OPTIMIZATIONS IN APPLIED MACHINE LEARNING
Research Article* | *Volume 4* | *Issue 4* | *Dec 2024
Received: 14 Nov 2024 | *Revised: 24 Nov 2024*
Accepted: 18 Dec 2024 | *Published Online: 20 December 2024*

A Study on the Mamba-ECANet Model for Intrusion Detection in Data Security Using End-to-End Learning

Meng Wang, Newmark Group, Email: wang070210@gmail.com;

Huitao Zhang*, Northen Arizona University, Email: hz345@nau.edu;

Ning Zhou, Zhejiang Future Technology LLC, Email: zhouning723@gmail.com

Abstract: With the rapid advancement of information technology, network security has become an increasingly critical concern. In particular, data security intrusions pose significant risks to the privacy of data and the security of both enterprise and personal systems. Traditional intrusion detection systems often struggle with low detection accuracy and high false alarm rates, especially in complex and dynamic network environments with diverse attack techniques. To address these challenges, this paper proposes a deep learning-based data security intrusion detection system that integrates the Mamba model and the ECANet model, employing an end-to-end learning approach for training and optimization. First, the Mamba model is utilized for initial data feature extraction, offering efficient feature representation that lays a strong foundation for the detection process. Next, the ECANet model is incorporated to optimize feature selection using the attention mechanism, allowing the model to focus on the most critical features. Finally, the entire system is trained and optimized through an end-to-end learning approach, ensuring robust performance and reliability in real-world applications. Experimental results demonstrate that the proposed intrusion detection system achieves higher detection accuracy across various test datasets, with a 5% improvement over traditional methods, offering a novel and effective solution for data security.

Keywords: *Data Security; Anomaly Detection; Mamba Model; ECANet Model; End-to-End Learning; Feature Extraction.*

1. **Introduction**

In the realm of cybersecurity, intrusion detection systems (IDS) are crucial for protecting networks from malicious attacks, enhancing overall system security, reducing the risk of data breaches, and ensuring data integrity. With the continuous advancement of information technology and the increasing complexity of network environments, the demand for automated and intelligent cybersecurity protection systems has become more urgent. These systems not only help to reduce the likelihood of potential threats but also demonstrate superior performance in enhancing network defense capabilities and handling complex network environments.

Deep learning-based models, such as Convolutional Neural Networks (CNNs) (Kim et al., 2020) and Long Short-Term Memory Networks (LSTMs) (Imrana et al., 2021), have achieved remarkable results in intrusion detection. CNNs utilize their weight-sharing characteristics to efficiently extract important features from network data, thereby accelerating processing speed, while LSTMs maintain long-term temporal relationships between data features. However, these models still underperform when dealing with limited or highly imbalanced intrusion samples. Additionally, these models typically rely on large amounts of labeled data to achieve high performance, which is often challenging to obtain in real-world cybersecurity environments.

State Space Models (SSMs) (Toorani & Beheshti, 2008), known for their efficiency in handling long sequence modeling, have recently been applied in the field of cybersecurity. For instance, the Mamba model, by introducing a data-dependent selection mechanism (Waleffe et al., 2024), significantly improves the model's efficiency and accuracy while maintaining linear scalability in processing long sequences. In the field of computer vision, variants of the Mamba model, such as VMamba (Shi et al., 2024), combine selective scanning mechanisms (S6) (Jia et al., 2021) to handle non-causal two-dimensional image data, further enhancing the model's processing capabilities. This innovative design not only improves detection accuracy but also significantly reduces computational costs. Moreover, the application of the Mamba model in multi-class unsupervised anomaly detection (MUAD) demonstrates its powerful modeling capability and computational efficiency, providing new solutions for intrusion detection in complex network environments. The unique aspect of the Jamba model lies in its integration of the Transformer (Han et al., 2022) and Mamba architectures. Although the Transformer is popular in the field of language modeling, its memory and computational requirements are high, and it is limited by the key-value cache size when handling long contexts. Additionally, generating each token requires computing the entire context, resulting in slow inference speed and low throughput. In contrast, traditional Recurrent Neural Networks (RNNs) (Yin et al., 2017) can summarize arbitrarily long contexts in a single hidden state without these limitations, but they are expensive to train and struggle to handle long-distance relationships.

Despite the significant advancements of these models in many areas, some unresolved issues persist in specific cybersecurity scenarios. To address these challenges, this paper proposes an end-to-end data security intrusion detection system that combines the Mamba model with the ECANet model, aiming to improve detection accuracy and efficiency. First, the Mamba model, through its selective state space model (SSMs) approach, addresses the weaknesses of traditional discrete modalities and designs hardware-friendly parallel algorithms, achieving efficient inference and linear scalability, suitable for analyzing large and complex log data in intrusion detection systems. Second, to further enhance the model's detection performance, this paper introduces the Efficient Channel Attention (ECA) module. The ECA module effectively reduces model complexity while improving the model's sensitivity and accuracy to abnormal behavior by avoiding dimensionality reduction and adopting a local cross-channel interaction strategy. Finally, through an end-to-end learning approach, this paper designs a complete intrusion detection framework that automatically performs data preprocessing, feature extraction, anomaly detection, and classification tasks, achieving efficient detection and classification of various types of attack behaviors.

The organization structure of this article is as follows: This section introduces the importance of cybersecurity and intrusion detection systems (IDS), provides an overview of the limitations and challenges of current intrusion detection methods, and presents the motivation and objectives of this study. The second section reviews recent research in the field of intrusion detection, covering both traditional methods and deep learning-based approaches, with a focus on the application and advantages and disadvantages of various models in different scenarios. The third section provides a detailed description of the proposed end-to-end data security intrusion detection system based on deep learning, including the design and integration of the Mamba model and ECANet model, as well as the application of end-to-end learning methods. The fourth section describes the experimental setup and procedures, including the selection of datasets, configuration of the experimental environment, and definition of evaluation metrics, and validates the effectiveness and superiority of the proposed system through comparisons with existing methods. The fifth section summarizes the main contributions and experimental findings of this paper, discusses the limitations of the research, and outlines future research directions. The main contributions of this paper are as follows:

1. The application of the Mamba model to the field of data security intrusion detection. This model, through its Selective State Space Model (SSM) approach, effectively addresses the weaknesses of traditional discrete modalities and designs hardware-friendly parallel algorithms, achieving efficient inference and linear scalability, suitable for analyzing large and complex log data in intrusion detection systems.

2. The introduction of the Efficient Channel Attention (ECA) module. By avoiding dimensionality reduction and adopting a local cross-channel interaction strategy, this module effectively reduces the complexity of the model while improving its sensitivity and accuracy in detecting abnormal behavior. This combination enhances the model's robustness and precision in handling diverse and complex attacks.

3. The design of a complete end-to-end intrusion detection framework capable of automatically performing data preprocessing, feature extraction, anomaly detection, and classification tasks. The end-to-end learning approach ensures that the optimization process of the entire system is global, improving the overall performance and robustness of the system.

2. **Related Work**

In recent years, intrusion detection systems (IDS) in cybersecurity have played a crucial role in addressing increasingly complex network threats. Traditional IDS methods are mainly divided into two categories: signature-based intrusion detection systems (SIDS) and anomaly-based intrusion detection systems (AIDS) (Khraisat et al., 2019). SIDS detect intrusions by matching the signatures of known attacks, offering high detection accuracy and low false alarm rates, but are limited in effectiveness against unknown attacks. AIDS, on the other hand, detect abnormal activities that deviate from expected behavior by constructing models of normal behavior, effectively identifying unknown attacks but potentially generating higher false alarm rates.

With the rapid development of machine learning and deep learning technologies, many researchers have applied these technologies to intrusion detection systems to improve their detection performance and ability to handle complex attacks. For example, Çavuşoğlu (C¸ avus¸o˘glu, 2019) proposed a new hybrid method that combines various machine learning techniques to enhance the accuracy and efficiency of intrusion detection. However, this method still faces challenges when dealing with large-scale high-dimensional data. Ferrag et al. (Aldhaheri et al., 2023) reviewed various deep learning-based intrusion detection methods, finding that these methods perform well in handling complex network traffic and evolving attack techniques, but are highly dependent on training datasets and may face issues with frequent model updates in practical applications. Longlong Li et al. (Li et al., 2024) proposed an end-to-end intrusion detection framework based on contrastive learning, employing hierarchical convolutional neural networks (CNNs) and gated recurrent units (GRUs) to automatically extract spatiotemporal features from raw network traffic data. This method achieved a detection accuracy of 99.9% for known attacks and a weighted recall rate of 95% for unknown attacks, demonstrating excellent detection capabilities.

In addition to the aforementioned studies, Yang and Wang (Yang & Wang, 2019) improved the application of convolutional neural networks (CNNs) for wireless network intrusion detection, significantly enhancing detection accuracy and efficiency. Jiang et al. (Jiang et al., 2016) proposed an energy-efficient multi-constrained routing algorithm in smart city applications, improving network efficiency through load balancing. Although this method mainly targets routing optimization, its approach is insightful for resource management in intrusion detection systems. Moreover, Dong and Wang (Dong & Wang, 2016) compared traditional methods and deep learning methods in network intrusion detection, finding that deep learning methods perform better in handling complex network traffic and unknown attacks. Sarvari et al. (Sarvari et al., 2020) proposed an efficient anomaly intrusion detection method by combining feature selection and evolutionary neural networks, significantly improving detection accuracy. Tian et al. (Tian et al., 2020) proposed an industrial network intrusion detection algorithm based on a multi-feature data clustering optimization model, demonstrating the potential of data fusion technology in enhancing detection performance. However, these methods still face challenges such as high dependency on datasets and frequent model updates, particularly in real-world applications where obtaining representative high-quality datasets remains a significant challenge.

In the study of deep learning-based end-to-end data security intrusion detection systems, the Mamba model, as a novel selective state space model (SSM), has shown significant advantages and potential. Albert Gu and Tri Dao (Gu & Dao, 2023) first proposed the Mamba model to address the computational efficiency issues of the Transformer architecture when processing long sequences. The Mamba model sets the SSM parameters as functions of the input, allowing the model to selectively propagate or forget information based on the current input, achieving linear time complexity expansion while maintaining context-relevant reasoning capabilities. Based on the Mamba architecture, researchers developed the Vision Mamba (Vim) (Xu et al., 2024) model for efficient visual representation learning. The Vim model, through bidirectional state space modeling and positional embedding techniques, performs excellently in image classification, object detection, and semantic segmentation tasks, significantly improving computational and memory efficiency. In the ImageNet classification task, the Vim model outperformed many existing visual Transformer models and demonstrated outstanding efficiency in high-resolution image processing. Additionally, Gu et al. (Gu & Dao, 2023) further extended the Mamba model by introducing the Selective State Space model (S6), enhancing the selectivity of information processing, resulting in superior performance in handling long-sequence data. This improvement has made the Mamba model more stable and efficient in multimodal learning tasks. However, the Mamba model still has some limitations. For example, the selective information processing mechanism of the Mamba model in handling discrete modalities (such as language) may lead to adaptability issues in specific application scenarios. Despite its hardware-friendly parallel algorithms enhancing computational efficiency, further optimization may be needed to avoid potential computational bottlenecks when processing extremely long sequence data.

Channel attention mechanisms have shown great potential in enhancing the performance of deep convolutional neural networks (CNNs). However, most existing methods focus on developing more complex attention modules to pursue better performance, inevitably increasing model complexity. To address the trade-off between performance and complexity, Wang et al. (Wang et al., 2020) proposed an Efficient Channel Attention (ECA) module. The ECA module effectively reduces model complexity while significantly improving performance by avoiding dimensionality reduction and adopting a local cross-channel interaction strategy. For instance, experiments on ResNet-50 demonstrated that the ECA module could achieve over a 2% increase in Top-1 accuracy with minimal additional parameters and computational load. Besides the work of Wang et al., Huynh-The et al. (Huynh-The et al., 2022) proposed a high-performance convolutional network (RF-UAVNET) based on the ECA attention mechanism for RF signal-based UAV monitoring systems. Their method integrated the ECA module to improve accuracy and efficiency in UAV detection and identification tasks. However, while their method performed well in specific applications, further validation of the model's generalization ability in handling other types of complex data is necessary. Huang et al. (Huang et al., 2021) explored the application of the ECA attention mechanism in multi-channel 1D convolutional neural networks for UAV detection and identification using RF signals. Their method highlighted the advantages of the ECA module in processing time-series data by reducing redundant computations to enhance real-time performance. However, the study also noted that the ECA module still faces challenges when dealing with high-noise and complex background data. Additionally, Chen et al. (Chen et al., 2022) proposed a deep learning method combining the ECA module for UAV detection and classification. Their method leveraged the ECA attention mechanism to enhance feature extraction capability in complex scenarios, effectively improving detection and classification accuracy. Nonetheless, further optimization is needed to address computational efficiency and resource consumption when handling larger datasets. ECA-Net, through its innovative design, strikes a balance between complexity and performance, providing new insights for performance enhancement in deep learning models.

3. **Method**

Figure 1 shows the overall algorithm architecture of the data security intrusion detection system used in this paper. This model first performs a linear projection on the input data and transforms it into both the frequency and time domains to capture multi-dimensional features. Subsequently, it utilizes ECANet for feature extraction and further optimizes the extracted features through a Selective State Space Model. The optimized features are then combined via linear projection and processed through the subsequent Add & Norm and Feed-forward mechanisms to ensure stable signal processing and feature extraction, ultimately outputting the detection results.

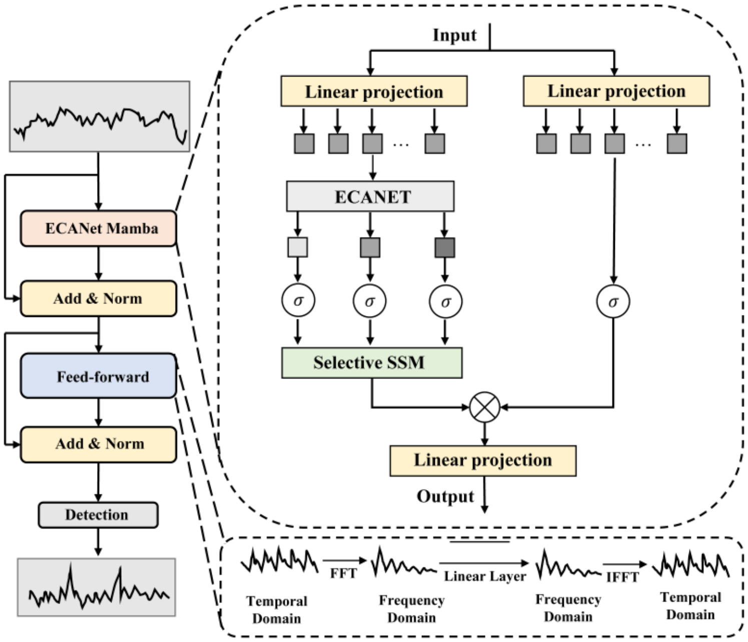

Figure 1. Overall algorithm architecture.

3.1 Mamba Architecture

Mamba is a Selective Structured State Space Model (S4) designed to handle long-sequence data. By introducing a selection mechanism, it overcomes the limitations of traditional State Space Models (SSMs) in contextual reasoning capability. The Mamba model effectively extracts complex data features, providing a solid foundation for subsequent deep learning processing. The architecture diagram of Mamba is shown in Figure 2.

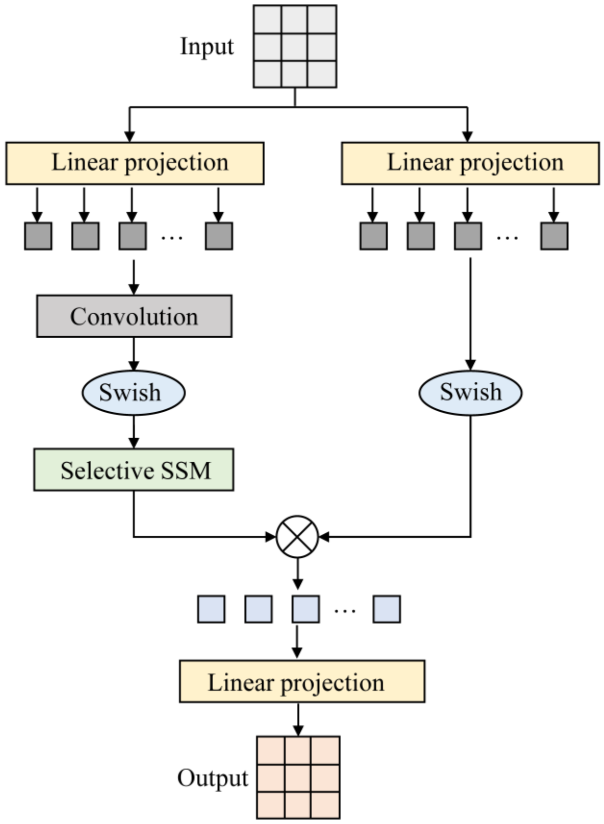

Figure 2. Structure diagram of Mamba.

The core idea of SSM is to connect the input and output sequences through latent states. The classic form of SSM is as follows:

$$h'(t) = Ah(t) + Bx(t) \tag{1}$$

$$y(t) = Ch(t) \tag{2}$$

where $A \in R^{N\times N}, B \in R^{N\times 1}, C \in R^{1\times N}$ are the model parameters. When processing discrete input sequences, SSM discretizes these parameters using the zero-order hold (ZOH) method. The discretized parameters are expressed as:

$$A = \exp(\Delta A) \tag{3}$$

$$B = (\Delta A)^{-1}(\exp(\Delta A) - I) \cdot \Delta B \tag{4}$$

Then, the discretized SSM is represented as:：

$$h_t = Ah_{t-1} + Bx_t, \tag{5}$$

$$y_t = Ch_t, \tag{6}$$

The recursive computation process of SSM can also be expressed as a convolution operation:

$$K = (CB, CAB, \ldots, CA^{L-1}B), \tag{7}$$

$$y = x * K, \tag{8}$$

where L is the length of the input sequence and $K \in R^{L}$ is the SSM convolution kernel.The key improvement of the Mamba model lies in its selection mechanism, which achieves context-related interaction by making the parameters of the SSM dependent on the input sequence. Specifically, the parameters B, C, Δ of the selective SSM are expressed as functions of the input sequence x:

$$B, C, \Delta = \text{Linear}(x), \tag{9}$$

3.2 ECANet Architecture

In this study, we adopt ECANet (Efficient Channel Attention Network) as one of the base models to enhance the feature selection capability of the data security intrusion detection system. ECANet, by introducing an efficient channel attention mechanism, improves model performance while reducing computational complexity. The channel attention mechanism aims to weight the channels of the input feature map, enabling the network to focus more on important features. The architecture diagram of ECANet is shown in Figure 3

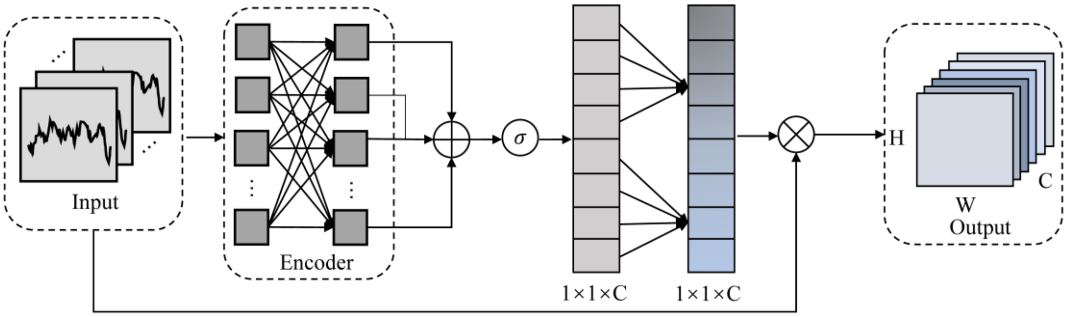

Figure 3. Structure diagram of ECANet.

The core idea of ECANet is to use one-dimensional convolution (1D Convolution) instead of fully connected layers to capture local cross-channel interactions. First, the input feature map $X \in R^{H\times W\times C}$ is globally average-pooled to obtain the global feature vector $z \in R^{C}$ along the channel dimension.

$$\mathbf{z} = F_{sq}(\mathbf{X}) = \frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\mathbf{X}(i, j) \tag{10}$$

Then, a one-dimensional convolution kernel of size k is applied to the global feature vector z, generating the channel attention weights $s \in R^{C}$. Here, k is an adjustable parameter used to control the size of the convolution kernel.

$$\mathbf{s} = F_{ex}(\mathbf{z}) = \sigma(\text{Conv1D}(\mathbf{z}, k)) \tag{11}$$

Finally, the generated channel attention weights s are multiplied with the original feature map X, resulting in the re-calibrated feature map $Y \in R^{H \times W \times C}$.

$$\mathbf{Y} = F_{scale}(\mathbf{X}, \mathbf{s}) = \mathbf{s} \cdot \mathbf{X} \tag{12}$$

where σ is the sigmoid activation function, and $\mathrm{Conv1D}(z, k)$ denotes convolution operation with a one-dimensional kernel of size k applied to the feature vector z.

3.3 End-to-End Learning

End-to-end learning refers to integrating the entire learning process into a single model, optimizing directly from the raw input to the final output. Compared to traditional staged learning methods, end-to-end learning can better capture the global features of the data, reduce information loss from intermediate steps, and enhance the overall performance and robustness of the model.

We use the cross-entropy loss function to measure the discrepancy between the model's predictions and the true labels. The formula for the cross-entropy loss function is as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)\right], \tag{13}$$

where N is the number of samples, y_i is the true label of the i-th sample, and $\hat{y}_i$ is the predicted probability from the model. We use the Adam optimizer to update the model parameters. The Adam optimizer can efficiently process large-scale data and high-dimensional parameter space through adaptive learning rate adjustment. Its update formula is as follows:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{14}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{15}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{16}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{17}$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{18}$$

Among them, m_t and v_t are the first-order and second-order moment estimates, respectively, β_1 and β_2 are the decay rates, α is the learning rate, g_t is the gradient, and θ_t is the model parameter. We divide the training data into training set and validation set, and continuously adjust the model parameters through iterative optimization until the loss function converges. During the training process, the Early Stopping method is used to prevent overfitting.

The pseudo code of this model is as follows:

Algorithm 1 Training Process for Mamba-ECANet

```
NSL-KDD Dataset, UNSW-NB15 Dataset, CICIDS 2017 Dataset, AWID Dataset Trained Mamba-ECANet Model, Performance Metrics: Accuracy, Precision, Recall, AUC Initialize parameters $\theta_{Mamba}$, $\theta_{ECANet}$, learning rate $\alpha$, batch size $B$, epochs $E$
Load datasets: $D_{NSL-KDD}$, $D_{UNSW-NB15}$, $D_{CICIDS2017}$, $D_{AWID}$
for each dataset $D$ in $\{D_{\mathrm{NSL-KDD}}, D_{UNSW-NB15}, D_{CICIDS2017}, D_{AWID}\}$ do
    Split $D$ into training set $D_{train}$ and test set $D_{test}$
    for epoch $e$ in $1$ to $E$ do
        for batch $b$ in $D_{train}$ do
            Extract features $X_b$ and labels $y_b$ from batch $b$
                $h_{\mathrm{Mamba}} \leftarrow \mathrm{MambaModel}(X_b, \theta_{\mathrm{Mamba}})$
                $h_{\mathrm{ECANet}} \leftarrow \mathrm{ECANetModel}(h_{\mathrm{Mamba}}, \theta_{\mathrm{ECANet}})$
                $y_{\mathrm{pred}} \leftarrow \mathrm{Softmax}(h_{\mathrm{ECANet}})$
            Compute loss $\mathcal{L}(y_{\mathrm{pred}}, y_b)$ using cross-entropy:
                $\mathcal{L}(y_{\mathrm{pred}}, y_b) = -\frac{1}{B}\sum_{i=1}^{B} y_b^{(i)} \log(y_{\mathrm{pred}}^{(i)})$
            Compute gradients $\nabla_{\theta_{\mathrm{Mamba}}}\mathcal{L}, \nabla_{\theta_{\mathrm{ECANet}}}\mathcal{L}$
            Update parameters:
                $\theta_{\mathrm{Mamba}} \leftarrow \theta_{\mathrm{Mamba}} - \alpha\nabla_{\theta_{\mathrm{Mamba}}}\mathcal{L}$
                $\theta_{\mathrm{ECANet}} \leftarrow \theta_{\mathrm{ECANet}} - \alpha\nabla_{\theta_{\mathrm{ECANet}}}\mathcal{L}$
        end
    end
    Evaluate model on $D_{\mathrm{test}}$
    for each sample$(x, y)$ in $D_{test}$ do
        $h_{\mathrm{Mamba}} \leftarrow \mathrm{MambaModel}(x, \theta_{\mathrm{Mamba}})$
        $h_{\mathrm{ECANet}} \leftarrow \mathrm{ECANetModel}(h_{Mamba}, \theta_{ECANet})$
        $y_{\mathrm{pred}} \leftarrow \mathrm{Softmax}(h_{\mathrm{ECANet}})$
        Collect predictions and true labels
    end
    Compute evaluation metrics:
                $\mathrm{Accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$
                $\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$
        $\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$
        $\text{AUC} = \int_0^1 TPR(FPR)\ d(FPR)$
    Store results for dataset $D$
end
Compare metrics across datasets and finalize model
```

4. **Experiment**

The experimental flow chart of this paper is shown in Figure 4.

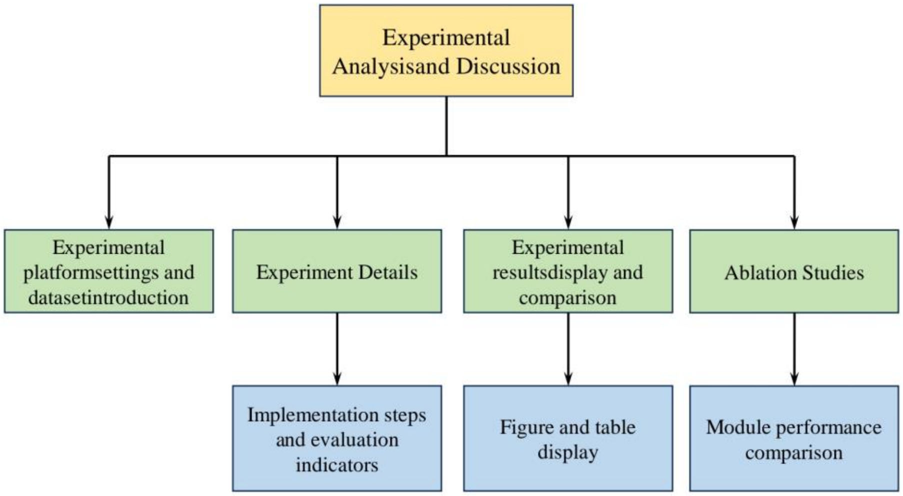
Experimental Analysisand Discussion
Experimental platformsettings and datasetintroduction
Experiment Details
Experimental resultsdisplay and comparison
Ablation Studies
Implementation steps and evaluation indicators
Figure and table display
Module performance comparison

Figure 4. Experimental flowchart.

4.1 Experimental Environment

The experiments were conducted on a high-performance computing platform to ensure the efficiency of data processing and model training. The hardware environment includes: Intel Core i9-10900K processor with 10 cores and 20 threads; NVIDIA GeForce RTX 3090 graphics processor with 24GB video memory; 256GB DDR4 memory; and 2TB NVMe SSD storage. The software environment includes: Ubuntu 20.04 LTS operating system; TensorFlow 2.4 and PyTorch 1.7.1 deep learning frameworks for model building and training; NumPy 1.19.4 and Pandas 1.1.4 data processing libraries for data preprocessing and analysis; Matplotlib 3.3.3 and Seaborn 0.11.0 visualization tools for visualization and analysis of results.

4.2 Experimental Data

- NSL-KDD Dataset

The NSL-KDD dataset is an improved version of the KDD Cup 1999 dataset, designed to address the redundancy issues in the original dataset. It contains 125,973 training records and 22,544 testing records. The records in the NSL-KDD dataset are categorized into normal traffic and various types of attack traffic, including DoS (Denial of Service), Probe, U2R (User to Root), and R2L (Remote to Local). The NSL-KDD dataset is widely used for research and evaluation of network intrusion detection systems because it provides a standardized testing platform and has a relatively small number of records, making it suitable for preliminary validation and comparative experiments.

- UNSW-NB15 Dataset

The UNSW-NB15 dataset, released by the University of New South Wales (UNSW), contains real network traffic and various attack traffic. This dataset includes 100,000 training records and 82,332 testing records. It covers 9 types of attacks, including Analysis, Backdoor, DoS (Denial of Service), Exploits, Fuzzers, Malware, Shellcode, Worms, and Generic. The diversity and complexity of the UNSW-NB15 dataset make it an important tool for evaluating the effectiveness of intrusion detection systems in real-world network environments.

- CICIDS 2017 Dataset

The CICIDS 2017 dataset, released by the Canadian Institute for Cybersecurity (CIC), contains network traffic data generated in 2017. This dataset records various types of attacks, including DDoS (Distributed Denial of Service), Brute Force, XSS (Cross-Site Scripting), SQL Injection, and more. The CICIDS 2017 dataset encompasses a wide range of network activities from normal traffic to complex attack scenarios, making it a valuable resource for studying and evaluating the performance of intrusion detection systems. Its detailed traffic records and diverse attack types aid in the application research of deep learning models in real-world scenarios.

- AWID Dataset

The AWID (Aegean Wi-Fi Intrusion Dataset) dataset focuses on intrusion detection in wireless networks and is released by TU Wien (Vienna University of Technology). This dataset includes normal traffic and various types of wireless attack traffic, primarily used for researching wireless network security and intrusion detection. The AWID dataset is divided into sub-datasets, including AWID-ATK-R and AWID-CLS, where AWID-ATK-R is mainly used for identifying attack types, and AWID-CLS is used for classifying attacks and normal traffic. The dataset's focus on the wireless network environment makes it suitable for evaluating the performance of wireless network intrusion detection systems.

4.3 Evaluation Metrics

In evaluating the proposed deep learning-based data security intrusion detection system, the following four main metrics are used to measure the performance of the model: Accuracy, Precision, Recall, and AUC. These metrics help comprehensively assess the detection capability and robustness of the system, ensuring it can effectively detect and defend against data security intrusions in practical applications.

- Accuracy

Accuracy is an intuitive metric that measures the overall prediction performance of the model. It represents the proportion of correctly predicted samples out of the total number of samples. For an intrusion detection system, high accuracy means the model can correctly classify normal traffic and attack traffic in most cases.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{19}$$

where TP represents True Positives, the correctly detected attack traffic; TN represents True Negatives, the correctly detected normal traffic; FP represents False Positives, the normal traffic mistakenly detected as attack traffic; and FN represents False Negatives, the attack traffic mistakenly detected as normal traffic.

- Precision：

Precision measures the proportion of actual attack traffic out of all samples predicted as attack traffic. High precision indicates that the model rarely misclassifies normal traffic as attack traffic, which is crucial for reducing interference with normal traffic.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{20}$$

- Recall：

Recall measures the proportion of correctly detected attack traffic out of all actual attack traffic samples. High recall means the model can effectively detect most attack traffic, reducing the risk of missed attacks.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{21}$$

- AUC：

In a data security intrusion detection system, AUC helps evaluate the system's detection performance for intrusion and non-intrusion behavior at different thresholds. By calculating the AUC, we can quantify the model's overall classification performance, ensuring the system has good detection capability and robustness in practical applications. The higher the AUC value, the better the classification performance in detecting intrusion and non-intrusion behavior.

$$AUC = \int_0^1 TPR(FPR)d(FPR) \tag{22}$$

Here, TPR stands for True Positive Rate, and FPR stands for False Positive Rate.

4.4 Experimental Comparison and Analysis

To validate the effectiveness of the proposed deep learning-based data security intrusion detection system, we conducted extensive experimental comparisons and analyses. The experiments utilized multiple public network security datasets, including NSL-KDD, UNSW-NB15, CICIDS 2017, and AWID, and were compared with traditional intrusion detection systems.

Table 1 shows the performance comparison of different models on the NSL-KDD and UNSW-NB15 datasets. It is evident that the proposed intrusion detection system significantly outperforms other models across all evaluation metrics. On the NSL-KDD dataset, our model achieved an accuracy of 96.45%, a precision of 97.64%, a recall of 96.14%, and an AUC of 97.64%. These results are substantially higher than those of Yang et al. (89.90% accuracy), Sarvari et al. (89.21% accuracy), and other comparison models. Similarly, on the UNSW-NB15 dataset, our model demonstrated excellent performance, achieving an accuracy of 95.64%, a precision of 97.54%, a recall of 96.73%, and an AUC of 96.76%, significantly surpassing other models. This indicates that the proposed system offers higher detection accuracy, lower false positive rates, and greater robustness in practical applications, effectively enhancing the overall performance of data security intrusion detection. Figure 5 visualizes the comparison of various metrics on the two datasets.

Table 1. Comparison of indicators of various models under NSL-KDD Dataset and UNSW-NB15 Dataset.

Model	NSL-KDD Dataset				UNSW-NB15 Dataset			
	ACC(%)	P(%)	R(%)	AUC(%)	ACC(%)	P(%)	R(%)	AUC(%)
Yang et al. (Yang & Wang, 2019)	89.90	89.10	91.34	90.88	89.66	92.44	89.87	89.04
Sarvari et al. (Sarvari et al., 2020)	89.21	90.87	89.68	93.63	89.63	91.16	90.40	91.60
Wang et al. (Wang et al., 2020)	90.07	89.68	90.84	91.81	88.21	89.73	89.94	88.08
Huynh-The et al. (Huynh-The et al., 2022)	89.67	92.27	92.20	89.06	88.30	91.51	91.09	89.42
Huang et al. (Huang et al., 2021)	92.27	89.63	90.92	93.79	90.06	90.46	90.67	90.42
Chen et al. (Chen et al., 2022)	92.14	89.99	92.66	92.76	90.94	89.48	92.39	89.55
Ours	96.45	97.64	96.14	97.64	95.64	97.54	96.73	96.76

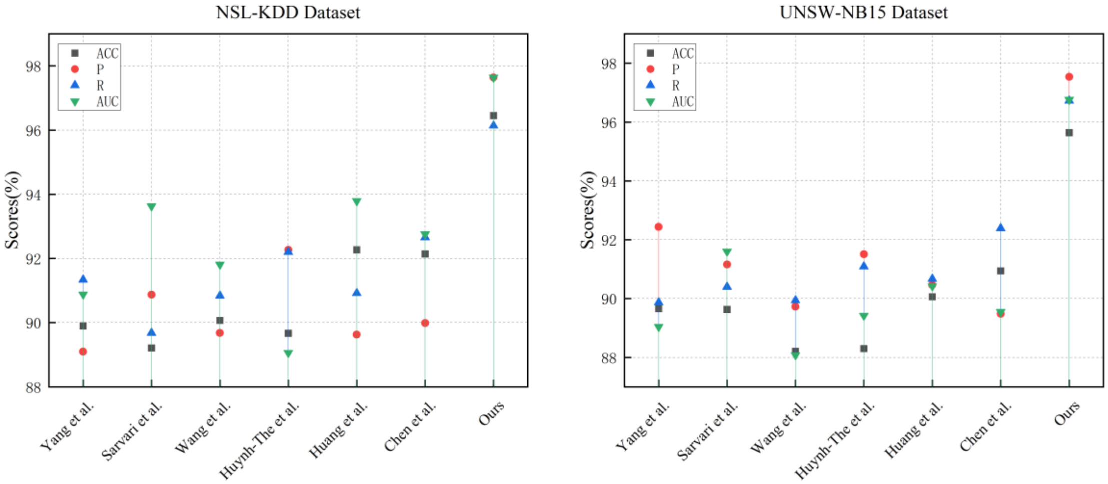

Figure 5. Comparative visualization of each model indicator under the NSL-KDD Dataset and UNSW-NB15 Dataset.

Table 2. Comparison of indicators of various models under the CICIDS 2017 Dataset and AWID Dataset.

Model	CICIDS 2017 Dataset				AWID Dataset			
	ACC(%)	P(%)	R(%)	AUC(%)	ACC(%)	P(%)	R(%)	AUC(%)
Yang et al. (Yang & Wang, 2019)	91.30	89.24	89.29	91.92	91.82	90.47	88.74	91.38
Sarvari et al. (Sarvari et al., 2020)	93.39	90.43	90.39	91.42	89.00	92.81	89.47	89.42
Wang et al. (Wang et al., 2020)	93.81	92.16	91.97	89.5	90.87	89.04	91.21	92.42
Huynh-The et al. (Huynh-The et al., 2022)	92.02	91.03	91.67	89.79	91.65	90.30	91.70	89.29
Huang et al. (Huang et al., 2021)	91.04	92.63	90.56	92.97	92.48	92.05	91.56	92.30
Chen et al. (Chen et al., 2022)	93.85	92.83	90.44	90.01	93.00	91.09	88.32	90.48

Ours	97.64	95.21	97.72	98.09	96.41	96.37	95.34	97.54

Table 2 presents the performance comparison of different models on the CICIDS 2017 and AWID datasets. The results indicate that our proposed intrusion detection system outperforms other models across all evaluation metrics. On the CICIDS 2017 dataset, our model achieved an accuracy of 97.64%, a precision of 95.21%, a recall of 97.72%, and an AUC of 98.09%, significantly higher than Sarvari et al. Similarly, on the AWID dataset, our model also demonstrated excellent performance, achieving an accuracy of 96.41%, a precision of 96.37%, a recall of 95.34%, and an AUC of 97.54%, all of which significantly surpass other models. These results show that our deep learning intrusion detection system has higher detection accuracy and robustness in dealing with different types of network attacks and datasets, effectively enhancing data security protection capabilities. Similarly, Figure 6 visualizes the comparison of various metrics on the two datasets.

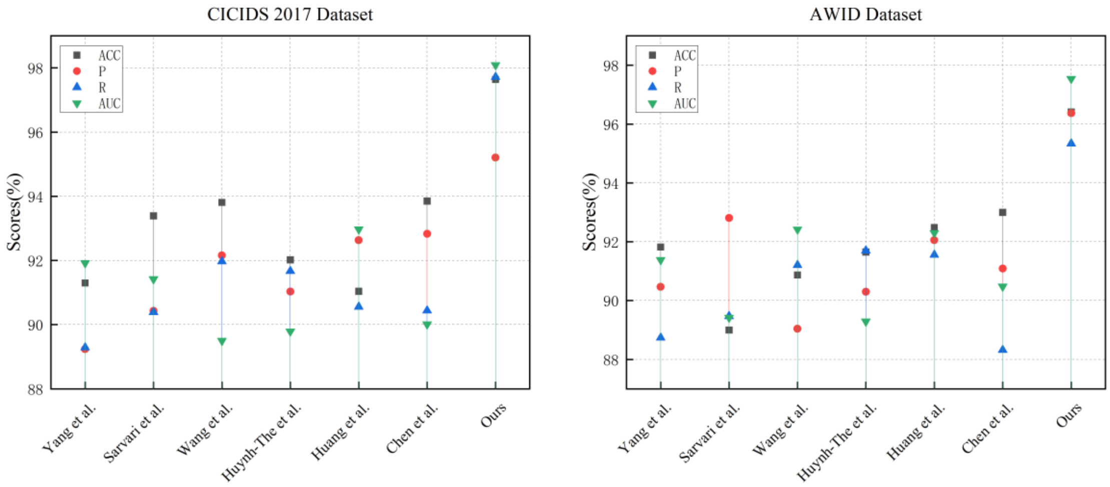

Figure 6. Comparative visualization of each model indicator under the CICIDS 2017 Dataset and AWID Dataset.

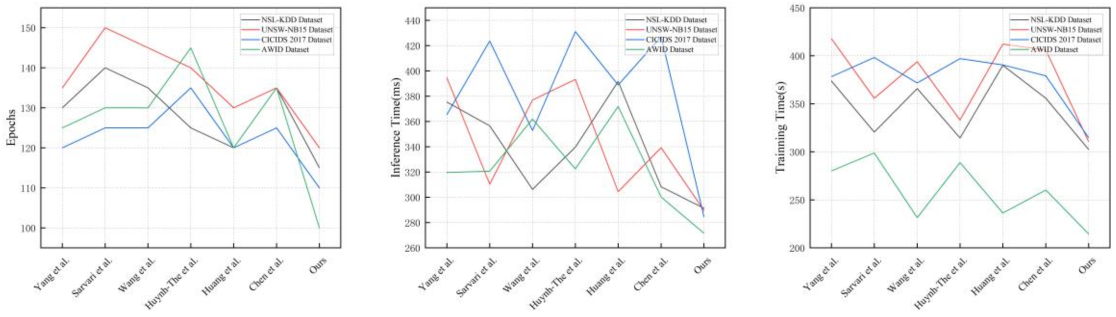

Figure 7. Visual comparison of training indicators of multiple models on four datasets.

Table 3 shows the training metrics of various models on four datasets, including the number of training epochs, inference time, and training time. The results indicate that our proposed model demonstrates superior training efficiency across all datasets. On the NSL-KDD and UNSW-NB15 datasets, our model requires only 115 and 120 epochs, respectively, with inference times of 291.34ms and 289.64ms and training times of 302.42s and 310.51s, respectively, all of which are the lowest values. On the CICIDS 2017 and AWID datasets, our model also performs excellently, requiring 110 and 100 epochs, respectively, with inference times of 284.64ms and 271.64ms and training times of 314.62s and 214.53s. This indicates that our proposed system not only outperforms other models in terms of performance but also shows significant advantages in training and inference efficiency, enabling it to quickly and effectively adapt to the needs of real-world applications. Figure 7 visualizes the comparison of various training metrics across the four datasets.

Table 3. Training indicators of each model on four datasets.

	NSL-KDD Dataset			UNSW-NB15 Dataset		
Model	Epochs	Inference Time(ms)	Trainning Time(s)	Epochs	Inference Time(ms)	Trainning Time(s)
Yang et al. (Yang & Wang, 2019)	130	375.39	373.75	135	394.64	417.77
Sarvari et al. (Sarvari et al., 2020)	140	356.58	320.48	150	310.42	355.73
Wang et al. (Wang et al., 2020)	135	306.32	365.86	145	377.05	393.72
Huynh-The et al. (Huynh-The et al., 2022)	125	339.71	314.41	140	393.35	333.17
Huang et al. (Huang et al., 2021)	120	391.74	390.09	130	304.43	412.11
Chen et al. (Chen et al., 2022)	135	308.32	355.8	135	339.3	405.99
Ours	115	291.34	302.42	120	289.64	310.51
	CICIDS 2017 Dataset			AWID Dataset		
Model	Epochs	Inference Time(ms)	Trainning Time(s)	Epochs	Inference Time(ms)	Trainning Time(s)
Yang et al. (Yang & Wang, 2019)	120	365.43	378.21	125	319.66	280.18
Sarvari et al. (Sarvari et al., 2020)	125	423.66	398.24	130	320.71	298.8
Wang et al. (Wang et al., 2020)	125	352.85	371.71	130	361.83	231.54
Huynh-The et al. (Huynh-The et al., 2022)	135	431.2	396.99	145	322.5	288.83
Huang et al. (Huang et al., 2021)	120	388.87	390.51	120	371.94	236.26
Chen et al. (Chen et al., 2022)	125	427.2	379.18	135	300.21	260.27
Ours	110	284.64	314.62	100	271.64	214.53

Table 4 presents the ablation study results of our model on the NSL-KDD and UNSW-NB15 datasets, verifying the impact of different components on the model's performance. The baseline model shows relatively low performance, with an accuracy of 86.54% on the NSL-KDD dataset and 85.25% on the UNSW-NB15 dataset. After introducing the Mamba model, all metrics improved, with accuracy increasing to 89.31% and 88.67% on the two datasets, respectively. The addition of the ECANet model further enhanced performance, achieving accuracies of 93.06% and

91.74% on the NSL-KDD and UNSW-NB15 datasets, respectively. When both the Mamba and ECANet models were introduced simultaneously, the model performance reached its peak, with an accuracy of 96.45% on the NSL-KDD dataset and 95.64% on the UNSW-NB15 dataset, significantly improving all metrics. This demonstrates that the combination of Mamba and ECANet models significantly enhances the detection capability and robustness of the intrusion detection system. Figure 8 visualizes the comparison of the ablation study.

Table 4. Ablation experiments of this model on the NSL-KDD Dataset and UNSW-NB15 Dataset.

Model	Dataset							
	NSL-KDD Dataset				UNSW-NB15 Dataset			
	ACC(%)	P(%)	R(%)	AUC(%)	ACC(%)	P(%)	R(%)	AUC(%)
baseline	86.54	87.5	86.34	88.6	85.25	86.19	86.16	85.64
+Mamba	89.31	91.73	90.6	91.63	88.67	89.15	90.17	88.14
＋ECANet	93.06	92.74	92.49	93.46	91.74	93.58	92.37	91.47
+Mamba ECANet	96.45	97.64	96.14	97.64	95.64	97.54	96.73	96.76

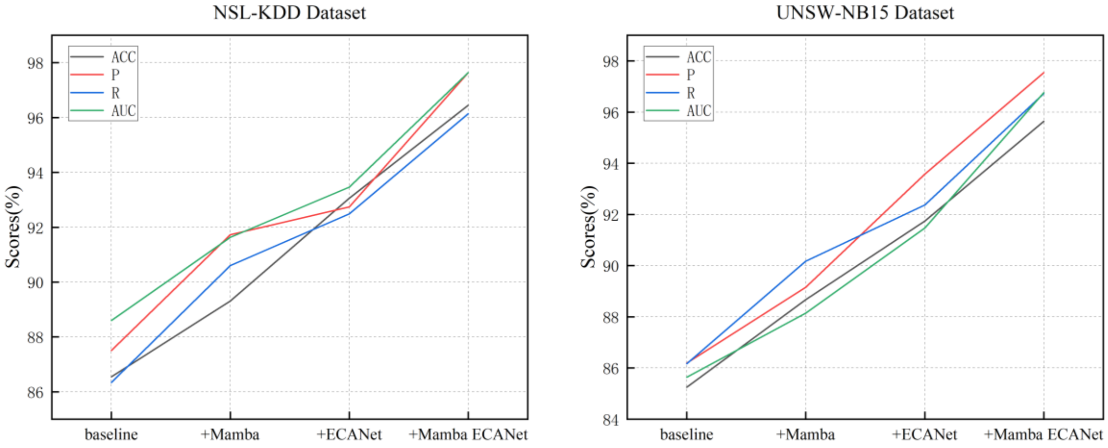

Figure 8. Comparative visualization of ablation experiments on NSL-KDD Dataset and UNSW-NB15 Dataset.

Table 5. Ablation experiments of this model on the CICIDS 2017 Dataset and AWID Dataset.

Model	Dataset							
	CICIDS 2017 Dataset				AWID Dataset			
	ACC(%)	P(%)	R(%)	AUC(%)	ACC(%)	P(%)	R(%)	AUC(%)
baseline	86.14	87.71	84.46	86.14	87.34	85.49	87.36	86.46
+Mamba	89.48	90.14	88.17	91.74	90.6	88.49	89.06	91.73
+ECANet	92.17	91.61	93.63	94.1	92.8	91.4	91.8	93.94
+Mamba ECANet	97.64	95.21	97.72	98.09	96.41	96.37	95.34	97.54

Table 5 shows the ablation study results of our model on the CICIDS 2017 and AWID datasets, further verifying the impact of different components on the model's performance. The baseline model shows relatively modest performance, with an accuracy of 86.14% on the CICIDS 2017 dataset and 87.34% on the AWID dataset. After introducing the Mamba model, all metrics significantly improved, with accuracies increasing to 89.48% and 90.6% on the CICIDS 2017 and AWID datasets, respectively. The addition of the ECANet model further enhanced performance, achieving accuracies of 92.17% on the CICIDS 2017 dataset and 92.8% on the AWID dataset. When both the Mamba and ECANet models were introduced simultaneously, the model performance reached its peak, with an accuracy of 97.64% on the CICIDS 2017 dataset and 96.41% on the AWID dataset. These results demonstrate that the combination of Mamba and ECANet models significantly enhances the detection capability of the intrusion detection system, especially on complex datasets. Similarly, Figure 9 visualizes the comparison of the ablation study.

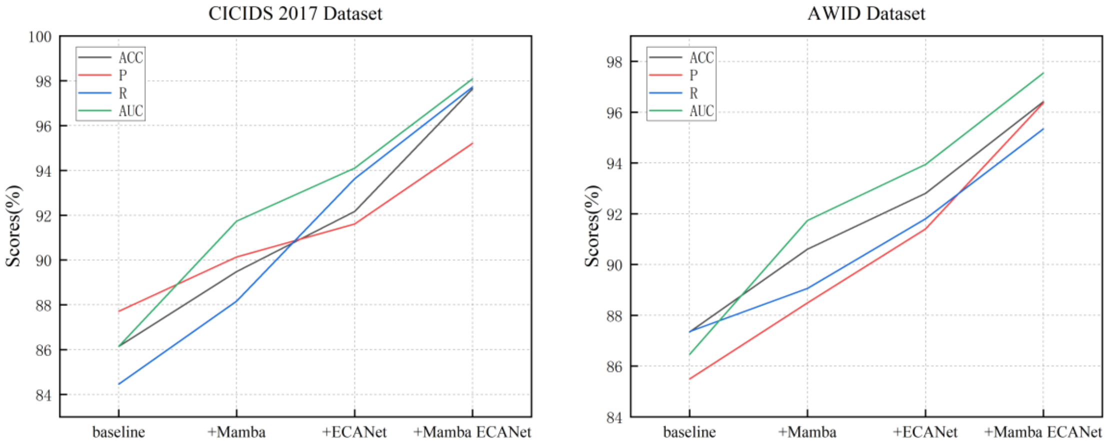

Figure 9. Comparative visualization of ablation experiments on CICIDS 2017 Dataset and AWID Dataset.

5. **Conclusion**

This paper proposes an end-to-end data security intrusion detection system based on deep learning, integrating the Mamba and ECANet models and employing end-to-end learning for training and optimization. By introducing the Mamba model, we effectively address the efficiency and accuracy issues of traditional methods in handling complex network data. The combination with the ECANet model further enhances feature selection through attention mechanisms, significantly improving the system's capability and accuracy in detecting anomalous behaviors. Experiments on multiple public datasets including NSL-KDD, UNSW-NB15, CICIDS 2017, and AWID validate the effectiveness and robustness of our approach, demonstrating the system's ability to maintain high detection performance across different network environments and attack types. Furthermore, through ablation studies, we further demonstrate the significant role of integrating the Mamba and ECANet models in enhancing system performance. Despite achieving satisfactory experimental results, there are still areas for further research. Future work could focus on optimizing the computational efficiency of the model to accommodate more complex and large-scale network environments. Exploring additional data augmentation techniques and unsupervised learning methods could reduce reliance on extensive labeled data. Applying the proposed method to more real-world scenarios would validate its generality and applicability across diverse network environments.

Funding
Not applicable

Author Contributions

Conceptualization, M. W. and N. Z.; writing—original draft preparation, H. Z. and N. Z.; writing—review and editing, M. W. and N. Z.; All of the authors read and agreed to the published the final manuscript.

Institutional Reviewer Board Statement
Not applicable

Informed Consent Statement
Not applicable

Data Availability Statement
Not applicable

Conflict of Interest

The authors declare no conflict of interest.

Reference

[1] Aldhaheri, A., Alwahedi, F., Ferrag, M. A., and Battah, A. (2023). Deep learning for cyber threat detection in iot networks: A review. Internet of Things and Cyber-Physical Systems

[2] C¸ avu s¸ o ˘glu, ˙U. (2019). A new hybrid approach for intrusion detection using machine learning methods. Applied Intelligence 49, 2735–2761

[3] Chen, H., Li, C., Li, X., Rahaman, M. M., Hu, W., Li, Y., et al. (2022). Il-mcam: An interactive learning and multi-channel attention mechanism-based weakly supervised colorectal histopathology image classification approach. Computers in Biology and Medicine 143, 105265

[4] Dong, B. and Wang, X. (2016). Comparison deep learning method to traditional methods using for network intrusion detection. In 2016 8th IEEE international conference on communication software and networks (ICCSN) (IEEE), 581–585

[5] Gu, A. and Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752

[6] Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., et al. (2022). A survey on vision transformer. IEEE transactions on pattern analysis and machine intelligence 45, 87–110

[7] Huang, H., Tang, B., Luo, J., Pu, H., and Zhang, K. (2021). Residual gated dynamic sparse network for gearbox fault diagnosis using multisensor data. IEEE Transactions on Industrial Informatics 18, 2264–2273

[8] Huynh-The, T., Pham, Q.-V., Nguyen, T.-V., Da Costa, D. B., and Kim, D.-S. (2022). Rf-uavnet: High-performance convolutional network for rf-based drone surveillance systems. IEEE Access 10, 49696–49707

[9] Imrana, Y., Xiang, Y., Ali, L., and Abdul-Rauf, Z. (2021). A bidirectional lstm deep learning approach for intrusion detection. Expert Systems with Applications 185, 115524

[10] Jia, H., Sun, H., Wang, H., Wu, Y., and Wang, H. (2021). Scanning strategy in selective laser melting (slm): a review. The International Journal of Advanced Manufacturing Technology 113, 2413–2435

[11] Jiang, D., Zhang, P., Lv, Z., and Song, H. (2016). Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications. IEEE Internet of Things Journal 3, 1437–1447

[12] Khraisat, A., Gondal, I., Vamplew, P., and Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity 2, 1–22

[13] Kim, J., Kim, J., Kim, H., Shim, M., and Choi, E. (2020). Cnn-based network intrusion detection against denial-of-service attacks. Electronics 9, 916

[14] Li, L., Lu, Y., Yang, G., and Yan, X. (2024). End-to-end network intrusion detection based on contrastive learning. Sensors 24, 2122

[15] Sarvari, S., Sani, N. F. M., Hanapi, Z. M., and Abdullah, M. T. (2020). An efficient anomaly intrusion detection method with feature selection and evolutionary neural network. IEEE Access 8, 70651–70663

[16] Shi, Y., Dong, M., and Xu, C. (2024). Multi-scale vmamba: Hierarchy in hierarchy visual state space model. arXiv preprint arXiv:2405.14174

[17] Tian, Q., Han, D., Li, K.-C., Liu, X., Duan, L., and Castiglione, A. (2020). An intrusion detection approach based on improved deep belief network. Applied Intelligence 50, 3162–3178

[18] Toorani, M. and Beheshti, A. (2008). Ssms-a secure sms messaging protocol for the m-payment systems. In 2008 IEEE Symposium on Computers and Communications (IEEE), 700–705

[19] Waleffe, R., Byeon, W., Riach, D., Norick, B., Korthikanti, V., Dao, T., et al. (2024). An empirical study of mamba-based language models. arXiv preprint arXiv:2406.07887

[20] Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., and Hu, Q. (2020). Eca-net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 11534–11542

[21] Xu, R., Yang, S., Wang, Y., Du, B., and Chen, H. (2024). A survey on vision mamba: Models, applications and challenges. arXiv preprint arXiv:2404.18861

[22] Yang, H. and Wang, F. (2019). Wireless network intrusion detection based on improved convolutional neural network. Ieee Access 7, 64366–64374

[23] Yin, W., Kann, K., Yu, M., and Sch ¨utze, H. (2017). Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923