of the previous moment (modulated by the reset gate). When the reset gate r_t is close to 0, the candidate hidden state mainly relies on the current input x_t, i.e., it ignores the hidden state s_{t-1} of the previous moment; when the reset gate r_t is close to 1, the candidate hidden state takes into account the hidden state s_{t-1} of the previous moment, and combines it with the current input x_t in combination with the following formula:

$$\tilde{s}_t = \tanh(W_s \cdot [r_t * s_{t-1}, x_t]) \tag{5}$$

In the Eq. 5, $\tilde{s}_t$ is the candidate hidden state at time step t, W_s is the weight matrix used to compute the candidate hidden state, r_t is the output of the reset gate controlling the effect of the hidden state s_{t-1} at the previous moment in the computation of the candidate hidden state, x_t is the input at the current time step, and tanh is the hyperbolic tangent activation function restricting the output to [-1,1].

4) Hidden state update: The hidden state update is a vital part of the GRU model, integrating the current input with previous hidden states at each time step. This mechanism balances the influence of current data and past states, enabling the model to effectively capture and retain essential information during sequence processing. The formula is as follows:

$$s_t = u_t * \tilde{s}_t + (1 - u_t) * s_{t-1} \tag{6}$$

In the Eq. 6, s_t is the final hidden state of time step t, u_t is the update gate of time step t with a value between [0,1], $\tilde{s}_t$ is the candidate hidden state of time step t, computed from the current input and the hidden state of the previous moment, and s_{t-1} is the hidden state of time step t-1.

We employ the GRU model to capture long-term and short-term dependencies in multi-sensor terminal time series, reducing the number of parameters and enhancing training and inference speed. This makes it well-suited for handling longer sequences and generating highly accurate predictions.

V. Attention Mechanism

We incorporate an attention mechanism following the GRU to selectively emphasize the most relevant information from the GRU's output for the final prediction, thereby enhancing the model's robustness to data variations and noise. The attention mechanism is illustrated in Figure 4.

1) Attention score: the importance score of each time step (or hidden state) relative to all other time steps is calculated to quantify the contribution of each hidden state in generating the final context vector with the following formula:

$$e_t = \mathrm{Va}(\tanh(\mathrm{Wa} \cdot h_t + \mathrm{Ua} \cdot h_{\mathrm{all}})) \tag{7}$$

In the Eq. 7, Va is a linear layer for mapping the output of the hidden state to a scalar fraction e_t, $tanh$ is a nonlinear

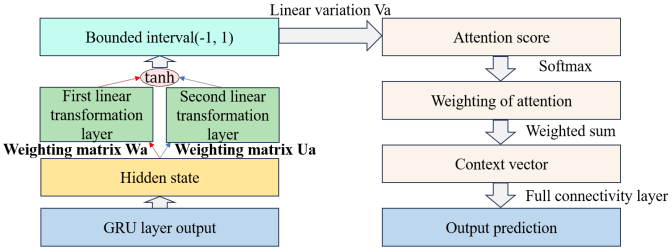

Fig. 4. Attention network structure.

activation function that maps the input to the range (-1, 1), Wa and Ua are linear transformation matrices with learnable weight parameters, h_t is the hidden state at the current time step, h_{all} is the hidden state at all time steps, h_{all} is the hidden state at the current time step and h_{all} is the hidden state at all time steps state.

2) Attention weights: The scores from the previous step are converted into a probability distribution that reflects the relative importance of each time step. Using the SoftMax function, these scores are normalized to positive values summing to 1, facilitating subsequent calculations. The formula is as follows:

$$\alpha_t = \text{softmax}(e_t) \tag{8}$$

3) Context vector: The context vector is formed by weighting and summing the hidden states of all time steps using the attention weights. It summarizes the sequence, emphasizing the most important time steps. The formula is as follows:

$$c = \sum_t \alpha_t \cdot h_t \tag{9}$$

VI. Anomaly Detector

CGAD model, after predicting the multi-sensor time series, this module compares the real sensor time series with the predicted sensor time series to calculate the anomaly scores, and judge the anomalies by sliding window thresholds.

1) Calculation of anomalous scores: the anomalous score S_i between the true value and the pretest is calculated using the Euclidean distance with the following formula:

$$S_i = \sqrt{(y_i - y_i')^2} \tag{10}$$

In the Eq. 10, y_i is the true value and y_i' is the predicted value.

2) Sliding window threshold: Sensor time series often show significant changes only during specific periods. Using an average anomaly score as a threshold can hinder detection. To improve accuracy, we introduce an interactive window threshold, setting distinct thresholds for stable and rapidly changing periods. Let S be a sequence of sensor anomaly scores, W be the window size, N be the length of sequence S, and T be the threshold sequence. For each time step i in the sequence S, the threshold T_i is defined as follows:

$$T_i = \begin{cases} \varphi_1\mu_{[i,i+W]} + \varphi_2\sigma_{[i,i+W]} & \text{if } i < W \\ \varphi_3\mu_{[i-W,i]} + \varphi_4\sigma_{[i-W,i]} & \text{if } i > N - W \\ \varphi_5\mu_{[i-W,i+W]} + \varphi_6\sigma_{[i-W,i+W]} & \text{otherwise} \end{cases} \tag{11}$$

In the Eq. 11, $\mu_{[a,b]}$ denotes the mean of the sequence S from time step a to b,$\sigma_{[a,b]}$ denotes the standard deviation of the sequence S from time step a to b, and φ_{1-6} is the method of coefficients to be determined.

3) Anomaly detection: for each time step i, if the anomaly score S_i is greater than the threshold T_i, the time step is considered to be an anomaly:

$$\text{anomaly}_i = \begin{cases} 1 & \text{if } S_i > T_i \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

With the anomaly detector, we can solve the problem of real-time adaptive multi-sensor time series anomaly detection with unsupervised learning.

VII. Evaluation

In order to evaluate the performance of our design, we implemented the CGAD model and used the KW51 railway bridge and the self-picked temperature and humidity dataset 1 as evaluation examples, the detailed statistics and settings of these two datasets are shown in Table 1.

TABLE I
Dataset Statistics

Statistic	KW51 Railway Bridge	Data Set 1
Time Series Length	12	8
Data Points	54,000	4,700
Training Set	0–34,560	0–3,008
Validation Set	34,561–43,200	3,009–3,760
Test Set	43,201–54,000	3,761–4,700

- **KW51 Railway Bridge.** We selected 54,000 data points from six arch and six bridge acceleration strain sensors over 15 days for our experiments. In this unsupervised learning approach, normal, unlabeled data were augmented with randomly introduced noise and anomalies to evaluate the model's robustness.
- **Data Set 1.** In an industrial environment, we employed four sets of temperature and humidity sensors to continuously collect a total of 4,700 data points regarding equipment conditions for the purpose of this experiment.

To further quantify the predictive effectiveness of each model, we used three statistical metrics to evaluate the performance of the models: the Precision, Recall, and F1 Score.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{13a}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{13b}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13c}$$

TP is the number correctly predicted as abnormal, FP is the number incorrectly predicted as abnormal, and FN is the number incorrectly predicted as normal.

TABLE II
ANOMALY DETECTION RESULTS ON TWO DATASETS

Method	KW51 Railway Bridge			Data Set 1		
	Pre	Rec	F1	Pre	Rec	F1
CNN	0.70	0.78	0.71	0.71	0.97	0.79
LSTM	0.09	0.17	0.12	0.69	0.92	0.74
HA	0.70	0.74	0.70	0.67	0.91	0.75
GAD	0.75	0.74	0.71	**1.00**	0.93	0.96
CAD	0.52	0.62	0.53	0.79	0.99	0.87
CGD	0.20	0.50	0.28	0.93	0.95	0.94
CGAD	**0.76**	**0.98**	**0.83**	0.95	**0.99**	**0.97**

As shown in Table 2, the performance of different multiple time series anomaly detection methods is reported, with the best scores highlighted in bold. We first compare CNN, LSTM, and History Average (HA), noting that while all show good recall on Data Set 1, they underperform relative to our CGAD model, particularly on the KW51 dataset. We also conducted ablation experiments to evaluate each component's contribution: (1) GAD omits the one-dimensional convolution module, (2) CAD excludes the Gated Recurrent Unit module, and (3) CGD removes the Attention Mechanism module.

In conclusion, our proposed CGAD model achieves high recall rates of 98% and 99% for anomaly detection across two datasets, with precision and F1 scores surpassing those of other models. This effectiveness addresses data security concerns in lightweight, low-cost sensing terminals, facilitating the advancement of the Internet of Everything.

VIII. CONCLUSION

In this paper, we propose a new anomaly detection model, CGAD, to address the data security issues of low-cost lightweight sensing terminals. To achieve this, we incorporate an early stopping mechanism to reduce the risk of model overfitting. We also design a dynamic sliding window threshold mechanism that adaptively adjusts the threshold for sensor time series with varying characteristics, thereby enhancing anomaly detection accuracy. Additionally, we develop an anomaly detector that automatically identifies and triggers alarms for anomalous data by calculating the anomaly score between actual and predicted values and integrating it with the sliding window threshold. Through experimental validation on the KW51 railway bridge dataset and a self-collected temperature and humidity dataset, we conduct a comprehensive evaluation. The results indicate that the CGAD model achieves recall rates of 98% and 99% for anomaly detection, with precision and F1 scores surpassing those of other methods. These findings demonstrate the CGAD model's effectiveness in reducing maintenance costs, enhancing data security, and facilitating the Internet of Everything.

ACKNOWLEDGMENT

This work is supported by the Natural Science Starting Project of SWPU (No.2023QHZ002), Sichuan Province science and technology Department key research and development project (2023YFG0129), and the Engineering Research Center for Intelligent Oil Gas Exploration and Development of Sichuan Province.

Supported By Open Fund(PLN202434)of State Key Laboratory of Oil and Gas Reservoir Geology and Exploitation (Southwest Petroleum University).

Supported by the key research and development project of the Sichuan Provincial Science and Technology Program (2023YFG0099).

REFERENCES

[1] T. Kalsoom, N. Ramzan, S. Ahmed, and M. Ur-Rehman, "Advances in sensor technologies in the era of smart factory and industry 4.0," *Sensors*, vol. 20, no. 23, 2020.

[2] C. Zhang, L. Lu, Y. Song, Q. Meng, J. Zhang, X. Shao, G. Zhang, and M. Hou, "Butterfly: w level ulp sensor nodes with high task throughput," *Sensors*, vol. 22, no. 8, 2022.

[3] M. Ibrahim, H. Harb, A. Mansour, A. Nasser, and C. Osswald, "All-in-one: Toward hybrid data collection and energy saving mechanism in sensing-based iot applications," *Peer-to-Peer Networking and Applications*, pp. 1154–1173, 2021.

[4] P. Arpaia, Bonavolontá, F. , and A. Cioffi, "Problems of the advanced encryption standard in protecting internet of things sensor networks." *Measurement*, p. 107853, 2020.

[5] R. Ahmad, R. Wazirali, and T. Abu-Ain, "Machine learning for wireless sensor networks security: An overview of challenges and issues," *Sensors*, p. 4730, 2022.

[6] O. A. Khashan, R. Ahmad, and N. M. Khafajah, "An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks," *Ad Hoc Networks*, vol. 115, p. 102448, 2021.

[7] W. Xue, C. Luo, Y. Shen, R. Rana, G. Lan, S. Jha, A. Seneviratne, and W. Hu, "Towards a compressive-sensing-based lightweight encryption scheme for the internet of things," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 3049–3065, 2020.

[8] A. K. Gautam and R. Kumar, "A comprehensive study on key management, authentication and trust management techniques in wireless sensor networks," *SN Applied Sciences*, vol. 3, no. 1, p. 50, 2021.

[9] M. Ali, L. T. Jung, A.-H. Abdel-Aty, M. Y. Abubakar, M. Elhoseny, and I. Ali, "Semantic-k-nn algorithm: An enhanced version of traditional k-nn algorithm." *Expert Systems with Applications*, p. 113374, 2020.

[10] I. Sarker, A. Colman, J. Han, A. Khan, Y. Abushark, and K. Salah, "Behavdt: A behavioral decision tree learning to build user-centric context-aware predictive model(article)," *Mobile Networks and Applications*, pp. 1151–1161, 2020.

[11] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for iot time series," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 112–122, 2022.

[12] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals," *IEEE Transactions on Knowledge and Data Engineering*, pp. 2118–2132, 2023.