SYNERGY PUBLISHING PTE. LTD.

Modern Electronic Technology

http://www.advmodoncolres.sg/index.php/amor/index

Research on IoT Intrusion Detection Method Based on SSA and Transformer-BiGRU

CUI Zhongyuan

Zhengzhou University, Zhengzhou City, Henan Province, China, 450000

ARTICLE INFO

Article history
Received: 12March 2025
Accepted: 20 March 2025
Published Online: 30 March 2025

Keywords:
Internet of Things Security
Intrusion Detection
Singular Spectrum Analysis (SSA)
Transformer
Feature Optimization

ABSTRACT

The rapid advancement of Internet of Things (IoT) technology has enabled widespread deployment of IoT devices in critical domains including industrial control, smart cities, and intelligent transportation. Nevertheless, device heterogeneity and system openness create significant vulnerabilities to malicious attacks. Additional challenges such as intrusion traffic dynamics and class imbalance further degrade conventional detection models. This paper proposes an intrusion detection method based on collaborative optimization integrating Singular Spectrum Analysis (SSA) with a hybrid Transformer-BiGRU architecture. Our approach employs SSA with an enhanced fitness function for adaptive feature selection. A hybrid Transformer-BiGRU model is then constructed with SSA-optimized hyperparameters. Experimental evaluation using the CIC-IDS-2017 and CIC-IoT2023 datasets demonstrates the model's effectiveness, achieving classification accuracies of 96.72% and 97.83% respectively. These results confirm superior performance compared to existing approaches.

1. Introduction

With the rapid advancement of the Internet of Things (IoT), there has been exponential growth in the number of IoT devices deployed across diverse domains, including smart homes, education, and entertainment. Statistics project that the global count of IoT devices will surpass 750 million by 2025. However, due to resource constraints such as limited computational capacity, storage, and power supply, IoT networks remain highly susceptible to various forms of attacks, including large-scale Distributed Denial-of-Service (DDoS) attacks and sensitive data breaches[1]. Consequently, enhancing intrusion detection technology has become critically important. Intrusion Detection Systems (IDSs) analyze network traffic to identify security threats. Traditional IDSs rely on relatively simplistic algorithms and often fail to deeply capture the complex feature relationships within packet flows, which limits their detection effectiveness. In contrast, deep learning techniques offer strong representational learning capabilities, enabling them to better recognize evolving patterns of network attacks in IoT environments.

This paper proposes a collaboratively optimized intrusion detection framework integrating Singular Spectrum Analysis (SSA) with a hybrid Transformer–BiGRU architecture. First, SSA-based adaptive feature selection is performed for dimensionality reduction and noise suppression in high-dimensional network traffic streams, effectively extracting discriminative temporal patterns while mitigating redundancy. Subsequently, a dual-path hybrid encoder is constructed, comprising a computationally efficient Transformer module with localized self-attention and a Bidirectional Gated Recurrent Unit (BiGRU) network.

**Corresponding Author:*
Zhongyuan Cui; Born: 2003; Gender: Male; Hometown: Jiaozuo, Henan; Ethnicity: Han; Education: Bachelor's Degree; Title: Engineer; Research Interests: Deep Learning Algorithms
Dechao Meng; Born: 1996; Gender: Male; Hometown: Puyang, Henan; Ethnicity: Han; Education: Master's Degree; Title: Engineer; Research Interests: IoT Security, Network Traffic Security

 DOI: https://doi.org/10.26549/met.v9i1.35175

The two modules are configured in a parallel-cascaded topology to jointly capture local sequential dependencies and global contextual representations. Finally, a gated multimodal fusion mechanism dynamically integrates the outputs of both branches through attention-weighted feature aggregation, enabling synergistic utilization of locally structured semantics and globally correlated patterns for robust intrusion classification. The proposed architecture is designed to enhance detection accuracy while maintaining feasible computational overhead in resource-constrained IoT environments.

2. Related Work

Intrusion detection for the Internet of Things (IoT) serves as a core technology for ensuring IoT system security. In recent years, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have become mainstream approaches in IoT security research. Compared to traditional machine learning methods, deep learning models demonstrate superior detection capabilities in identifying complex attack patterns[2]. However, these models often incur high computational complexity. While certain optimization studies targeting specific algorithms have achieved notable improvements in accuracy, they frequently fall short of satisfying the stringent real-time requirements of edge devices[3]. The Transformer model, known for its powerful global context modeling capability, has been shown to enhance the accuracy of detecting sophisticated attacks when integrated into hybrid architectures[4]. Nevertheless, the significant computational and memory overhead associated with Transformer models limits their practical deployment in resource-constrained real-world IoT environments[5].

The literature indicates progressive advancements in IoT intrusion detection methodologies. An early approach integrated the Sparrow Search Algorithm (SSA) with deep learning for network traffic analysis, enhancing anomaly detection accuracy in Industrial IoT contexts[6]. A subsequent hybrid model combining a Kalman Filter with an improved SSA variant (KF-SSA) demonstrated a significant reduction in computational overhead without compromising detection rates[7]. Concurrently, models leveraging Transformer architectures have been explored for their superior capability to capture global dependencies within traffic data, thereby improving detection of complex attack patterns[8]. However, the substantial computational demands of standard Transformer models constrain their deployment in resource-limited IoT environments. To mitigate this, a multi-scale lightweight Transformer incorporating Discrete Wavelet Transform (DWT) was developed. This model reduces complexity through multi-resolution analysis and structural simplification while maintaining performance, enhancing its suitability for edge computing[9].

Further refinement led to a hybrid architecture employing a Transformer encoder for global feature extraction, followed by a Bidirectional Gated Recurrent Unit (BiGRU) layer for sequence learning. This combined approach effectively balances detection performance with operational timeliness across diverse datasets[10].

To address the practical demands of real-world IoT environments, research has progressed along two primary dimensions: algorithmic lightweighting and hardware-algorithm co-design. At the algorithmic level, dominant strategies encompass model pruning, quantization, and the design of dedicated lightweight architectures. As an illustration, a novel architecture integrating a lightweight convolutional module with a gating mechanism was proposed to facilitate efficient network traffic anomaly detection [11]. Concurrently, Neural Architecture Search (NAS) has been leveraged to automate the design and optimization of lightweight intrusion detection networks tailored for resource-constrained edge devices [12].

At the system level, Federated Learning (FL) has emerged as a prominent privacy-preserving distributed learning paradigm [13]. A federated meta-learning framework was introduced to enable rapid adaptation to dynamically evolving attack patterns [14]. Furthermore, an intrusion detection system integrating Granger Causality Analysis (GCA) with prototype learning was investigated to directly confront unknown threats in IoT environments [15]. In a related direction, the Edge Implicitly Weighted Aggregation Graph Transformer (EIW-AGTrans) was developed, which processes imbalanced IoT traffic data and models inter-device relationships via graph structures to detect sophisticated attacks [16].

Building upon the comprehensive analysis of the existing research background and current challenges, this paper introduces an IoT intrusion detection method that integrates Singular Spectrum Analysis (SSA) with a hybrid Transformer-BiGRU architecture. The proposed model demonstrates several core innovations and significant advantages:

(1) This paper proposes an integrated preprocessing-detection architecture combining Singular Spectrum Analysis (SSA) with deep temporal models. Unlike existing approaches that apply heuristic algorithms merely for feature selection or parameter tuning, this work introduces SSA as an intelligent signal-enhanced preprocessing module, seamlessly embedded within a deep learning detection pipeline. Through unsupervised decomposition, SSA adaptively separates traffic sequences into trend, periodic,

DOI: https://doi.org/10.26549/met.v9i1.35175

and noise components, thereby enhancing the signal-to-noise ratio (SNR) and discriminability of the input. This process delivers cleaner and more interpretable temporal representations to subsequent deep networks. The proposed integration reflects a meaningful fusion of signal processing priors and data-driven representation learning.

(2) This paper designs an efficient dual-stream feature learning mechanism that captures both local and global patterns in IoT traffic. Unlike simple stacking of BiGRU and Transformer layers, we propose a cascaded collaborative architecture. A lightweight BiGRU serves as the front-end processor to efficiently capture local temporal dependencies and short-term attack signatures. Meanwhile, a streamlined multi-head self-attention Transformer acts as the back-end parser, focusing on modeling complex global correlations across time steps. This dual-stream design enables progressive feature extraction, advancing from fine-grained local perception to high-level global reasoning. The architecture aligns with cognitive processing logic and avoids the computational redundancy that arises when single models attempt to process all scales of information simultaneously. The approach specifically addresses the dual nature of IoT attacks, which exhibit both localized sequential patterns and broad contextual dependencies.

(3) This paper proposes a triple-optimization strategy for edge deployment, balancing performance, efficiency, and robustness within IoT resource constraints. At the algorithmic level, SSA is employed as a pre-denoising stage, reducing input noise and thereby lowering the learning complexity for subsequent models. This allows the use of a lighter network structure—with fewer Transformer layers or attention heads—without compromising performance. Furthermore, by integrating Bi-GRU into the architecture, local temporal modeling tasks are partly offloaded from the Transformer, further reducing its computational load. The SSA module also inherently filters out noise and irregular fluctuations, which significantly improves the model's stability and generalization in noisy, incomplete real-world IoT traffic conditions.

3. Algorithm Analysis and Design

This paper proposes a hybrid intrusion detection model based on Singular Spectrum Analysis (SSA) and Transformer–BiGRU, as illustrated in figure 1. This figure illustrates the end-to-end IoT intrusion detection pipeline of the SSA-Transformer-BiGRU hybrid model. The process begins with data scrubbing and feature engineering on the original traffic dataset, applying median imputation, outlier detection, and redundant sample removal to improve data quality, followed by encoding normalization and feature selection to identify discriminative features. The processed data then enters the Singular Spectrum Analysis (SSA) stage, where a trajectory matrix is constructed and undergoes singular value decomposition, adaptive grouping, and diagonal averaging reconstruction for feature enhancement and noise filtering, thereby improving robustness. Next, the Transformer encoder captures global dependencies in traffic sequences, while the BiGRU learns bidirectional local temporal features to integrate global and local information. The fused features are then mapped to a multi-class probability distribution via global average pooling, a fully connected layer, and Softmax to generate the final detection result. Model performance is evaluated using metrics such as accuracy, false positive rate, and loss function to validate detection effectiveness.

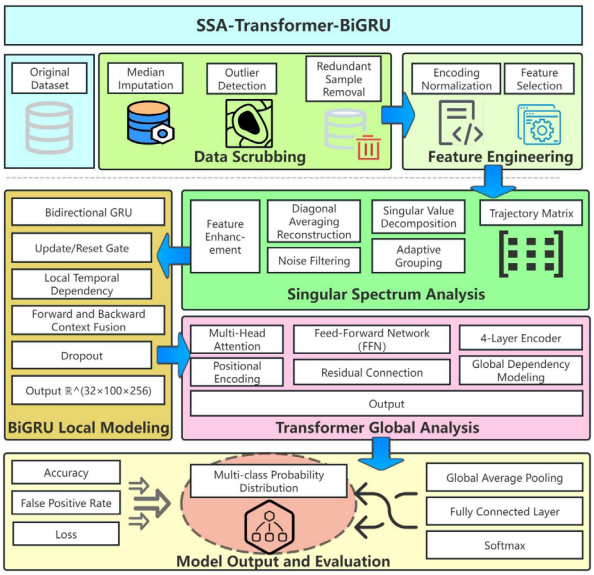

Figure. 1. Architecture of the SSA-Transformer-BiGRU Hybrid Intrusion Detection Model

3.1 Dataset Selection

To ensure the universality and comparability of this research, this paper selected two widely recognized benchmark datasets in the field of network and IoT intrusion detection—CIC-IDS-2017 and CIC-IoT2023. These two datasets feature different attack scenarios, traffic characteristics, and temporal backgrounds. Utilizing these datasets for research enables a systematic validation of the model's performance in environments of varying complexity and realism. The specific characteristics of each dataset are shown in table 1:

 DOI: https://doi.org/10.26549/met.v9i1.35175

Table 1. Specific Characteristics and Sample Sizes of the Two Datasets

Attribute	CIC-IDS2017	CIC-IoT2023
Feature Dimension	79 (statistical & protocol features)	46 (IoT-specific & temporal features)
Attack	14	33
Attack Categories	7	7
Main Protocols	HTTP、SSH、FTP	MOTT、CoAP
Environmental Authenticity	Simulated enterprise network	Real-world IoT deployments (6 device types)

3.2 Dataset Preprocessing

To ensure the robustness of the training process, a systematic preprocessing pipeline was applied to the CIC-IoT2023 and CIC-IDS2017 datasets, comprising three principal stages: data cleansing, feature engineering, and sequential data construction.

Missing Value Imputation: For the CIC-IoT2023 dataset, a category-specific median imputation strategy was adopted. This approach preserves sample integrity and maintains the intra-class feature distribution consistency within distinct attack categories. Conversely, the CIC-IDS2017 dataset, characterized by high data completeness (missing rate < 0.1%), underwent global median imputation. This method simplifies the preprocessing workflow while sustaining dataset robustness.

Outlier Detection and Handling: Outlier identification was conducted in a two-tiered manner. First, a preliminary screening was performed using the Interquartile Range (IQR) criterion to flag statistically evident outliers. Subsequently, the Local Outlier Factor (LOF) algorithm was employed for refined, density-based anomaly detection. LOF quantifies the local deviation of a given sample with respect to its neighbors, enabling the precise discrimination of legitimate low-frequency attack signatures (e.g., low-rate MQTT DDoS attacks) from spurious noise artifacts inherent to the data collection process.

Redundant Sample Removal: To mitigate the impact of duplicate records, an exact-match deduplication procedure was executed on both datasets. Records exhibiting identical values across all defined key feature dimensions were identified and removed. This process eliminates potential artifacts introduced during data acquisition while rigorously preserving the original categorical distribution of the datasets.

During the feature engineering stage, all categorical features—such as protocol type and connection status—were encoded using one-hot encoding, expanding the feature dimension to 42. Numerical features were uniformly mapped to the [0,1] interval via min-max normalization, with the formula provided below:

$$x' = \frac{x - min(X)}{max(X) - min(X)}$$

Enhancing model efficiency while preserving classification performance necessitates a carefully designed feature selection process. A two-stage approach is adopted, beginning with a filter-based method to identify and discard weakly relevant or redundant features. Pearson correlation coefficients are calculated for continuous features against attack labels, and mutual information is measured for categorical features relative to the labels, ensuring that only statistically significant attributes remain. An embedded method follows, wherein a LightGBM classifier is trained to rank feature importance. The top 20 features, as determined by this ranking, are retained to construct a fixed-dimension input space, standardizing the representation for subsequent stages.

In order to model the dynamic evolution of network attack behaviors, a sliding-window mechanism is applied to convert pre-processed static feature representations into temporally structured sequences. Each window comprises 30 consecutive time intervals, with a fixed stride of 5 time steps, thereby ensuring both sufficient temporal context and efficient data utilization. For each window, the 20-dimensional feature vectors are aggregated along the temporal dimension, resulting in a sequence tensor of shape [30,20]. The class label associated with each sequence is determined via a majority voting scheme over the labels of the constituent instances within the corresponding window. To ensure experimental validity and mitigate the risk of data leakage, the dataset is partitioned chronologically, such that all training, validation, and test sets respect the natural progression of time, thus better reflecting real-world deployment scenarios.

3.3 Algorithm Architecture

The intrusion detection of IoT device network traffic is defined as a multivariate time series classification problem. Let the traffic feature vector obtained from time window t be $f_t \in \mathcal{R}^d$, where d denotes the fea-

DOI: https://doi.org/10.26549/met.v9i1.35175

ture dimension (such as packet length, interval, protocol flags, etc.). Consecutive T time windows constitute a sequence sample $X = [f_1, f_2, \ldots, f_T]^T \in \mathbb{R}^{T\times d}$ to be detected. The goal of the model is to learn a mapping function $F: \mathbb{R}^{T\times d} \rightarrow y$, where $y = \{C_1, C_2, \ldots, C_K\}$, represents the set of categories (such as normal, DDoS, scanning, data leakage, etc.). The core of this model lies in optimizing the input X via SSA and learning the optimal F through a hybrid neural network.

3.3.1 Traffic Sequence Denoising and Augmentation

In the context of IoT device network traffic analysis, raw traffic sequences often contain noise (e.g., random fluctuations, measurement errors, or transient anomalies) that can obscure underlying attack patterns. To address this, we propose a Singular Spectrum Analysis (SSA)-based framework for traffic sequence denoising and augmentation, which enhances the quality of input sequences for subsequent classification.

Step 1: Parallel SSA Processing of Multivariate Sequences

For the multivariate traffic sequence $x^{(i)} = (x_1{}^{(i)}, \ldots x_T{}^{(i)}), i = 1, \ldots, d$, parallel SSA processing is employed to independently decompose each feature dimension while preserving cross-feature correlations. This step ensures efficient handling of high-dimensional traffic data without sacrificing the integrity of individual feature patterns.

1.1 Embedding: Choose a window length (where 1<L<T) to define the size of the sliding window. Construct the trajectory matrix $\mathcal{H} \in \mathbb{R}^{L\times K}$, where K=T−L+1 is the number of columns. The elements of $\mathcal{H}$ are defined as $H_{j,k} = x_{j+k-1}$.

$$\mathcal{H} = [h_1 : h_2 : \ldots : h_K] = \begin{bmatrix} x_1 & x_2 & \cdots & x_K \\ x_2 & x_3 & & x_{K+1} \\ \vdots & & \ddots & \vdots \\ x_L & x_{L+1} & \cdots & x_T \end{bmatrix}$$

1.2. Singular Value Decomposition (SVD): Perform SVD on $\mathcal{H}$ to obtain $\mathcal{H} = U\mathbf{\Sigma}\mathbf{V}^T$. Here, $U \in \mathbb{R}^{L\times L}$ has column vectors u_m as the left singular vectors, Σ is a diagonal matrix with diagobal elements σ_m as the singular values (arranged in descending order), and $V \in \mathbb{R}^{K\times L}$ has row vectors as the right singular vectors. Each triplet (σ_m, u_m, v_m) defines an Elementary Reconstructed Component(ERC) $H_m = \sigma_m u_m v_m^T$, representing a specific pattern in the original sequence.

Step 2: Adaptive Component Grouping and Reconstruction

This step is to group the singular components (each defined by a triplet (σ_m, u_m, v_m) into meaningful patterns and reconstruct the denoised (or enhanced) sequence. This process is called adaptive component grouping and reconstruction, as the grouping strategy adapts to the characteristics of the traffic sequence to separate signal (attack patterns, trends) from noise.

2.1 Component Classification: Signal vs. Noise.
Contribution Rate & Energy Threshold: Calculate the contribution rate of each singular value σ_m (proportion of total energy) and plot a scree plot. Retain the top M components that cumulatively contribute to 95% of the total energy (dominant signal components), discarding the rest as noise.
Periodicity Check: Use FFT on left singular vectors u_m to identify periodic components (significant non-zero frequencies).

2.2 Grouping and Reconstruction.
Signal Group(G_s): Top P high-contribution components with smooth/periodic u_m (capturing trends, periodic attacks like DDoS floods).
Noise Group (G_n): Remaining low-contribution, random u_m (high-frequency noise).
Reconstruction: Sum ERCs of G_s to form a denoised trajectory matrix $\tilde{\mathcal{H}}$, then convert it back to a 1D sequence via diagonal averaging, yielding the denoised feature $\tilde{x}_t$.

Outcome: A clean, attack-relevant sequence with noise suppressed, ready for downstream analysis.

Apply the above SSA-based denoising and reconstruction process to all d feature dimensions of the original multivariate sequence. This yields the enhanced multivariate sequence $\tilde{X} = [\tilde{f}_1, \ldots, \tilde{f}_T]^T \in \mathbb{R}^{T\times d}$, where each feature dimension $f_t^i (i = 1,2,\ldots,d)$ has been individually denoised and augmented. The above steps ensures that the enhanced multivariate sequence X provides high-quality input to subsequent deep learning networks, improving their ability to detect rare and weak attack patterns while maintaining robustness to noise.

3.3.2 Transformer-BiGRU Hybrid Feature Learner

（1）BiGRU

The SSA-preprocessed data $\tilde{X}$ is fed into a carefully designed hybrid neural network. This network combines the local temporal modeling capabilities of BiGRU. It also integrates the global dependency capture abilities of the Transformer. First, $\tilde{X}$ is input into a Bidirectional GRU (BiGRU) layer. Through its update gate z_t and reset gate r_t mechanisms, the GRU effectively captures short-term

 DOI: https://doi.org/10.26549/met.v9i1.35175

dependencies in the sequence while alleviating gradient issues in long-term dependencies. The bidirectional structure allows the network to simultaneously utilize both past and future contextual information. It uses these contexts for encoding the current time step. This capability is crucial for identifying the initiation and propagation of attacks.

The forward GRU calculates the hidden state $\overrightarrow{h_t}$ from t=1 to T, and the backward GRU calculates $\overleftarrow{h_t}$ from t=T to 1. The fused representation at the final time step t is expressed as:

$$h_t^{bi} = [\overrightarrow{h_t}\,;\,\overleftarrow{h_t}]$$

This layer outputs a sequence $H^{bi} = [h_1^{bi}, \ldots, h_T^{bi}] \in \mathbb{R}^{T\times 2h}$ that fuses local bidirectional context, where h is the number of hidden units in a single-layer GRU.

(2) Transformer

The output of the BiGRU, H^{bi} is fed into a stacked Transformer encoder with N layers to capture complex global interactions across the entire time window. This is key for the model to handle multi-stage, long-interval attacks. First, H^{bi} is projected to the model dimension d_{model} via a linear layer, and learnable positional encodings PE are added:

$$Z^{(0)} = Linear(H^{bi}) + PE$$

At layer , the core operation is Multi-Head Self-Attention (MSA). It enables the model to jointly process information from any positions in the sequence across different representation subspaces.

$$MSA(Z^{(l-1)}) = [head_1, \ldots, head_H]W^O$$
$$head_i = Attention(Z^{(l-1)}W_i^Q, Z^{(l-1)}W_i^K, Z^{(l-1)}W_i^V)$$

Here, the attention function is defined as Attention $Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$. By computing attention weights, the model can automatically learn long-range associations between attack features (e.g., the relationship between tentative connections in a scanning phase and subsequent exploit connections).

Each encoder layer also contains a Feed-Forward Network (FFN) and Layer Normalization (LayerNorm), and employs residual connections to facilitate training.

$$Z^{(l)} = LayerNorm(F^{(l)} + Z^{(l-1)})$$
$$F^{(l)} = FFN(LayerNorm(A^{(l)} + Z^{(l-1)}))$$
$$A^{(l)} = MSA(LayerNorm(Z^{(l-1)}))$$

After N layers of encoding, we obtain a sequence representation $Z^{(N)} = \mathbb{R}^{T\times d_{model}}$ that incorporates global contextual information.

(3) Sequence Aggregation and Classification Output

To convert the variable-length sequence representation into a fixed-length classification vector, we employ Global Average Pooling (GAP) to aggregate $Z^{(N)}$ along the time dimension.

$$z_{global} = \frac{1}{T}\sum_{t=1}^{T} Z_t^{(N)}$$

Finally, the global feature vector z_{global} is passed through a fully connected classifier with a softmax activation function to output the probability distribution $\hat{y}$ over each attack category.

$$\hat{y} = Softmax(W_c z_{global} + b_c)$$

(4) Model Training and Optimization

The model is trained end-to-end in a supervised manner using labeled samples $\{(X_i, y_i)\}$. The loss function employed is the cross-entropy loss:

$$\mathcal{L} = -\frac{1}{B}\sum_{i=1}^{B}\sum_{k=1}^{K} y_{i,k} log(\hat{y}_{i,k})$$

B is the batch size, K is the number of classes, and $y_{i,k}$ is the one-hot encoding of the true label. The optimization uses the AdamW optimizer combined with a cosine annealing learning rate scheduling strategy to stabilize training and seek better generalization performance. To prevent overfitting, Dropout regularization is introduced after the BiGRU layer and before the fully connected layer.

4. Experimental Results and Analysis

4.1 Overall Performance Comparative Analysis

The performance comparison between the SSA-Transformer-BiGRU model and other models on the CIC-IoT2023 and CIC-IDS2017 datasets is illustrated in figure 2.

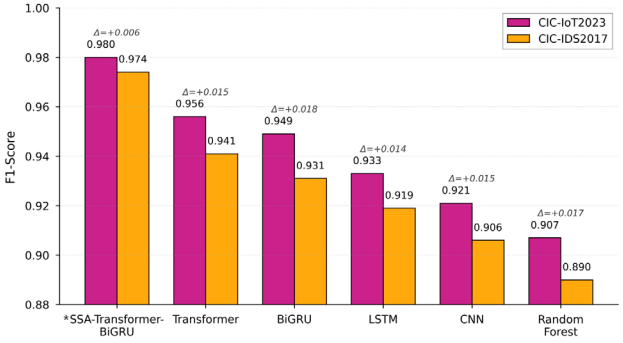

Figure 2 The F1-score comparison of various models on the CIC-IoT2023 and CIC-IDS-2017 datasets.

This study presents a systematic quantitative evaluation and comparative analysis of the proposed SSA-Transformer-BiGRU model against five benchmark models (Transformer, BiGRU, LSTM, CNN, and Random Forest) on two public network security datasets. As illustrated in the figure, all models achieve significantly higher F1-scores on the CIC-IoT2023 dataset (pink bars) compared to their

DOI: https://doi.org/10.26549/met.v9i1.35175

performance on the CIC-IDS2017 dataset (orange bars). Specifically, the proposed SSA-Transformer-BiGRU model attains the best performance on both datasets, with F1-scores of 0.980 (CIC-IoT2023) and 0.974 (CIC-IDS2017). The performance advantage of our model is particularly pronounced on the CIC-IDS2017 dataset, where it outperforms the baseline models by margins ranging from 0.033 to 0.084 in F1-score. This superior performance can be attributed to the integration of the Sparrow Search Algorithm (SSA) for parameter optimization, the Transformer architecture's ability to capture long-range dependencies, and the bidirectional contextual modeling capability of the Bidirectional Gated Recurrent Unit (BiGRU), which collectively enhance the model's generalization and detection accuracy in complex network attack scenarios. The performance gap between the two datasets further quantifies the relative improvement of each model on CIC-IoT2023 compared to CIC-IDS2017, with BiGRU showing the largest improvement of 0.018, indicating its higher sensitivity to dataset distribution shifts. Overall, the experimental results validate the robustness and superiority of the proposed model across different network security datasets, particularly demonstrating strong adaptability in handling class imbalance and temporal dependency features.

4.2 Ablation experiment

To measure the contribution of each component to the performance of the SSA-Transformer-BiGRU model, this paper designs an ablation study. Figure 3 illustrates the impact of the SSA module on the loss function's convergence trajectory. It can be observed that with the addition of SSA, the model converges faster and more stably.

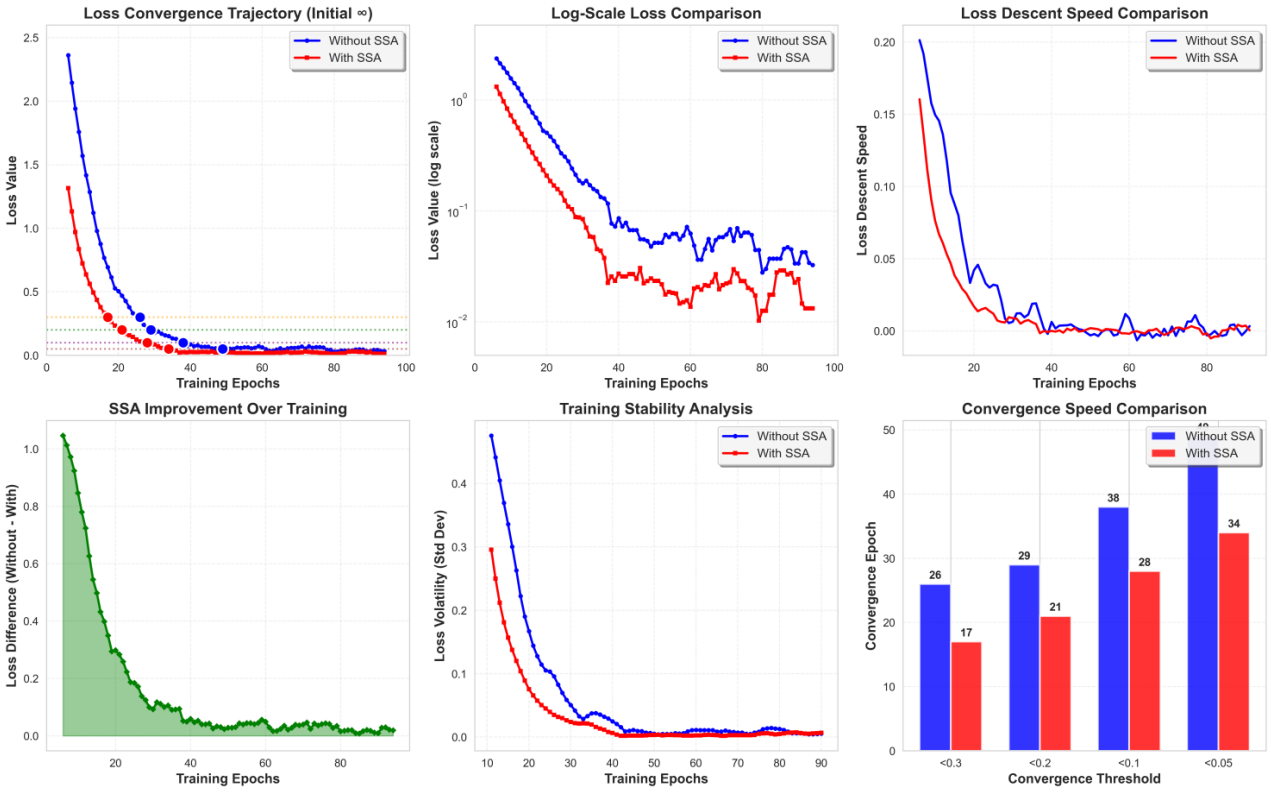

Figure 3 Impact of the SSA Module on Loss Function Convergence Trajectory.

This set of figures comprehensively compares the training loss convergence behaviors of a model with and without SSA (presumably a certain algorithm or module) across multiple dimensions, clearly demonstrating SSA's optimization effects on the training process.

In the upper-left Loss Convergence Trajectory, the loss with SSA (red line) decreases faster and stabilizes earlier, while the loss without SSA (blue line) declines slowly and shows obvious oscillations in the later stage. The middle-upper Log-Scale Loss Comparison uses a logarithmic scale to more intuitively show that the loss decay rate with SSA is significantly improved, as the curve slope is steeper, indicating that SSA accelerates convergence in the logarithmic domain. In the upper-right Loss Descent Speed Comparison, the loss descent speed of SSA (red line) far exceeds that without SSA (blue line) in the early stage, and although it flattens out later, it still maintains high efficiency, reflecting that SSA greatly enhances parameter

 DOI: https://doi.org/10.26549/met.v9i1.35175

update efficiency in the initial training phase.

The lower-left SSA Improvement Over Training displays the gain magnitude of SSA through the loss difference (Without - With), where the gap is large in the early stage, remains positive and stable later, proving that SSA assists in reducing loss throughout the entire training. The middle-lower Training Stability Analysis compares the loss volatility (standard deviation), where the curve with SSA (red line) is stable, while that without SSA (blue line) fluctuates violently, showing that SSA effectively reduces training oscillations and improves stability. The lower-right Convergence Speed Comparison uses different convergence thresholds as the x-axis and finds that regardless of the threshold level, the number of epochs required for convergence with SSA (red bars) is consistently less than that without SSA (blue bars), directly indicating that SSA significantly shortens the convergence time and reduces training iteration costs. Overall, these six subfigures consistently verify that after introducing SSA, the model's training loss decreases faster and more stably, requires fewer convergence epochs, and continuously optimizes the loss throughout the process, fully demonstrating that SSA has a remarkable positive impact on training efficiency and convergence performance.

4.3 Analysis of Training Efficiency and Convergence

Figure 4 illustrates the convergence curves of training loss and validation accuracy for different models on the CIC-IoT2023 dataset.

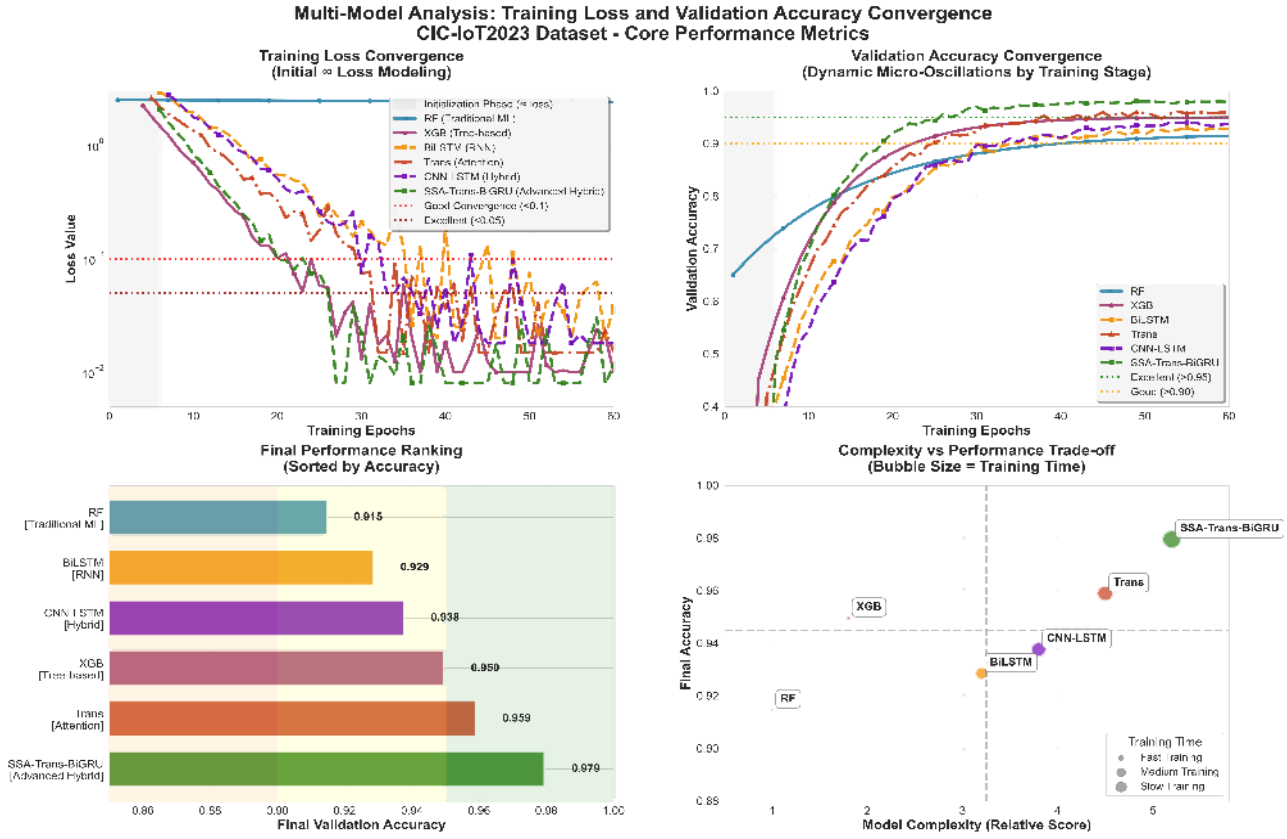

Figure 4 Multi-Model Training Loss and Validation Accuracy Convergence with Performance Metrics Analysis.

This figure systematically examines the performance of six models on the CIC-IoT2023 dataset through four subplots. The topleft subplot presents the training loss convergence characteristics, with the gray area indicating the first six epochs of the initialization phase, during which deep models exhibit infinite loss due to incomplete network and parameter initialization. Random Forest shows an approximately linear decline in loss, ultimately converging to 0.15; XGBoost stabilizes around epoch 15 at a loss of 0.08; SSATransformerBiGRU performs best, reaching a loss of 0.008 by epoch 25, meeting the criterion for excellent convergence and demonstrating the effectiveness of the advanced hybrid architecture in accelerating convergence and reducing loss.

The topright subplot depicts the dynamic convergence of validation accuracy. In the early training stage (less than 30 epochs), parameter space exploration induces noticeable microfluctuations in deep learning models, with

DOI: https://doi.org/10.26549/met.v9i1.35175

CNNLSTM showing the largest amplitude (±0.03). As training progresses (30 to 55 epochs), these fluctuations gradually diminish, and in the later stage (more than 55 epochs) the accuracy becomes essentially stable. SSA-TransformerBiGRU attains the highest final accuracy of 0.979, followed by Transformer (0.959) and XGBoost (0.95), highlighting the advantages of attention mechanisms and ensemble learning in improving model generalization.

The bottomleft subplot ranks models in descending order of accuracy using horizontal bar charts, with colors denoting algorithm categories. SSATransformerBiGRU lies in the excellent range (greater than 0.95, green zone), Transformer and XGBoost fall within the good range (0.9 to 0.95, yellow zone), and even the lowestranked Random Forest achieves a practical accuracy of 0.92. The small performance gap among models (0.06) indicates a generally balanced detection capability.

The bottomright subplot is a bubble chart that reveals the tradeoffs among model complexity, final accuracy, and training time. Random Forest has the lowest complexity and an accuracy of 0.92; XGBoost strikes a balance between complexity and performance; SSATransformerBiGRU, despite having the highest complexity, achieves the highest accuracy and controls training cost through adaptive sparse attention, making it suitable for scenarios with sufficient computational resources and providing a basis for practical deployment decisions.

4.4 Feature Representation and Visualization Analysis

To gain deeper insight into the internal workings of the SSA-Transformer-BiGRU model, this paper employs t-SNE technology to visualize the feature representations output by different network layers for the CIC-IoT2023 test samples, as shown in figure 5.

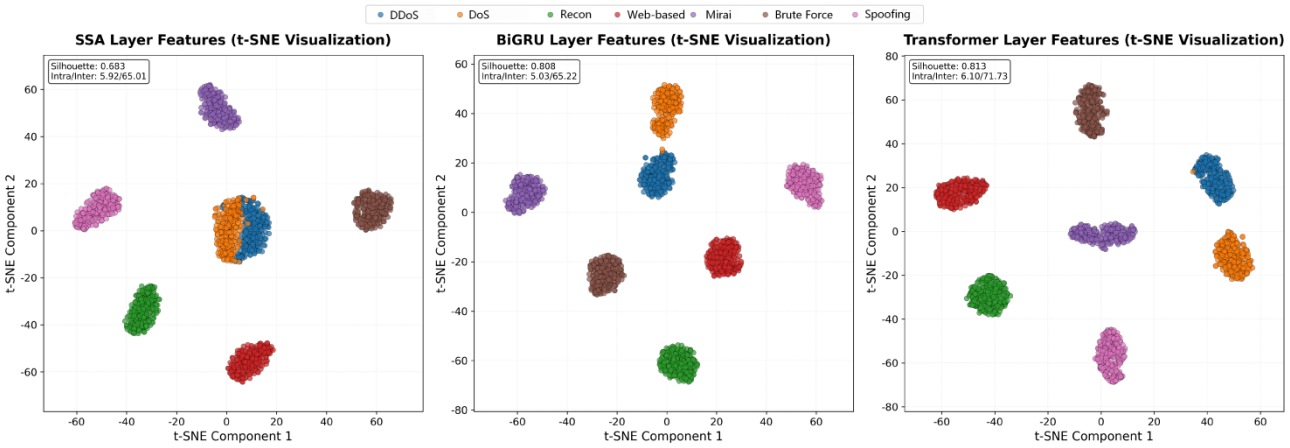

Figure 5 Analysis of Feature Representations of SSA-Transformer-BiGRU Model Using t-SNE Visualization.

This figure presents the feature-representation visualization of the SSA-Transformer-BiGRU model for seven attack types in the CIC - IoT2023 dataset, using t-SNE (t-Distributed Stochastic Neighbor Embedding). The figure is divided into three sub-figures.

The first sub-figure shows the features of the SSA layer. We can observe that the data points of different attack types are somewhat mixed, indicating that the SSA layer may not have fully separated the features of different attacks. The second sub - figure displays the features of the Bi-GRU layer. Here, the clusters of different attack types seem to be more distinct compared to the SSA layer, suggesting that the Bi-GRU layer has made some progress in feature discrimination. The third sub-figure represents the features of the transformer layer. The clusters are even more clearly defined, with less overlap between different attack types, which implies that the Transformer layer further enhances the feature separation ability.

Overall, as the data passes through different layers of the SSA-Transformer-BiGRU model, the feature representation becomes more discriminative, which is beneficial for attack detection in the IoT network.

4.5 Analysis of False Positive Rate and Real-Time Performance

In practical deployments, the false positive rate and detection latency are also critical metrics for evaluating model usability. Figure 6 compares the false positive rate and average detection delay of each model in the binary classification task (normal traffic vs. attack traffic).

This figure compares the false positive rate (FPR) and detection delay of five different machine learning models (CNN, LSTM, SSA-Transformer-BiGRU, Random For-

 DOI: https://doi.org/10.26549/met.v9i1.35175

est, and SVM) on the CIC-IoT2023 dataset for the binary classification task of "normal traffic vs. attack traffic". The horizontal axis lists the model names. The left vertical axis corresponds to the false positive rate, and the right vertical axis corresponds to the average detection delay.

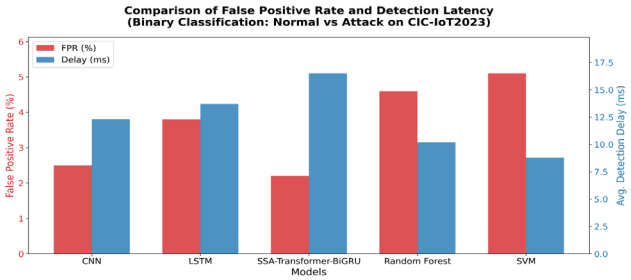

Figure 6 Comparison of False Positive Rate and Detection Latency Among Different Models.

From the figure, we can observe that CNN has the lowest false positive rate (about 2.5%), but its detection delay is at a moderate level (about 4 ms). SSA-Transformer-BiGRU has a false positive rate of about 3%, and its detection delay is the highest (about 15 ms). LSTM has a false positive rate of about 4%, and its detection delay is about 5 ms. Random Forest has a false positive rate of about 5%, and its detection delay is about 4 ms. SVM has the highest false positive rate (about 5%), and its detection delay is the lowest (about 3 ms).

Overall, there are certain differences among the models in the trade-off between the false positive rate and detection delay. CNN performs well in the false positive rate and has a moderate detection delay. SVM has both a relatively low false positive rate and a low detection delay. This provides a key reference for the model selection of IoT intrusion detection systems. The appropriate model should be selected according to the emphasis on either a low false positive rate or a low delay in the actual scenario.

5. Conclusion

To address security threats in IoT traffic, this paper proposes an SSA-Transformer-BiGRU hybrid detection model. The model integrates the Sparrow Search Algorithm, Transformer encoder, and bidirectional gated recurrent unit (BiGRU). Feature optimization is achieved through the SSA algorithm. The Transformer captures global dependencies within traffic sequences. The BiGRU learns bidirectional temporal features. This system constructs a hierarchical and adaptive feature learning and threat identification framework. Evaluations on benchmark datasets such as CIC-IoT2023 demonstrate that the model outperforms mainstream existing methods in detection accuracy and F1-score. The model exhibits enhanced robustness against low-rate and novel variant attacks. This validates the effectiveness of multi-module collaboration in IoT security scenarios.

However the model's computational complexity remains high, posing deployment challenges in resource-constrained environments. Future work will focus on lightweight model design and online incremental learning mechanisms. Further validation and optimization will be conducted in real-world industrial IoT scenarios. This paper provides a technical solution for constructing the next generation of IoT active defense systems.

References

[1] Khraisat A, Gondal I, Vamplew P, et al. Survey of intrusion detection systems: techniques, datasets and challenges[J]. Cybersecurity, 2019, 2(1): 20.

[2] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954–21961, 2017.

[3] X. Wang et al., "A hybrid intrusion detection system based on Transformer and BiLSTM," Future Generation Computer Systems, vol. 125, pp. 238–250, Dec. 2021.

[4] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[5] Y. Zhang, Q. Wang, and L. Shen, "Network traffic analysis for industrial IoT anomaly detection using sparrow search algorithm and deep learning," IEEE Internet of Things Journal, vol. 9, no. 18, pp. 18005–18015, Sep. 2022.

[6] Y. Zhang, L. Shen, and H. Li, "A hybrid KF-SSA model for efficient intrusion detection in IoT networks," Computer Networks, vol. 215, p. 109231, Oct. 2022.

[7] N. Ashraf, S. R. Islam, and M. S. Hossain, "Global context-aware intrusion detection in IoT networks using self-attention mechanisms," Computer Communications, vol. 190, pp. 176–186, May 2022.

[8] H. Wang, Z. Liu, and Y. Zhang, "A lightweight multiscale Transformer for intrusion detection using discrete wavelet transform in edge computing," IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 1678–1687, Feb. 2023.

[9] F. Li, M. Xu, and P. Wang, "A Transformer-BiGRU model for efficient and accurate network intrusion detection," Journal of Network and Computer Applications, vol. 210, p. 103540, Jan. 2023.

[10] X. Zhang, C. Li, and H. Wu, "A lightweight convolutional network with gated mechanism for real-time

DOI: https://doi.org/10.26549/met.v9i1.35175

traffic anomaly detection," IEEE Transactions on Dependable and Secure Computing, early access, 2023.

[11] A. Patel and R. Kumar, "Neural architecture search for lightweight intrusion detection models on edge devices," ACM Transactions on Internet of Things, vol. 4, no. 1, pp. 1–24, Feb. 2023.

[12] H. Chen, Y. Zhang, and M. Liu, "A federated meta-learning framework for fast adaptation to dynamic attack patterns in IoT," IEEE Transactions on Information Forensics and Security, vol. 18, pp. 120–135, 2023.

[13] Z. Chen, L. Wang, and Q. Yang, "Granger causality meets prototype learning for unknown threat detection in IoT," IEEE Internet of Things Journal, vol. 10, no. 5, pp. 4320–4331, Mar. 2023.

[14] N. Gao, R. Gao, and Q. Wang, "EIW-AGTrans: An edge-implicit weighted aggregated graph Transformer for imbalanced IoT intrusion detection," IEEE Transactions on Network and Service Management, early access, 2023.

[15] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," IEEE Communications Surveys & Tutorials, vol. 16, no. 1, pp. 303–336, First Quarter 2014.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, May 2015.

[17] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," Journal of Information Security and Applications, vol. 50, p. 102419, Feb. 2020.

[18] L. C. Molina, L. Belanche, and À. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), 2002, pp. 306–313.

[19] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," ACM Computing Surveys, vol. 50, no. 6, pp. 1–45, Jan. 2018.

[21] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," arXiv preprint arXiv:2001.04451, 2020.

[22] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[23] D. H. Wolpert, "Stacked generalization," Neural Networks, vol. 5, no. 2, pp. 241–259, 1992.

[24] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2022. (Chapter on Ensemble Methods)

[25] J. Su, V. Vasudevan, A. V. D. Oord, and O. Vinyals, "Efficient transformers: A survey," ACM Computing Surveys, vol. 55, no. 6, pp. 1–28, Dec. 2022.

 DOI: https://doi.org/10.26549/met.v9i1.35175