Journal of Artificial Intelligence and Technology, 2026, 6, 399-409
https://doi.org/10.37965/jait.2026.0962

ISTP
RESEARCH ARTICLE

Entropy-Guided Gradient Pruning with Informer for Intrusion Detection System in Internet of Things

Yamuna Raju and Pushpa Chikkatotlikere Nagappa
Department of Computer Science and Engineering, University of Visvesvaraya College of Engineering, Bangalore, India

(Received 08 October 2025; Revised 02 February 2026; Accepted 06 March 2026; Published online 29 March 2026)

Abstract: The internet of things (IoT) is becoming increasingly significant in computer networks and applications. The existing deep learning (DL)-based intrusion detection system (IDS) suffers from high computational complexity and poor generalization, where irrelevant network flows are due to the inability of the filter. To address this challenge, this research proposes a novel entropy-guided gradient pruning (EGGP) with an Informer for IDS classification. The EGGP eliminates the less impactful and redundant network flows based on entropy and gradients that focus on significant traffic. The EGGP is integrated with the Informer backbone, which utilizes ProbSparse self-attention and sequence distillation while effectively capturing long-term temporal dependencies. The experiments are conducted on the ToN-IoT, BoT-IoT, IoT-23, and CICIDS2019 datasets; the EGGP-Informer outperforms other state-of-the-art methods while maintaining lesser memory usage and inference time. Furthermore, this research includes preprocessing for data quality and class imbalance mitigation using class weights, thereby validating its robustness through an ablation study. Therefore, the EGGP-Informer achieves an accuracy of 99.98% for ToN-IoT, 99.99% for BoT-IoT, 99.96% for IoT-23, and 99.89% for CICIDS2019, demonstrating its scalability to diverse networks and uneven distribution of attacks.

Keywords: Entropy-guided gradient pruning; informer; internet of things; intrusion detection system; long-term temporal dependencies; ProbSparse self-attention

I. INTRODUCTION

The internet of things (IoT) presents many possibilities for using a larger number of potential applications in the fields of environmental control, industry, residential premises, and farming [1]. Intrusion detection systems (IDSs) are critical in network security because the existing models determine and address malicious actions [2]. A large proportion of these attacks is executed under the black-box threat model, where adversaries develop perturbations that lead to misclassifications without the access of model parameters, thereby making it highly dangerous [3]. The features of an IDS are usually detected by intruders and suspicious activities while reporting to network administrators [4]. The detection of intrusions in a network is a difficult task, which is important for ensuring data security [5]. Cyber-attacks are continuously divided into two categories: active and passive [6]. Passive attacks are more difficult and elusive than active attacks. Similarly, active attacks are typically easier to detect [7]. Network Intrusion Detection Systems (NIDS) actively watch and evaluate the network traffic to discover malicious movements of digital resources with the intent of securing history [8]. The NIDS aims to enforce the fundamental concepts of information security, confidentiality, integrity, and availability over operational infrastructure [9]. Current malicious actors are becoming highly intelligent and are able to explore, trace networks, and access sensitive information without any permission [10].

An anomaly-based IDS (AIDS) evolves with the changes in the network environment, and detection capability is enhanced constantly [11]. The implementation of machine learning (ML) and artificial intelligence (AI) improves the accuracy and efficiency of IDS [12]. Beyond the abilities of conventional ML techniques, the deep computing provides an advantage in handling complex and diverse data for IoT scenarios [13]. This deep learning (DL) is an advanced version of ML, which is inspired by the functioning of neural networks in the human brain that achieves a remarkable result of malicious activity [14]. However, in contrast to conventional approaches, ML-based IDS can train large volumes of data by facilitating generalized models for IDS detection [15]. This ability enables it to identify and classify not only advanced natural attacks but also artificially generated attacks by recognizing patterns and adjusting to new threats accordingly [16]. Convolutional neural networks (CNN) [17] and recurrent neural networks (RNN) are better than DL models. The major methods include the detection of outliers, autoencoders, generative adversarial networks (GAN), and transfer learning techniques [18]. Nevertheless, the existing model has a significant limitation, such as the absence of interpretability and transparency of DL models, which makes the user less confident and confused regarding how the system decides [19]. In addition, the DL models are more accurate but require significant computational resources to train and deploy the datasets [20]. Within the realm of the metaverse, IoT networks increase the expansion of the metaverse, which creates an abundance of data production and computational expenses; this generates more problems with the virtual infrastructure scaling and efficiency [21]. The novelty of this research lies in introducing an entropy-guided gradient pruning (EGGP) that integrates prediction uncertainty and gradient sensitivity to prioritize traffic flows. Compared to existing

Corresponding author: Yamuna Raju (e-mail: yamunaraju2003@gmail.com/ yamunaraju2003@rediffmail.com).

IDS approaches, which perform heavy on feature engineering or apply attention to each flow extensively, the EGGP-Informer prunes redundant flows before the attention layer, thereby making it better for IDS to combine entropy-gradient scoring with the Informer backbone. This significantly minimizes memory usage and computational complexity while improving detection for subtle and complex attacks among heterogeneous IoT environments.

A. RESEARCH GAP

Despite the significant progress in DL-based IDS models, the existing approaches exhibit several contradictions, which limits the practical deployment in large-scale IoT environments. First, the existing models introduce attention mechanisms to the network by increasing architectural depth; this approach increases computational and memory overhead. As a result, these models become unsuitable for real-time deployment. Second, the feature selection method attempts to reduce complexity but often relies on multistage pipelines. This introduces additional latency and reduces adaptability to evolving attacking patterns. Furthermore, most of the attention-based models treat all traffic flows uniformly; this fails to prioritize informative flows, which is critical for detecting subtle and low-frequency attacks under severe class imbalance. These contradictions highlight the need for an IDS, which simultaneously achieves accuracy, computational efficiency, and adaptive flow prioritization.

B. PROBLEM STATEMENT

Existing DL-based IDS models, such as Informer, process each network flow equally without filtering irrelevant data. This approach results in higher memory usage, computational overhead, and degraded performance in IoT. Furthermore, the inability to focus on uncertain flow reduces the detection accuracy of subtle attack patterns. The selective mechanism does not prioritize the data flow before passing it into the attention mechanism.

C. OBJECTIVE

The objective is to develop an optimized IDS framework using EGGP with an informer. The proposed model aims to selectively retain higher impact and undefined network flows before attention layers by reducing noise to focus on important traffic. This enhances classification accuracy and generalization across various datasets for IDS in IoT networks.

D. CONTRIBUTIONS

The main contributions of this research are listed as follows:

- This research introduces a novel EGGP method that prioritizes significant network traffic that flows by considering predicting confidence and gradients. By pruning irrelevant flows before attention computation, this model reduces noise and improves the focus on anomalous patterns, thereby effectively learning in IDS.
- The Informer consists of a backbone network, which is enhanced with ProbSparse self-attention and sequence distillation to capture long-range dependencies in traffic sequences. This enables the model to manage large-scale data efficiently and reduces memory utilization.
- The proposed EGGP-Informer is validated using four IDS datasets: ToN-IoT, BoT-IoT, IoT23, and CICIDS2019. It outperforms other state-of-the-art methods and delivers better generalization across various attack types.
- Furthermore, this research integrates preprocessing such as duplicate removal, label encoding, and min-max normalization to ensure data quality and uniform feature scaling. It also addresses class imbalance using class weight assignment during training, which enhances the model's ability to detect attacks.

This research is organized as follows: Section II explains the literature review, and Section III elaborates on the proposed methodology. Section IV provides the result analysis, and Section V concludes the research with future work.

II. LITERATURE REVIEW

Raju *et al.* [22] presented a Taylor Optimal Strategy with Starting Murmuration Optimizer (TOSSMO) and Kolmogorov–Arnold Networks (KAN) for network security. The bio-inspired optimization of TOSSMO was used for feature selection. The adaptive learning of KAN for robust classification enhanced the model's performance. TOSSMO effectively recognized appropriate features while balancing exploration and exploitation. The KAN classifier enabled different attacks in an IDS by mapping nonlinear features, which addressed higher dimensionality issues.

Silivery *et al.* [23] developed a dual-path feature extraction based on hybrid IDS for IoT. This dual-path feature extraction was used to achieve feature extraction and fusion processes. Following that, the fused features were introduced into the Neural Architecture Search Network (NASN) for attack detection and classification. Conditional Tabular Generative Adversarial Network (CTGAN) was developed to attain a balanced dataset compared to the conventional augmentation. NasNet-based DL was applied to classify network traffic to enhance the classification accuracy.

Shebl *et al.* [24] introduced a deep CNN (DCNN) for IDS binary and multi-class classification. The DCNN achieved higher flexibility to detect intrusions and malicious activities for binary as well as multi-class classification. This enhanced the applicability among various IDS. Here, the CNN was better for recognizing significant patterns through rows and columns as a spatial feature, while the deep neural network (DNN) captured complex relationships among features, particularly in higher-dimensional data.

Sadhwani *et al.* [25] developed CNN and long short-term memory (LSTM) and bidirectional LSTM (Bi-LSTM) for IDS classification. The optimal model was selected for each dataset based on the performance and training time. The SHapely Additive exPlanations (SHAP) was utilized to provide feature explanations. The features were impacted by the model decision, which was extracted by the explanation produced from SHAP. Data augmentation was performed in all healthy and malignant cases.

Soltani *et al.* [26] suggested a robust IDS for network communication in IoT that used a hybrid ML approach. The hybrid ML approach used K-nearest neighbors (KNN) and random forest (RF), which were classified to improve the accuracy of IDS to reduce attack risk. Additionally, it utilized backward elimination and linear discriminant analysis for feature reduction; this reduced computation costs.

Gheni and Al-Yaseen [27] explained a two-step data cluster for an enhanced IDS. Cluster addressed the challenge based on IDS in terms of higher dimensionality; therefore, it reduced the size to

enhance efficiency. It was developed using a combination of an optimization algorithm on a static tool. The effectiveness of the developed model was estimated using different evaluation metrics.

Saikam and Ch [28] introduced an ensemble approach based on IDS, which used an Improved Spotted Hyena Optimization (ISHO) and Honey Badger Algorithm (HBA) to address class imbalance and overfitting issues. The dataset was balanced through enhanced data sample and detection accuracy. The Squeeze-and-Excitation (SE)-Deep Residual Network 152 (SE-ResNet152) was applied to eliminate the less-significant features. Each iterative stage contained a decision tree (DT), which handled a classifier performance to prevent overfitting issues.

From the overall analyses, recent IDS approaches, such as TOSSMO-KAN, dual-path NASNet, and hybrid ML models, introduced optimizers but relied on complex feature selection, data generation, and multi-stage processing; this led to higher computational complexity.

III. PROPOSED METHODOLOGY

This research proposes a novel framework for intrusion detection in IoT, namely the EGGP-Informer model. This model uses four IDS datasets: ToN-IoT, BoT-IoT, IoT-23, and CICIDS2019, which contain various attack types and network scenarios. Preprocessing includes duplicate removal, label encoding, and min-max normalization to ensure clean and normalized input for the model. The class weights are dynamically assigned to address the class imbalance during training to enhance the model's sensitivity to minority attack classes. The EGGP-Informer integrates pruning to filter out lower-impact flows based on entropy and gradient scores before the attention mechanism. The former utilizes ProbSparse self-attention and sequence distillation for effective long-sequence modeling to enhance detection accuracy. The Efficient Cartesian Genetic Programming (ECGP)-Informer lies in its early-stage pruning mechanism. Instead of applying self-attention to all traffic flows, the model provides a novel entropy-gradient scoring system to filter flows before sequence encoding. This is a novel enhancement over existing Informer-based IDS models; this does not integrate selective pruning. By combining pruning within Informer, the model achieves a rare balance of computational efficiency, robustness, and higher detection accuracy. Fig. 1 provides the complete process of the proposed EGGP transformer.

A. DATASET

To evaluate the effectiveness of the IDS approach, this research uses four publicly available and widely recognized IoT security datasets, such as ToN-IoT [29], BoT-IoT [30], IoT-23 [31], and CICIDS2019 [32]. These datasets contain various types of malicious traffic across different IoT scenarios, attack types, and network protocols.

1). ToN-IoT DATASET. The ToN-IoT is used by the Australian Center of Cyber Security, which contains over 22 million network flows that can represent both the normal and abnormal traffic in an IoT environment. Moreover, 796,380 benign flows and 21,542,641 malicious flows are divided into 44 various types of attacks. As such, these threats comprise denial-of-service (DoS), man-in-the-middle (MITM), reconnaissance, scanning, and injection attacks.

2). BoT-IoT DATASET. Cyber Range Lab and the University of New South Wales (UNSW) Canberra use a realistic network environment to develop the BoT-IoT dataset. It incorporates general and botnet traffic to capture various types of attacks, including distributed DoS (DDoS), information theft, surveillance, and DoS. The data contains over 73 million records with 46 features and five classes.

3). IoT-23 DATASET. The IoT-23 data consists of 23 labeled network traffic that was collected in 2018–2019 with three benign and 20 malicious scenarios. Malicious botnet attacks on IoT devices utilize malicious traffic, and benign traffic is gathered on real devices, such as Amazon Echo, Somfy smart locks, and Philips Hue lights. The log files and packet capture 21 features and approximately 325 million records.

4). CICIDS2019 DATASET. The CICIDS2019 dataset mimics realistic DDoS attacks, which exploit numerous Transmission Control Protocol and User Datagram Protocol (TCP/UDP)-based protocols and consist of both exploitation- and reflection-based threats. The attacks are recorded during a time span of two days to separate training and testing stages; the dataset provides more than 80 features that are based on flows. It also consists of a broad range of attacks, such as SNMP, LDAP, UDP-Lag, MSSQL, SYN, NetBIOS, NTP, DNS, and WebDDoS.

Collectively, all these datasets are partitioned as a chronology-aware strategy, wherever timestamp information is available, that ensures fair and reproducible evaluation of the proposed EGGP-Informer. The IoT-23 and CICIDS2019, the datasets consist of explicit temporal ordering for network flows; the data are sorted chronologically and split into 80% training and 20% testing based on time. This prevents information leakage from future traffic into a training phase, which reflects a realistic IDS deployment scenario. Furthermore, a chronological separation of the dataset is not feasible due to the use of aggregated flows. The representation includes BoT-IoT and ToN-IoT; a stratified random split is adopted while preserving the ordinal class distribution. The random selection is used in these cases to ensure sufficient representation of minority attack classes in both the training and testing sets for stable learning in the highly imbalanced IDS dataset.

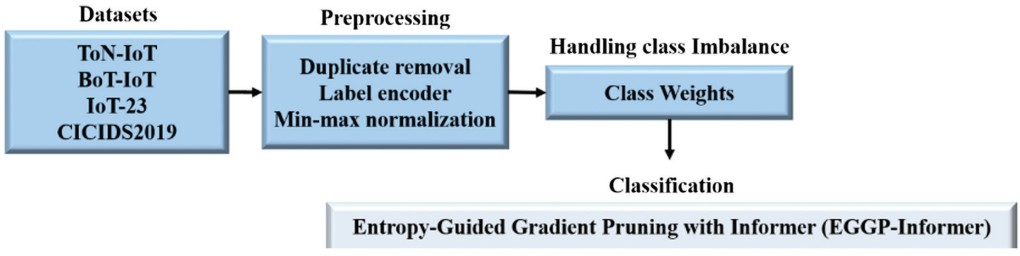

Fig. 1. Overall workflow of the proposed methodology dataset.

B. PREPROCESSING

Some preprocessing processes are conducted to ensure data quality, uniformity, and aptness in training models on the availability of DL-based intrusion detection datasets. Redundancy is reduced by spotting duplicate records to eliminate the possibility of bias in knowledge learning. For example, the ToN-IoT dataset contains 11,071 duplicates, which eliminates a part of the deduplication process that is similar to the cases of the BoT-IoT, CICIoT2023, and CICIDS2019 datasets.

Categorical features, such as protocol types, services, and attack labels, are converted by label encoding, where each category is converted into a unique integer. The necessity to convert this lies in DL models requiring numerical information to process it successfully.

To normalize the input features, all numerical attributes are scaled by min-max normalization within the range of [0, 1]. This ensures that all the features contribute equally and prevents higher-scale features from dominating the learning process, stabilizes training, and accelerates convergence. Moreover, it also increases the comprehensiveness of the model in differentiated data distributions.

1). HANDLING CLASS IMBALANCE USING CLASS WEIGHTS. This research uses the class imbalance in the ToN-IoT, BoT-IoT, CICIoT2023, and CICIDS2019 datasets by weighting the classes in training the DL-based models. The sample size difference between benign and attack classes being large, this strategy ensures that the model is attentive to underrepresented types of attacks. The class weight w_c for the class c is estimated using Equation (1):

$$w_c = \frac{N}{n_c \cdot C} \tag{1}$$

where n_c denotes the number of samples in class c, N and C represent the total number of samples and classes, respectively. This formula enhances minority class weights, which enables the model to learn better using these classes. The loss function is trained using weights in the class, and the original data are not changed. This method is also more favorable than oversampling methods, such as Synthetic Minority Over-sampling Technique (SMOTE), on larger datasets, namely, BoT-IoT and CICIDS2019, to prevent the stability of the training process and computational complexity. All these are done automatically by using built-in support of class weights in DL libraries so that the balanced learning is ensured across classes that follow preprocessing data, including duplicate removal, encoding, and normalization using min-max algorithms.

C. CLASSIFICATION

The informer is a powerful prediction algorithm based on a transformer architecture that utilizes an encoder–decoder structure. It transforms the input sequences into an encoder that mitigates temporal complexity using a ProbSparse self-attention mechanism. This is followed by self-attention distillation, which compresses the time to aspect the input sequence. The predictions are subsequently produced by the decoder. The architecture of the informer is highly scalable and efficient, which makes it suitable for large-scale and complex time-series information, making it a better base model. Fig. 2 illustrates the architecture of the proposed EGGP-Informer.

1). ENTROPY-GUIDED GRADIENT PRUNING. To enhance computational complexity and detection accuracy, this research proposes the ECGP, which dynamically selects the significant flow embeddings prior to input into the former encoder. This ensures that only network activities are highlighted, whereas the irrelevant flows are pruned early. In this case, each input flow instance x_i is the first embedding using linear transformation and positional encoding. During training, the forward pass estimates the class

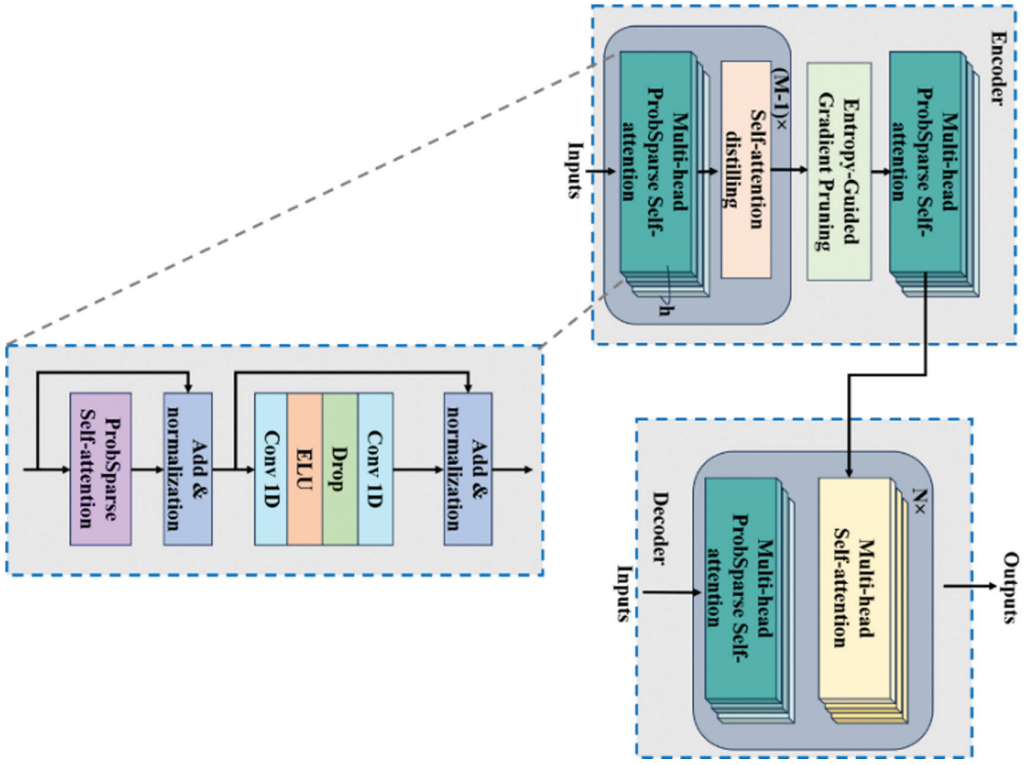

Fig. 2. Architecture of proposed entropy-guided gradient pruning with informer.

probabilities P_i for each flow, and the following entropy is represented in Equation (2):

$$H(x_i) = -\sum_j p_{ij}\, log(p_{ij}) \tag{2}$$

where j is the class index. A higher entropy value indicates that the model is uncertain about the classification flow, whereas a lower entropy value provides a confident prediction. Similarly, this research estimates the gradient norm, which is depicted in Equation (3):

$$G(x_i) = \left\| \frac{\partial L}{\partial x_i} \right\| \tag{3}$$

This represents the influence of each flow on the loss model L. The final significance score for each flow is estimated as a weighted combination of gradient sensitivity and confidence, which is demonstrated in Equation (4):

$$\begin{aligned} Score(x_i) &= \lambda \cdot G(x_i) \\ &+ (1-\lambda)\cdot(1 - Normalized\, Entropy(x_i)) \end{aligned} \tag{4}$$

where $\lambda \in [0,1]$ the gradient magnitude impacts and predicts certainty. Flows with fewer combined scores are pruned before attention computation in the encoder. This pruning is better for an IDS, where many benign flows share similar patterns that contribute to learning. By removing it earlier, the informer model allocates more attention to enhance both efficiency and detection, specifically for attack types.

2). ProbSparse SELF-ATTENTION. Traditional self-attention requires $O(L_Q L_K)$ memory and quadratic dot product computation, which significantly limits its prediction ability. The long-tail distribution in the self-attention feature map has fewer dot product pairs, which contribute to the main attention, whereas others are ignored. To ensure the significance for each key under a given query, an attention probability distribution $p(k_j|q_i)$ and a uniform distribution $q(k_j|q_i)$ are introduced in Equations (5) and (6):

$$p(k_j|q_i) = \frac{k(q_i,k_j)}{\sum_l k(q_i,k_l)} \tag{5}$$

$$q(k_j|q_i) = \frac{1}{L_K} \tag{6}$$

where q_i and k_j are the query and key vectors, respectively, L_K is the total number of keys, and $k(q_i,k_j)$ is an asymmetric exponential kernel function $\exp(\frac{q_i k_j^T}{\sqrt{d}})$. Moreover, the direct correlation between two distributions is evaluated by the discrete Kullback–Leibler divergence formula as provided in Equation (7):

$$D_{KL}(P||Q) = \sum_{i\in X} P(i) * \left[log\left(\frac{P(i)}{Q(i)}\right)\right] \tag{7}$$

where $Q(i)$ and $P(i)$ represent probabilities of probability distributions Q and P at event i, respectively. This formula is applied to measure the expected information loss when the probability distribution approximates another distribution. Each query requires evaluating its sparsity, which is assessed by estimating Kullback-Leibler divergence between the two distributions, as mentioned above, and its substitutes are presented in Equations (5) to (8):

$$\begin{aligned} D_{KL}(P||Q) &= \sum_{j=1}^{L_K}\left(\frac{1}{L_K}\cdot In\frac{1}{L_K} - \frac{1}{L_K}\cdot In\frac{k(q_i,k_j)}{\sum_l k(q_i,k_l)}\right) \\ &= -InL_K - \sum_{j=1}^{L_K}\frac{1}{L_K}\cdot\left(In\, e^{\frac{q_i k_j^T}{\sqrt{d}}} - In\sum_l e^{\frac{q_i k_j^T}{\sqrt{d}}}\right) \\ &= -InL_K - \frac{1}{L_K}\sum_{j=1}^{L_K}\frac{q_i k_j^T}{\sqrt{d}} + In\sum_{l=1}^{L_K} e^{\frac{q_i k_j^T}{\sqrt{d}}} \end{aligned} \tag{8}$$

After removing the constant term, a sparsity measure of the ith query vector is presented in Equation (9):

$$\begin{aligned} M(q_i,K) &= In\sum_{j=1}^{L_K} e^{\frac{q_i k_j^T}{\sqrt{d}}} - \frac{1}{L_K}\sum_{j=1}^{L_K}\frac{q_i k_j^T}{\sqrt{d}} \\ &\le In\left(L_K \cdot max_j\left\{\frac{q_i k_j^T}{\sqrt{d}}\right\}\right) - \frac{1}{L_K}\sum_{j=1}^{L_K}\left(\frac{q_i k_j^T}{\sqrt{d}}\right) \\ &= InL_K + max_j\left\{\frac{q_i k_j^T}{\sqrt{d}}\right\} - \frac{1}{L_K}\sum_{j=1}^{L_K}\left(\frac{q_i k_j^T}{\sqrt{d}}\right) \end{aligned} \tag{9}$$

Based on the sparsity measure results, the random selection of $L_Q InL_K$ queries participate in the dot product calculation of the attention mechanism. This minimizes the complexity of $O(L^2)$ to $O(LInL)$. The selected queries are taken as being relatively far from the uniform distribution.

3). ENCODERS. Due to the presence of numerous irrelevant vectors in sequences, the processed ProbSparse self-attention mechanism is necessary to utilize the self-attention distilling operation to selectively improve the significant parts of the input data. This overcomes the limitations of the transformer, such as memory usage bottlenecks and stacked layers, while minimizing the network parameters without dropping significant data. The series of operations based on 1D convolution and max pooling is called a distilling operation. The process from layer j to $j+1$ layer is summarized in Equation (10):

$$X_{j+1}^t = MaxPool\left(ELU\left(Conv1D\left(\left[X_j^t\right]\right)\right)\right) \tag{10}$$

where $[\cdot]$ is a ProbSparse self-attention block $Conv1D(\cdot)$, which uses $ELU(\cdot)$ as an activation function for the executing convolution operations in the time dimension. After each convolution layer, the max pooling layer includes downsampling to half of its length, thereby reducing the memory utilization to $O((2-\lambda)LlogL)$.

4). DECODERS. Compared to the conventional transformers, which predict step-by-step outputs, the generative decoder of Informer predicts longer sequence outputs in a single forward propagation; this enhances the prediction speed and reduces the cumulative errors. The input of the decoder at time t is the concatenation of the two parts, as presented in Equation (11):

$$X_{feed_{de}}^t = Concat(X_{\text{token}}^t, X_0^t) \in \mathbb{R}^{(L_{token}+L_y)\times d_{model}} \tag{11}$$

where $X_{feed_{de}}^t$ is an input to the decoder, X_{token}^t is a start token of the sequence, X_0^t is a placeholder for the target sequence, and zero padding is used to maintain the consistency of an input dimension. Subsequently, masked multi-head self-attention is utilized to focus each location on the data proceeding, thereby conserving autoregressive features and preventing future data leakage, which improves generalization abilities.

Algorithm for the proposed EGGP-Informer:

Input: IDS dataset D, pruning threshold θ, balancing factor λ

Output: Predicted class labels

Preprocessing:

Remove duplicate records from D
Apply label encoding to categorical features.
Apply min–max normalization to numerical features.
Compute class weights w_c for every class c using Equation (1).

Model Initialization:

Initialize the Informer backbone with ProbSparse self-attention using Equations (5)–(9) and sequence distillation using Equation (10).
Set the pruning threshold θ and balancing factor λ.
Initialize optimizer and weighted cross-entropy loss using w_c from Equation (1).

Training:

For each epoch = 1 to E do
For each mini-batch B_D, do
Embedding for each flow $x \in B$, compute embedding z through linear projection and positional encoding.
Prediction probabilities $p = Softmax(f(z))$
Compute entropy $H(x)$, gradient norm $G(x)$, and significance score $S(x)$ using Equations (2), (3), and (4), respectively.
Keep flows where $S(x) \geq \theta$, discard others.
Apply ProbSparse self-attention using Equations (5)–(9).
Apply the distilling operation using Equation (10).
Prepare input using Equation (11) and generate outputs.
Compute weighted cross-entropy loss from Equation (1).
Backpropagate and update parameters.

IV. RESULT ANALYSIS

The EGGP-Informer is simulated in Python 3.8 with system configurations of 16 GB RAM, an Intel i7 processor, and Windows 10 OS. Metrics, such as precision, recall, accuracy, and F1-score, are considered to estimate EGGP-Informer performance. The mathematical formulas for these metrics are illustrated in Equations (12)–(15):

$$Precision = \frac{TP}{TP + FP} \times 100 \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \times 100 \tag{13}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \tag{14}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100 \tag{15}$$

where TP,TN,FP, and FN are true positive, true negative, false positive, and false negative, respectively.

Table I compares the performance of EGGP-Informer with the existing models, such as vision transformer (ViT), temporal fusion transformer (TFT), and conventional Informer, using benchmark IDS datasets, including ToN-IoT, BoT-IoT, IoT-23, and CICIDS2019. The EGGP-Informer on the ToN-IoT dataset reaches 99.98% of accuracy, 99.98% of precision, 99.98% of recall, and an F1-score of 99.98%, which outperforms ViT, TFT, and Informer. On the BoT-IoT dataset, EGGP-Informer performs better in all performance measures of (99.99%), which has some excellent capabilities in the classification of data. The EGGP-Informer has an accuracy of 99.96%, an F1-score of 99.93%, a precision of 99.93%, and a recall of 99.94% on the IoT-23 dataset. Finally, CICIDS2019 continues to achieve better performance with 99.85% precision, 99.89% accuracy, a 99.89% F1-score, and 99.85% recall. The higher performance is attributed to the abilities of the longer sequence modeling in the former together with the EGGP to filter out irrelevant traffic flows, thereby enabling the model to reorganize its attention to relevant high-value patterns. This architecture is more effective for detecting scenarios, particularly in complicated and imbalanced network traffic environments. Moreover, the baseline models are intentionally chosen as architecture-aligned transformer models rather than task-specific IDS models. This selection enables and controls fair comparison under experimental conditions, including the same preprocessing pipeline, data splits, training epochs, and evaluation metrics used to isolate performance gains.

Table II provides the EGGP-Informer performance after applying K-fold validation for four datasets of the IDS: ToN-IoT, BoT-IoT, IoT-23, and CICIDS2019. The K values of $K = 3$, $K = 5$, $K = 7$, and $K = 8$ are considered. Furthermore, to mitigate potential bias, which is introduced by a single data split, and validate the robustness of reported results, multiple K values are evaluated. Compared to all the configurations, $K = 5$ achieves the highest scores on all dataset and for all metrics. This means that $K = 5$ provides a better trade-off between the generalization and learning abilities of the model. Smaller values, such as $K = 3$, lead to a lower generalization due to the availability of lower validation data; in contrast, with larger values, such as $K = 7$ or $K = 8$, the performance will be lower due to the lower training data. Therefore, $K = 5$ proposes good learning without overfitting; the results provide stable and generalized performance in various situations of network traffic.

Table I. Performance of different transformers with EGGP-Informer on IDS dataset

Dataset	Method	Precision (%)	Recall (%)	Accuracy (%)	F1-score (%)
ToN-IoT	ViT	99.53	99.42	99.46	99.47
	TFT	99.36	99.27	99.23	99.31
	Informer	99.11	99.06	99.04	99.09
	EGGP-Informer	99.98	99.98	99.98	99.98
BoT-IoT	ViT	99.62	99.51	99.53	99.56
	TFT	99.45	99.32	99.33	99.38
	Informer	99.21	99.13	99.12	99.17
	EGGP-Informer	99.99	99.99	99.99	99.99
IoT-23	ViT	99.44	99.31	99.34	99.37
	TFT	99.26	99.15	99.13	99.20
	Informer	99.02	98.91	98.92	98.96
	EGGP-Informer	99.93	99.94	99.96	99.93
CICIDS2019	ViT	99.24	99.03	99.12	99.13
	TFT	99.05	98.84	98.91	98.94
	Informer	98.83	98.62	98.71	98.72
	EGGP-Informer	99.93	99.85	99.89	99.89

Table II. Performance of K-fold validation as an ablation study of EGGP-Informer on the IDS dataset

Dataset	Method	Precision (%)	Recall (%)	Accuracy (%)	F1-score (%)
ToN-IoT	$K=3$	99.74	99.70	99.72	99.72
	$K=5$	99.98	99.98	99.98	99.98
	$K=7$	99.88	99.85	99.86	99.86
	$K=8$	99.83	99.79	99.80	99.81
BoT-IoT	$K=3$	99.80	99.76	99.78	99.78
	$K=5$	99.99	99.99	99.99	99.99
	$K=7$	99.90	99.86	99.87	99.88
	$K=8$	99.84	99.80	99.81	99.82
IoT-23	$K=3$	99.70	99.60	99.65	99.65
	$K=5$	99.93	99.94	99.96	99.93
	$K=7$	99.85	99.82	99.84	99.83
	$K=8$	99.78	99.75	99.76	99.76
CICIDS2019	$K=3$	99.60	99.40	99.52	99.50
	$K=5$	99.93	99.85	99.89	99.89
	$K=7$	99.82	99.72	99.75	99.77
	$K=8$	99.76	99.68	99.71	99.72

Table III lists the standard deviation of the proposed EGGP-Informer on ToN-IoT, BoT-IoT, IoT-23, and CICIDS2019. On the ToN-IoT dataset, EGGP-Informer achieves a precision of $99.98 \pm 0.01\%$, a recall of $99.98 \pm 0.02\%$, an accuracy of $99.98 \pm 0.01\%$, and an F1-score of $99.98 \pm 0.02\%$. The lower standard deviation of the overall dataset presents better stability and reliability in the recurrence runs and data distributions of the EGGP-Informer.

Table IV displays the corresponding results of the statistical and complexity hydroxyzine price, which is analyzed in the proposed EGGP-Informer and IDS dataset. The results are

Table III. Performance of the standard deviation of EGGP-Informer on the IDS dataset

Dataset	Precision (%)	Recall (%)	Accuracy (%)	F1-score (%)
ToN-IoT	99.98 ± 0.01	99.98 ± 0.02	99.98 ± 0.01	99.98 ± 0.02
BoT-IoT	99.99 ± 0.01	99.99 ± 0.01	99.99 ± 0.01	99.99 ± 0.01
IoT-23	99.93 ± 0.03	99.94 ± 0.03	99.96 ± 0.02	99.93 ± 0.03
CICIDS2019	99.93 ± 0.03	99.85 ± 0.04	99.89 ± 0.03	99.89 ± 0.03

Table IV. Performance of statistical and complexity analysis of EGGP-Informer on IDS datasets

Dataset	*P*-value (*t*-test)	Training time (s)	Memory usage (MB)	Inference time (s)
ToN-IoT	0.005	168.4	712	102.6
BoT-IoT	0.004	152.1	685	94.8
IoT-23	0.006	161.7	730	109.3
CICIDS2019	0.003	145.9	690	97.5

statistically significant in all the *p*-values that are received during the *t*-test, which are less than 0.05. The model train takes different amounts of time and is 168.4 s on ToN-IoT and 145.9 s on CICIDS2019. The computational cost is lesser, such as 685 MB and 730 MB of memory usage and 94.8 s and 109.3 s inference time on the BoT-IoT and IoT-23 datasets, which indicates the EGGP-Informer is computationally efficient. Therefore, to provide a comprehensive and hardware-independent assessment of the model complexity, the computational efficiency not only evaluates training and inference time but also use multiple complementary metrics, including memory usage, attention computation reduction, and architectural sparsity. The proposed model reduces computational overhead by pruning low-impact traffic that flows prior to the attention mechanism. This pruning mechanism decreases the effective sequence to lengthen the process encoder. This directly lowers the quadratic attention complexity and reduces intermediate activation storage. Furthermore, the ProbSparse self-attention and sequence distillation mechanisms of the proposed model reduce memory footprint and improve scalability. The training and inference times are reported for reproducibility. These findings justify the fact that the model is accurate, but also large-scale and real-time IDS applications are deployed based on the model. Moreover, the proposed model demonstrates computational efficiency, but it is important to distinguish this from its strict real-time deployment guarantees. The experimental evaluation is conducted in a controlled software environment, and it is intended to assess algorithmic efficiency rather than to validate hard real-time and deadline compliance. Although the reduced attention complexity and the early pruning mechanisms indicate the proposed model, it adapts to real-time deployment intrusion detection scenarios.

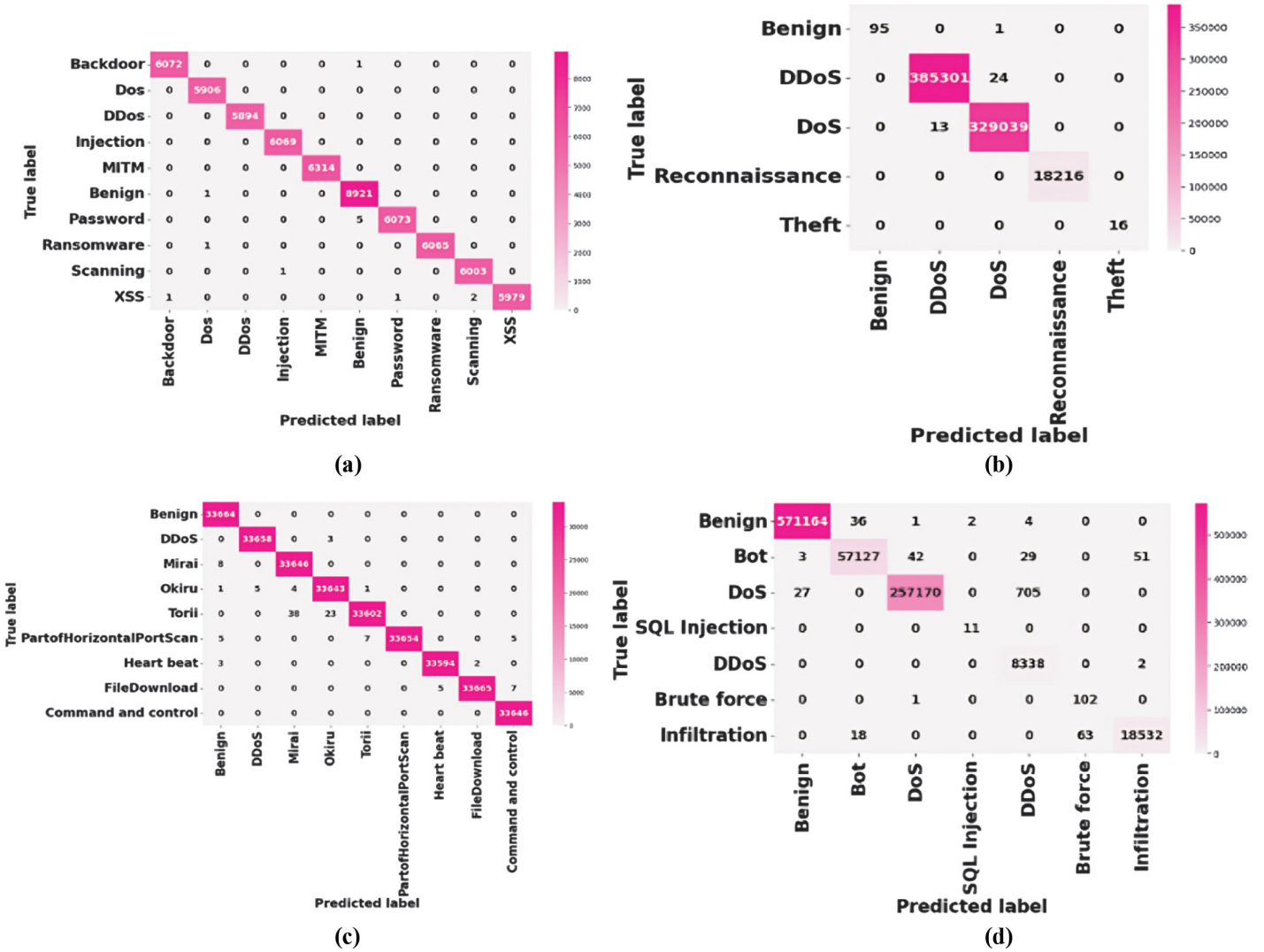

Fig. 3. Confusion matrix for (a) ToN-IoT, (b) BoT-IoT, (c) IoT-23, and (d) CICIDS2019 datasets.

Table IV proposes the EGGP-Informer model introducing additional computations for entropy estimations and gradient norm calculation per flow. This overhead is linear with respect to the number of flows, which is defined as $O(N)$. On the other hand, the self-attention mechanism in transformer-based models has quadratic complexity that represents $O(N^2)$ with respect to sequence length. By eliminating the low-impact flows earlier for attention computation, the EGGP model significantly reduces the effective sequence length by the Informer encoder, which results in a substantial net reduction in overall computational cost. Compared to the simpler pruning or feature selection methods, which rely on static heuristics or offline preprocessing mechanisms. The proposed model operates dynamically while training, which allows flow importance to be adaptively determined based on both prediction uncertainty and gradient sensitivity. This introduces additional per-flow computation, and the cost is returned by the reduced attention operations and lower memory usage in subsequent layers. Consequently, for large-scale IoT traffic with millions of flows, the EGGP provides a favorable trade-off between pruning overhead and attention complexity reduction, which leads to improved scalability.

Figure 3 presents the confusion matrices of the ToN-IoT, BoT-IoT, IoT-23, and CICIDS2019 datasets. In each dataset, the EGGP-Informer model has a better classification accuracy with a clear diagonal and fewer misclassifications. Even the finest-grained attack types, such as MITM, Okiru, and Torii, are accurately discovered in the ToN-IoT and IoT-23. The BoT-IoT and CICIDS2019 matrices provide almost perfect detection of DoS- and DDoS-type attacks on a larger scale, and the separation of the benign traffic is also better. The lower weights on the off-diagonal represent higher precision and recall in both the common and infrequent categories of attacks.

Figure 4 provides the ROC curves of the ToN-IoT, BoT-IoT, IoT-23, and CICIDS2019 and the discriminative ability of the EGGP-Informer model. All classes within the dataset have very high area under the curve (AUC) values; this provides a higher true positive rate and a lower false positive rate. Even complex or rare attacks, such as infiltration, SQL injection, and botnet variants, are perfectly separable according to the model. This union has various datasets that ensure the model is robust, generalizable, and efficacious in performing intrusion detection tasks.

A. COMPARATIVE ANALYSIS

A comparative analysis of the proposed EGGP-Informer against the state-of-the-art intrusion detection methods is reported in Table V for all four datasets of IDS: TON-IoT, BoT-IoT, IoT-23, and CICIDS2019. The comparison is against complex models, such as TOSSOM-KAN [22], dual-path feature extraction [23], CNN-X [24], hybrid ML [25], and deep architectures including DCNN [26], MLP-AE [27], as well as ISHO-HBA and SE-ResNet152 [28]. In all datasets, the proposed EGGP-Informer performs better in the terms of accuracy. Specifically, it achieves 99.98% for TON-IoT, 99.99% for BoT-IoT, 99.96% for IoT-23, and 99.89% for CICIDS2019. The comparison models publish sources that evaluate under their respective original experimental settings, datasets, and hyperparameters. These results are presented in indirect comparison with different experimental conditions for the same dataset to contextualize the effectiveness of EGGP-Informer rather than strict head-to-head evaluations. These performance figures are far better than those of current practices that prove the strength and generalization factor of EGGP-Informer in various types of attacks. The improved performance can be explained by adopting EGGP into the former architecture, which

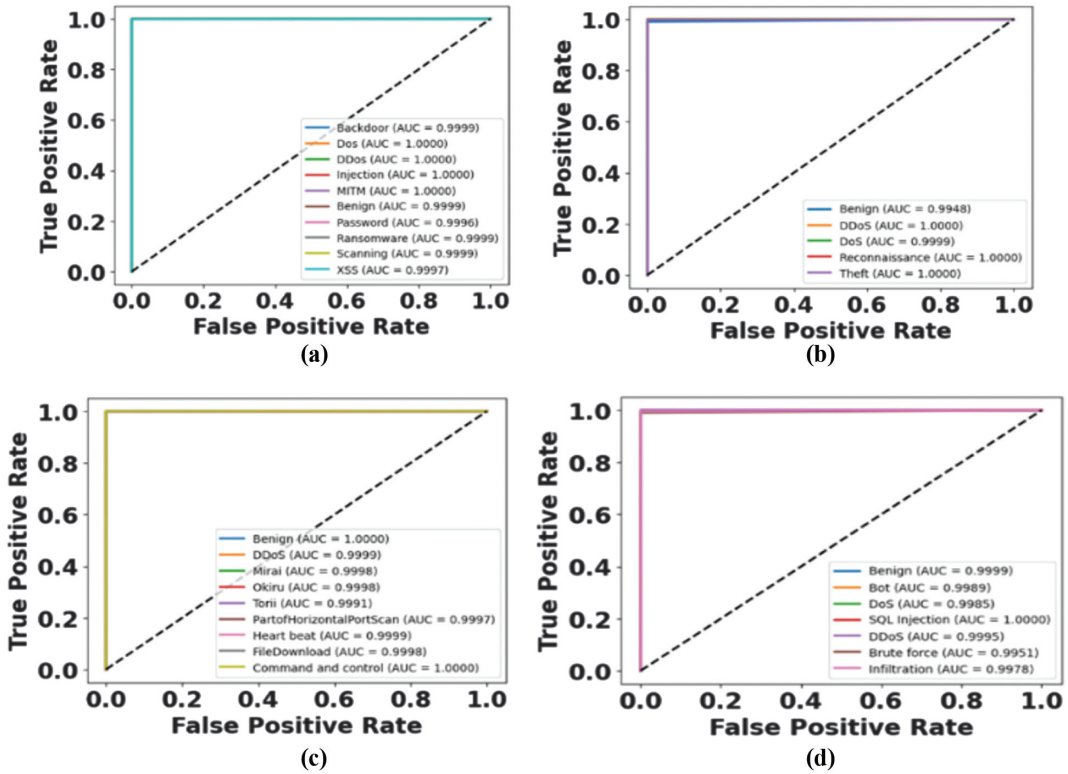

Fig. 4. ROC curve for (a) ToN-IoT, (b) BoT-IoT, (c) IoT-23, and (d) CICIDS2019 datasets.

Table V. Comparative analysis of proposed EGGP-Informer on IDS datasets

Dataset	Method	Precision (%)	Recall (%)	Accuracy (%)	F1-score (%)
ToN-IoT	TOSSOM-KAN [22]	99.97	99.97	99.96	99.96
	Dual-path feature extraction [23]	99.58	99.54	99.63	99.56
	CNN-X [25]	97.91	97.67	97.80	97.72
	Hybrid ML [26]	99.96	99.96	99.96	99.96
	EGGP-Informer	99.98	99.98	99.98	99.98
BoT-IoT	TOSSOM-KAN [22]	99.98	99.98	99.99	99.98
	Dual-path feature extraction [23]	99.47	99.43	99.51	99.45
	EGGP-Informer	99.99	99.99	99.99	99.99
IoT-23	Dual-path feature extraction [23]	99.61	99.57	99.67	99.59
	DCNN [24]	99.91	99.68	99.50	99.75
	MLP and AE [27]	99	99	99.26	99
	EGGP-Informer	99.93	99.94	99.96	99.93
CICIDS2019	ISHO-HBA and SE-ResNet152 [28]	99.42	99.37	99.63	99.15
	EGGP-Informer	99.93	99.85	99.89	99.89

helps to make decisions about the selection of more discriminative features and more effective learning templates of the temporal sequence in the context of intrusion detection. Therefore, it is concluded that the EGGP-Informer is an effective and efficient model for the real-time monitoring of security in networks within heterogeneous IoT environments.

V. DISCUSSION

The performance with better generalization of the proposed EGGP-Informer appears in the experimental outcomes, which are listed in various intrusion detection datasets, such as the ToN-IoT, BoT-IoT, IoT-23, and CICIDS2019. Combining EGGP with an Informer architecture enables the model to provide selective attention to informative and more relevant network flows, thereby eliminating redundant patterns early. This leads to efficiency and better accuracy in identifying the complex or subtle attack behaviors. The EGGP-Informer is more effective than the state-of-the-art methods, such as TOSSOM-KAN, dual-path feature extraction, DCNN, and hybrid ML, in terms of all metrics. It is important to consider that it achieves an accuracy of 99.98% for ToN-IoT, 99.99% for BoT-IoT, 99.96% for IoT-23, and 99.89% for CICIDS2019, demonstrating its scalability to diverse trafficking and uneven distribution of attacks. The higher accuracy and lower standard deviation achieved by the proposed model across multiple datasets raise the concern regarding potential overfitting. To mitigate this risk, the pruning mechanism in the proposed model does not rely on dataset-specific heuristics or handcrafted rules. Rather, it dynamically prioritizes flow based on prediction uncertainty and gradient sensitivity, which are model-intrinsic properties and adapt to changing the data distributions. By highlighting uncertain and high-impact traffic patterns, the proposed model is inherently inclined to focus on anomalous behaviors instead of memorizing static attack patterns. These characteristics recommend improved resilience to evolving previously unseen attack strategies when compared to static feature selection or rule-based pruning methods. Therefore, high-performance values are used for the long-sequence modeling ability of the former, and the role played by the EGGP in redundant flows focuses on the relevant flow in the process. Thus, the EGGP-Informer is scalable and reliable for real-time intrusion detection in heterogeneous and IoT systems.

VI. CONCLUSION

This research proposed a new intrusion detection framework, namely EGGP-Informer, which leveraged EGGP as a part of the Informer framework, thereby achieving higher detection precision and reduced computational complexity. Its accuracy was tested on the ToN-IoT, BoT-IoT, IoT-23, and CICIDS2019 datasets, which represented higher performance in terms of precision, recall, accuracy, and F1-score. EGGP changed irrelevant network flows but retained the most informative ones during encoding. Additionally, ProbSparse self-attention, together with sequence distillation actions, accomplished long-sequence modeling with fewer multiplications. The experimental results demonstrated that the EGGP-Informer outperformed the existing methods and reached 99.99% accuracy for BoT-IoT and 99.96% for IoT-23. Moreover, the effectiveness of the proposed model primarily improved three key indicators, such as detection accuracy, computational efficiency, and model stability. First, the consistent accuracy and F1-score across all four datasets demonstrated the model's strong capability to capture complex and subtle attack patterns even under highly imbalanced traffic conditions. This result confirmed the benefit of the proposed model and computational efficiency, which was reflected in reduced memory usage and inference time. This evaluation highlighted the advantage of pruning redundant flows before attention computation. This made the proposed model suitable for large-scale and real-time IoT environments where the resource constraints were essential. Finally, the lower standard deviation and stable performance across k-folds indicated strong robustness and generalization ability. Collectively, these indicators provided a coherent explanation of the proposed model achieving superior performance compared to the existing methods while improving the logical foundation of conclusions. This ensured the model's ability to generalize across different types of attacks and traffic distributions. In addition, its strength were enhanced by using class weights, normalization of data, and elimination of duplicate data. Nevertheless, the current evaluation was limited to an offline benchmark dataset, where the explicit evaluation under adversarial settings, concept drift, and zero-delay attack scenarios were required to quantify effectiveness. The proposed model was computationally lesser so that it could be deployed in a heterogeneous IoT in real-time applications. Overall, EGGP-Informer

established a major approach toward intelligent and scalable intrusion detection due to its deep modeling of temporal behavior and the use of entropy-based attention to perform security monitoring with higher performance in IDS networks. In the future, this research will focus on integrating explainable AI (XAI) to enhance the interpretability of the model. Although the proposed model was considered computationally effective, it was deployed in a controlled software environment. Therefore, to claim the real-time deployment practically, there was a requirement to deploy the proposed model with explicit deadline constraints and edge-level benchmarking.

REFERENCES

[1] D. R. I. Ali and A. Hamdouchi, "Evaluating the performance of TinyML singular and ensemble techniques for intrusion detection in IoT networks," *Microprocess. Microsyst.*, vol. 117, p. 105172, 2025.

[2] J. Saikam and K. Ch, "EESNN: Hybrid deep learning empowered spatial–temporal features for network intrusion detection system," *IEEE Access*, vol. 12, pp. 15930–15945, 2024.

[3] S. Wali, Y. A. Farrukh, and I. Khan, "Explainable AI and random forest-based reliable intrusion detection system," *Comput. Secur.*, vol. 157, p. 104542, 2025.

[4] A. Hamdouchi and A. Idri, "Enhancing IoT security through boosting and feature reduction techniques for multiclass intrusion detection," *Neural Comput. Appl.*, early access, pp. 1–24, 2025. DOI: 10.1007/s00521-025-11001-2.

[5] R. Devendiran and A. V. Turukmane, "Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy," *Expert Syst. Appl.*, vol. 245, p. 123027, 2024.

[6] K. Mittal and P. Khurana Batra, "Graph-ensemble fusion for enhanced IoT intrusion detection: Leveraging GCN and deep learning," *Cluster Comput.*, vol. 27, pp. 10525–10552, 2024.

[7] M. Sarhan *et al.*, "Feature extraction for machine learning-based intrusion detection in IoT networks," *Digital Commun. Netw.*, vol. 10, no. 1, pp. 205–216, 2024.

[8] R. Gangula, M. M. Vutukuru, and M. R. Kumar, "Stacked auto encoder with weighted loss function for intrusion detection in IoT application," *Multimedia Tools Appl.*, vol. 84, pp. 23501–23529, 2024.

[9] R. A. N. Diaz *et al.*, "Enhancing multi-agent reinforcement learning intrusion detection systems using random forest Q-Value estimation," *Eng. Technol. Appl. Sci. Res.*, vol. 15, no. 4, pp. 24455–24459, 2025.

[10] J. Li *et al.*, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *J. Big Data*, vol. 11, p. 36, 2024.

[11] G. Ayad, N. A. Sakr, and N. A. Hikal, "A hybrid approach for efficient feature selection in anomaly intrusion detection for IoT networks," *J. Supercomput.*, vol. 80, pp. 26942–26984, 2024.

[12] S. Hizal, U. Cavusoglu, and D. Akgun, "A novel deep learning-based intrusion detection system for IoT DDoS security," *Internet Things*, vol. 28, p. 101336, 2024.

[13] M. M. Abualhaj *et al.*, "Enhancing intrusion detection system performance using a hybrid of Harris Hawks and Whale Optimization Algorithms," *Eng. Technol.Appl. Sci. Res.*, vol. 15, no. 4, pp. 24354–24361, 2025.

[14] R. Jablaoui and N. Liouane, "Network security based combined CNN-RNN models for IoT intrusion detection system," *Peer-to-Peer Netw Appl.*, vol. 18, p. 129, 2025.

[15] S. A. Khanday, H. Fatima, and N. Rakesh, "Implementation of intrusion detection model for DDoS attacks in lightweight IoT networks," *Expert Syst. Appl.*, vol. 215, p. 119330, 2023.

[16] M. S. Islam *et al.*, "A novel few-shot ML approach for intrusion detection in IoT," *Arab. J. Sci. Eng.*, vol. 50, pp. 7765–7779, 2024.

[17] R. A. Elsayed *et al.*, "Securing IoT and SDN systems using deep-learning-based automatic intrusion detection," *Ain Shams Eng. J.*, vol. 14, no. 10, p. 102211, 2023.

[18] D. Krishnan and P. Shrinath, "Robust botnet detection approach for known and unknown attacks in IoT networks using stacked multi-classifier and adaptive thresholding," *Arab. J. Sci. Eng.*, vol. 49, pp. 12561–12577, 2024.

[19] M. B. M. Shtayat *et al.*, "An explainable ensemble deep learning approach for intrusion detection in industrial internet of things," *IEEE Access*, vol. 11, pp. 115047–115061, 2023.

[20] T. Gaber *et al.*, "Metaverse-IDS: Deep learning-based intrusion detection system for Metaverse-IoT networks," *Internet Things*, vol. 24, p. 100977, 2023.

[21] V. P. Kumar and L. K. Awasthi, "A resilient intrusion detection system for IoT environment based on a modified stacking ensemble classifier," *SN Comput. Sci.*, vol. 5, p. 1020, 2024.

[22] Y. Raju, P. C. Nagappa, and T. Javarappa, "Taylor optimal strategy with starling murmuration optimizer and Kolmogorov-Arnold networks for detecting security attacks in IoT environments," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 5, p. 876, 2025.

[23] A. K. Silivery, K. R. M. Rao, and R. Solleti, "Dual-path feature extraction based hybrid intrusion detection in IoT networks," *Comput. Electr. Eng.*, vol. 122, p. 109949, 2025.

[24] A. Shebl *et al.*, "DCNN: A novel binary and multi-class network intrusion detection model via deep convolutional neural network," *EURASIP J. Inf. Secur.*, vol. 2024, p. 36, 2024.

[25] S. Sadhwani *et al.*, "IoT-based intrusion detection system using explainable multi-class deep learning approaches," *Comput. Electr. Eng.*, vol. 123, p. 110256, 2025.

[26] N. Soltani *et al.*, "Robust intrusion detection for network communication on the Internet of Things: A hybrid machine learning approach," *Cluster Comput.*, vol. 27, pp. 9975–9991, 2024.

[27] H. Q. Gheni and W. L. Al-Yaseen, "Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset," *e-Prime–Adv. Electr. Eng. Electron. Energy*, vol. 9, p. 100673, 2024.

[28] J. Saikam and K. Ch, "An ensemble approach-based intrusion detection system utilizing ISHO-HBA and SE-ResNet152," *Int. J. Inf. Secur.*, vol. 23, pp. 1037–1054, 2023.

[29] ToN-IoT dataset link: Accessed on: August, 2025, Available: https://www.kaggle.com/datasets/amaniabourida/ton-iot.

[30] BoT-IoT dataset link: Accessed on: August, 2025, Available: https://www.kaggle.com/datasets/vigneshvenkateswaran/bot-iot.

[31] IoT-23 dataset link: Accessed on: August, 2025, Available: https://www.kaggle.com/datasets/astralfate/iot23-dataset.

[32] CICIDS2019 dataset link: Accessed on: August, 2025, Available: https://www.kaggle.com/datasets/tarundhamor/cicids-2019-dataset.