frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 28 July 2025
DOI 10.3389/frai.2025.1593944

OPEN ACCESS

EDITED BY
Sarunas Girdzijauskas,
Royal Institute of Technology, Sweden

REVIEWED BY
Saurav Verma,
SVKM's Narsee Monjee Institute of Management Studies, India
Dilip Motwani,
University of Mumbai, India
Amira Soliman,
Halmstad University, Sweden

*CORRESPONDENCE
Mohsen Imani
✉ m.imani@uci.edu

RECEIVED 14 March 2025
ACCEPTED 07 July 2025
PUBLISHED 28 July 2025

CITATION
Masukawa R, Yun S, Jeong S, Huang W, Ni Y, Bryant I, Bastian ND and Imani M (2025) PacketCLIP: multi-modal embedding of network traffic and language for cybersecurity reasoning. *Front. Artif. Intell.* 8:1593944. doi: 10.3389/frai.2025.1593944

PacketCLIP: multi-modal embedding of network traffic and language for cybersecurity reasoning

Ryozo Masukawa[1], Sanggeon Yun[1], Sungheon Jeong[1], Wenjun Huang[1], Yang Ni[1], Ian Bryant[1], Nathaniel D. Bastian[2] and Mohsen Imani[1]*

[1]Department of Computer Science, University of California, Irvine, Irvine, CA, United States, [2]Department of Electrical Engineering & Computer Science, United States Military Academy, West Point, NY, United States

Traffic classification is vital for cybersecurity, yet encrypted traffic poses significant challenges. We introduce PacketCLIP which is a multi-modal framework combining packet data with natural language semantics through contrastive pre-training and hierarchical Graph Neural Network (GNN) reasoning. PacketCLIP integrates semantic reasoning with efficient classification, enabling robust detection of anomalies in encrypted network flows. By aligning textual descriptions with packet behaviors, PacketCLIP offers enhanced interpretability, scalability, and practical applicability across diverse security scenarios. With a 95% mean AUC, an 11.6% improvement over baselines, and a 92% reduction in intrusion detection training parameters, it is ideally suited for real-time anomaly detection. By bridging advanced machine-learning techniques and practical cybersecurity needs, PacketCLIP provides a foundation for scalable, efficient, and interpretable solutions to tackle encrypted traffic classification and network intrusion detection challenges in resource-constrained environments.

KEYWORDS

contrastive pre-training, graph neural network, machine learning, multimodal, reasoning

1 Introduction

Traffic classification plays a crucial role in modern network security analytics, significantly influencing areas such as threat detection and micro-segmentation strategies. As networks become increasingly dynamic, the ability to accurately classify traffic is essential for enhancing security and swiftly responding to business needs. Traditionally, traffic classification techniques relied on inspecting packet headers and payloads. However, the rise of encrypted and anonymized traffic presents significant challenges by obscuring content, making it harder to distinguish between benign and malicious flows.

Recent advances in machine learning, particularly with pre-trained models based on architectures like BERT (Devlin, 2018; Lin et al., 2022; Meng et al., 2024) and masked autoencoders (MAEs) (Zhao et al., 2023), have attempted to address this issue and achieved state-of-the-art performance in various security-related tasks, including encrypted traffic classification. Hyperdimensional Computing (HDC)-based hardware-efficient methods have also been proposed (Lu et al., 2024). These methods leverage deep learning to identify patterns in packet metadata and encrypted content, bypassing the need for

payload inspection. Despite their technical advancements, these models still struggle to track and interpret the semantics of network behaviors, particularly when trying to discern the underlying intent or strategy of cyberattacks. Reasoning about the semantics of cyberattacks remains a key research challenge. In the field of video anomaly detection, MissionGNN (Yun et al., 2025), a cutting-edge hierarchical graph neural network (GNN) model, has demonstrated exceptional capability in reasoning about anomalies using mission-specific knowledge graphs (KGs). By incorporating node embeddings derived from dual modalities—natural language and image data—and employing joint-embedding models such as CLIP (Radford et al., 2021), MissionGNN effectively reasons across both visual and textual domains. This success prompts an intriguing question: *Can this hierarchical GNN-based reasoning be adapted for encrypted traffic detection, and if so, could it address the persistent challenge of semantic reasoning in the cybersecurity domain?* This question arises from the conceptual similarity between video (a sequence of images) and network flows (a sequence of packets), which, while differing in modality, share a sequential structure.

To explore this possibility, we propose a semantic reasoning framework for encrypted traffic detection, illustrated in Figure 1. In our framework, the user first defines a specific task in encrypted traffic detection (e.g., detecting a Denial of Service (DoS) attack as shown in Figure 1a). Then, a Large Language Model (LLM) generates a KG, which is a Directed Acyclic Graph (DAG) serving as an abstract representation of the task to be used later as a medium for reasoning (Figure 1b).

To adapt hierarchical GNN reasoning to encrypted traffic detection, we need a joint-embedding model capable of mapping both encrypted traffic data and Natural Language (NL) into a unified vector space (Figure 1c). This approach allows us to leverage the semantic reasoning capabilities of hierarchical GNNs while addressing the unique challenges posed by encrypted network traffic (Figure 1d). To enable alignment between text and packet data, we propose **PacketCLIP**, which utilizes recent advancements in LLMs (Achiam et al., 2023; Touvron et al., 2023) to create a multi-modal joint embedding via contrastive pretraining. Inspired by Contrastive Language–Image Pre-training (CLIP), which links images with text, PacketCLIP connects packet-level traffic data with semantic descriptions. This alignment not only improves traffic classification accuracy but also provides human operators with NL explanations of packet behavior within the network flow, enhancing interpretability.

We conducted experiments to evaluate the effectiveness and efficiency of PacketCLIP in conjunction with hierarchical GNN reasoning. The results demonstrate that uses features from PacketCLIP achieves not only high classification accuracy but also significant improvements in robustness and efficiency. Specifically, hierarchical reasoning with graph neural network delivers an impressive **11.6**% mean Area Under the Curve (mAUC) of the receiver operating characteristics (ROC) improvement compared to baseline methods. Notably, it maintains **95**% mAUC performance when trained on just 30% of the data, significantly outperforming ET-BERT, which achieves only 50% mAUC under the same conditions. In addition to its performance advantages, the training of GNN reasoning model using PacketCLIP embeddingis highly efficient, achieving a **92**% reduction in the number of trainbale parameters and a **98**% reduction in FLOPs compared to existing methods. These efficiency gains underscore the model's ability to deliver strong performance with a significantly smaller computational footprint. Overall, these findings highlight PacketCLIP's capability to generalize effectively in data-constrained scenarios and its suitability for practical deployment in environments with limited computational resources.

Finally, we evaluated the performance of both PacketCLIP and its hierarchical GNN within a real-time traffic intrusion detection framework using the ACI-IoT-2023 dataset (Bastian et al., 2023). Our results show that PacketCLIP effectively aligns packet and text modalities, while the hierarchical GNN achieves robust and energy-efficient intrusion detection. Because our GNN-based reasoning framework is intended for practical deployment in routers (Figure 1e), these results underscore its potential for real-world applications. The key contributions of this research are as follows:

- Proposed PacketCLIP, a multi-modal framework aligning encrypted traffic data with NL.
- Introduced contrastive pretraining and hierarchical GNN reasoning for robust intrusion detection, outperforming baselines by **11.6**% in mean ROC-AUC scores.
- Showed strong data scarcity resilience, maintaining **95**% mAUC even with **30**% training data, compared to **70**% for ET-BERT.
- Achieved **92**% parameter and **98**% FLOPs reduction for training hierarchical GNN reasoning, enabling deployment in resource-constrained environments.
- Validated on real-world datasets, combining robust traffic classification with efficient and scalable anomaly detection for practical network security applications.

2 Background and related works

This section aims to highlight advancements in traffic intrusion detection, outlining significant progress while identifying ongoing challenges. By examining modern approaches, particularly those leveraging machine learning and GNNs, we underscore the field's evolution and remaining challenges in achieving effective, privacy-preserving, and interpretable detection techniques. We also emphasize the differences between our proposed method and previous works.

Port-based classification methods (Moore et al., 2001), which historically provided effective means for categorizing network traffic, have encountered limitations due to dynamic port allocations, rendering it difficult to track application-specific

Abbreviations: AUC, Area Under the Receiver Operating Characteristic Curve; mAUC, Mean Area Under the ROC Curve; DPI, Deep Packet Inspection; FLOPS, Floating Point Operations per Second; GNN, Graph Neural Network; IDS, Intrusion Detection System; IoT, Internet of Things; InfoNCE, Information Noise-Contrastive Estimation; KG, Knowledge Graph; LLM, Large Language Model; NLP, Natural Language Processing; ROC, Receiver Operating Characteristic; SL, Supervised Learning; SSL, Self-Supervised Learning.

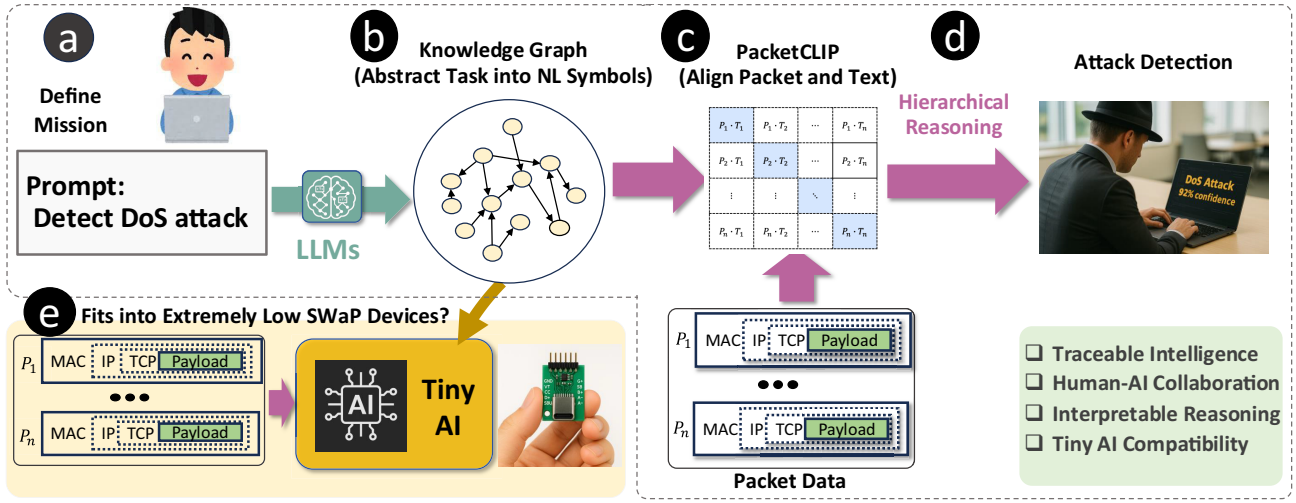

FIGURE 1
Semantic AI framework for detecting traffic related to specific cyber-attacks defined by a user **(a)**, combining LLM-driven knowledge graphs **(b)**, PacketCLIP alignment **(c)**, hierarchical reasoning **(d)**, and Tiny AI to enable efficient, interpretable, and traceable detection on low-resource devices **(e)**.

patterns accurately. Traditional deep packet inspection (DPI) techniques (Papadogiannaki and Ioannidis, 2021), which analyze data payloads for distinguishing patterns, have similarly become impractical, especially for encrypted traffic. The computational burden and diminishing returns on accuracy for DPI methods, as encryption becomes more widespread, highlight the critical need for machine learning-driven approaches that accommodate complex traffic patterns while maintaining privacy. Statistical feature-based approaches leverage manually selected traffic features, requiring substantial domain expertise (Hayes and Danezis, 2016; Panchenko et al., 2016; Zaki et al., 2022; Taylor et al., 2016). For instance, AppScanner (Taylor et al., 2016) utilizes statistical attributes of packet sizes to train random forest classifiers; however, these methods suffer from limitations in capturing high-level semantic patterns essential for robust intrusion detection. Recent works have introduced GNN-based frameworks for enhanced traffic classification, leveraging graph structures to capture relational dependencies within traffic data (Huoh et al., 2022; Zhang et al., 2023, 2024; Alrahis et al., 2023). Among them, TFE-GNN (Zhang et al., 2023) is notable for modeling packet payloads at the byte level, treating each byte as a node and creating edges based on point-wise mutual information (PMI) between nodes. Additionally, a novel contrastive learning-based intrusion detection framework extending TFE-GNN has shown promising results. However, these GNN-based methods often struggle with interpretability, as they rely on encrypted byte representations that do not lend themselves to human understanding. Consequently, these methods may fall short in supporting human security analysts in devising precise micro-segmentation policies.

In the field of video anomaly detection, MissionGNN (Yun et al., 2025) has demonstrated state-of-the-art performance by employing KG reasoning techniques and shows powerful following works. Building upon this approach, we introduce a novel framework that combines a GNN-based reasoning component with PacketCLIP, a cross-modal embedding model designed to align packet data with NL descriptions within a shared vector space. To the best of our knowledge, this is the first integration of NL processing with GNN-based network traffic intrusion detection, facilitating intuitive and interpretable reasoning over encrypted traffic patterns.

3 Methodology

3.1 Mission-specific knowledge graph generation

To enable traffic classification and reasoning about the semantics of the attack, the mission-specific KG generation framework is used to create a KG that extracts relevant information from a given packet. In encrypted traffic detection, each mission-specific KG represents structured knowledge about a particular event or scenario. The process begins by obtaining a set of vocabularies for each event, referred to as **Key Concepts**. This is done using an LLM such as GPT-4o (Achiam et al., 2023) with the prompt: *"List $V(\in \mathbb{N})$ typical vocabularies to represent [event name]? Note: Everything should be a single word."* For example, in case of DoS attack, LLM may provide key concepts as follows, $K = \{\texttt{flood}, \texttt{botnet}, \texttt{amplification}, \texttt{target}, \texttt{saturation}\}$.

Next, we expand K by querying the LLM with the prompt: *"What are associated words with vocabularies in set K?"* This produces a set of associated vocabularies $K_a^{(i)}$ for each key concept (e.g., for "flood": $K_a^{(1)} = \{\texttt{overwhelm}, \texttt{packetstorm}\}$), ensuring no overlap with the original set ($K \cap K_a^{(i)} = \emptyset$). The set K is updated using the equation:

$$K = K \cup K_a^{(i)} \quad (1 \le i \le N) \tag{1}$$

For instance, after the first iteration: $K = \{\texttt{flood}, \ldots\} \cup \{\texttt{overwhelm}, \texttt{packetstorm}\}$. This process is repeated for N iterations, after which edges are drawn from the $(i-1)$th key

concept to the *i*th key concept, forming a hierarchical directed acyclic graph (DAG).

On top of the mission-specific KG, a sensor node is added, containing sensory information such as packet data encoded by joint-embedding models like PACKETCLIP. Directed edges are projected from the sensor node to key concept nodes (e.g., $s \rightarrow$ `flood`, $s \rightarrow$ `botnet`), and related concept nodes also project edges to an embedding node, which aggregates messages passed through the graph.

This KG design allows the GNN to pass interpretable messages by embedding multimodal information from all nodes into a unified vector space thanks to the PACKETCLIP alignment.

3.2 NL explanation for intrusion

A key challenge in our framework is achieving a rich textual representation of cyberattacks, as, to the best of our knowledge, no existing datasets related to network traffic classification include NL descriptions. Most current datasets (Bastian et al., 2023; Neto et al., 2023; Draper-Gil et al., 2016; Dadkhah et al., 2022) generally provide two main types of data: (1) raw packets stored in PCAP files with corresponding labels and (2) tabular data representing network flows derived from these PCAP files, typically in CSV format (see 1 in Figure 2). Our approach focuses specifically on the tabular flow files, as certain columns within this data have the potential to serve as elements in generating NL descriptions for each packet's semantic context. As shown in Figure 2, we first convert the tabular data at row i into a template-based text expression $\mathbf{T}_i^*$ (Step 2) by embedding each column value into structured sentences.

To train PACKETCLIP with a rich and diverse vocabulary comparable to LLMs, we incorporate a mission-specific KG that aligns with each tabular data row's label, such as those associated with DoS attack detection. We enhance textual variety by sampling nodes from the KG and integrating them into template-based descriptions (Step 3). Recognizing the limitations of static templates, we use lightweight LLMs to paraphrase descriptions, ensuring grammaticality and semantic diversity (Step 4). However, due to the lack of direct correspondence between knowledge graph keywords and observable flow features—especially in encrypted or anonymized traffic—there exists a fundamental "chicken-and-egg" problem. This motivates our contrastive pretraining: by aligning packet features with LLM-generated concepts, PACKETCLIP learns associations between observed data and high-level semantics, even without explicit mapping. Modern LLMs such as GPT (Achiam et al., 2023) and LLaMA (Touvron et al., 2023) have demonstrated strong paraphrasing capabilities, and are frequently used as paraphrase oracles in recent studies (Jayawardena and Yapa, 2024). Accordingly, we leverage LLM-paraphrased sentences as diverse packet descriptions. To mitigate LLM hallucination, similar to CLIP (Radford et al., 2021), we prepend each text with a prompt like "A network traffic of label," followed by the generated paraphrase, and append relevant key concepts—especially for anomalous flows—to serve as nodes in the GNN reasoning module. Example prompts and outputs are shown in Figure 3.

Finally, we can obtain a diverce text expression of each packet (Step 5). This augmentation not only mitigates the constraint of limited class labels but also elevates the diversity of template-based learning, enabling broader generalization.

This NL augmentation method we described so far can be formulated as follows. We first obtain $n(\in \mathbb{N})$ template-based text data ($\mathcal{D}_T^* = \{\mathbf{T}_i^*\}_{i=1}^n$) from flow that is represented as tabular data. After this, we use LLMs as follows

$$\mathbf{T}_i = LLM(\mathbf{T}_i^*)(1 \leq i \leq n), \tag{2}$$

where *LLM* indicates LLMs and we obtain augmented highly diverse text expression of each packet.

3.3 PacketCLIP contrastive pre-training

Using the approach described in the previous section, PACKETCLIP obtains a set of text data, $\mathcal{D}_T = \{\mathbf{T}_i\}_{i=1}^n$, paired with corresponding packet payload data, $\mathcal{D}_P = \{\mathbf{p}_i\}_{i=1}^n$. We clarify the pre-training of PACKETCLIP described in Figure 4a.

During contrastive pre-training, we keep the weights of both the text encoder (f_T) and the packet encoder (f_P) fixed. This strategy preserves the consistency of text representations and

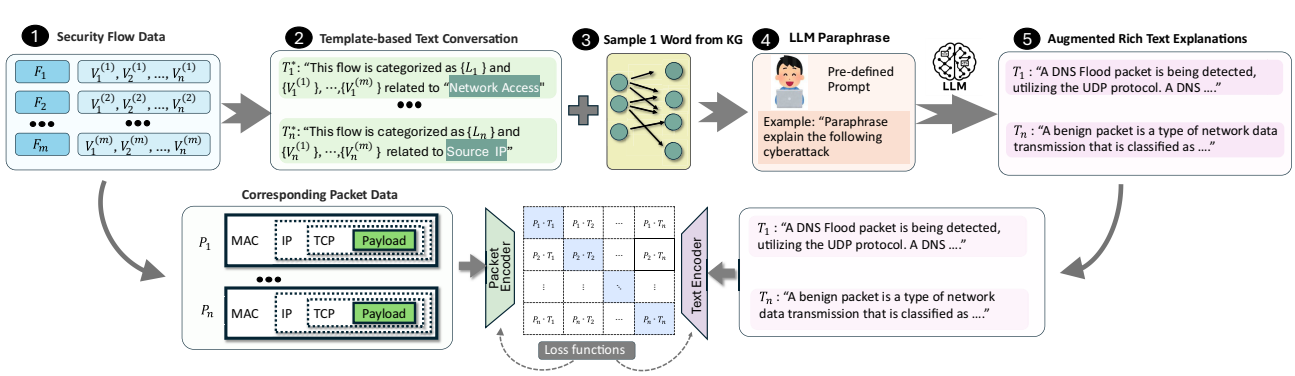

FIGURE 2
A framework to generate NL explanations for intrusion scenarios by mapping tabular security flow data (1) to text templates (2), leveraging LLM-generated knowledge graphs (3), utilizing LLMs for paraphrased explanations (4), and producing interpretable descriptions of network events (5).

System prompt:
You are a cybersecurity expert who explains cyber security incidents.

User prompt:
Paraphrase the following and explain what's going on in one sentence (approximately up to 20 words): <input template text:$\mathbf{T}_i^*$>:

Example Outputs ($\mathbf{T}_i$)
Anomalous packet description:
A network traffic of OS Scan: A malicious packet is sent to map a wireless network's infrastructure, classified as an OS Scan in a Recon attack. Key Concepts : {*Kernel*}

Benign packet description:
A network traffic of Benign: A packet from a wired connection is identified as benign, part of a non-malicious attack group.

FIGURE 3
LLM prompt and sample outputs illustrating paraphrasing and concise explanation of cybersecurity network traffic incidents.

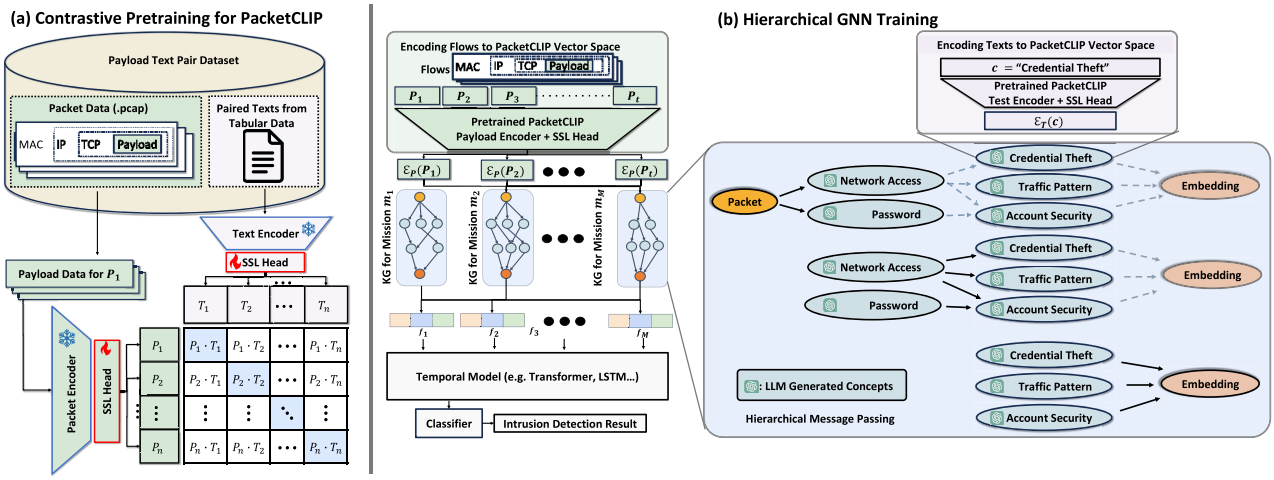

FIGURE 4
(a) The overall architecture of the contrastive pre-training process for PACKETCLIP, including encoding packets and paired texts for learning. **(b)** A mission-specific hierarchical GNN framework that integrates PACKETCLIP with temporal models and classifiers to derive intrusion detection results.

avoids catastrophic forgetting (French, 1999). Rather than fine-tuning the large pre-trained models used for the packet and text encoders, we adopt a method from Gupta et al. (2022), introducing a simple linear projection layer as a self-supervised learning (SSL) head for each encoder: one for the packet encoder (g_P) and another for the text encoder (g_T). During pre-training, only these projection layers are updated.

The text $\mathbf{t} \in \mathcal{D}_T$ and its paired packet $\mathbf{p} \in \mathcal{D}_P$ are encoded as follows:

$$\mathbf{z_t} = g_T \circ f_T(\mathbf{t}), \quad \mathbf{z_p} = g_P \circ f_P(\mathbf{p}). \tag{3}$$

For contrastive pre-training, we use the InfoNCE loss (Chen et al., 2020) l as defined below:

$$l(\mathbf{z}_t, \mathbf{z}_p; \mathcal{Z}^{\setminus}) = -\log \frac{\exp\left(\cos(\mathbf{z_t}, \mathbf{z}_p^+)/\tau\right)}{\sum_{\mathbf{z_p} \setminus \in \mathcal{Z}^{\setminus}} \exp\left(\cos(\mathbf{z_t}, \mathbf{z}_p^{\setminus})/\tau\right)}, \tag{4}$$

where $\mathcal{Z}^{\setminus}$ denotes a set of embedded vectors sampled from $\mathcal{D}_P$ that excludes the packet vector ($\mathbf{z}_p^+$) matching the text vector $\mathbf{z_t}$, and $\tau > 0$ represents the temperature parameter. This loss function encourages alignment between embeddings from paired text and packet instances while pushing apart embeddings from different instances.

By completing this contrastive pre-training process, PACKETCLIP learns robust, aligned representations for both text and packet data, enhancing its ability to capture semantic connections between textual and packet-based cyberattack data.

3.4 Downstream hierarchical GNN reasoning module

After generating M KGs G_{m_i} ($1 \leq i \leq M$, where m_i denotes the i-th mission), we train a hierarchical GNN model to classify events or anomalies in network traffic data (Figure 4b). GNNs capture relational information using feature vectors for each node, connecting packet node features $x_{s,m_i}^{(0)}$ for packet at timestamp t (P_t) from the packet encoder $\mathcal{E}_P(= g_P \circ f_P)$ and concept node

features $x^{(0)}_{c,m_i}$ for each concept c from the text encoder $\mathcal{E}_T(= g_T \circ f_T)$ as follows:

$$\boldsymbol{x}^{(0)}_{s,m_i} = \mathcal{E}_P(P_t), \quad \boldsymbol{x}^{(0)}_{c,m_i} = \mathcal{E}_T(c) \tag{5}$$

A multi-layer perceptron (MLP) then embeds these node features at layer $l(1 \leq l \leq L)$ of GNN as follows:

$$\boldsymbol{x}^{(l)}_{m_i} = W^{(l)}_{m_i}\boldsymbol{x}^{(l-1)}_{m_i} + \boldsymbol{b}^{(l)}_{m_i}, \tag{6}$$

where $W^{(l)}_{m_i}$ denotes a trainable weight matrix and $\boldsymbol{b}^{(l)}_{m_i}$ indicates the bias. The core idea is hierarchical message passing, where messages are propagated through three levels of the KG hierarchy: packet nodes to key concepts, key concepts to associated concept nodes, and finally to embedding nodes. This structure allows efficient and targeted aggregation of information across modalities, resulting in interpretable, goal-oriented embeddings.

Hierarchical message passing from node v to neighboring node in the previous hierarchy u at layer l is recursively defined as:

$$\boldsymbol{x}^{(l)}_v = \frac{1}{|\mathcal{N}^{(h-1)}(v)|} \sum_{u \in \mathcal{N}^{(h-1)}(v)} \phi^{(l)}(\boldsymbol{x}^{(l-1)}_v \cdot \boldsymbol{x}^{(l-1)}_u) \tag{7}$$

where ϕ indicates the activation function and $\mathcal{N}^{(h)}(v)$ represents the neighbors of node v at hierarchy h. The final embeddings feature node $\boldsymbol{x}_{emb}$ for each mission-specific KG are combined into a single vector:

$$\boldsymbol{f}^{(t)} = [\boldsymbol{x}^{(L)}_{\text{emb},m_1}, \boldsymbol{x}^{(L)}_{\text{emb},m_2}, \ldots, \boldsymbol{x}^{(L)}_{\text{emb},m_M}] \tag{8}$$

For each packet F_t, the sequence of tokens X_t is constructed as:

$$X_t = \{\boldsymbol{f}^{(t-A+1)}, \boldsymbol{f}^{(t-A+2)}, \ldots, \boldsymbol{f}^{(t)}\},$$

where A represents a hyperparameter that specifies a fixed number of time frames to be input into the temporal model. This sequence is input into a Transformer encoder $\mathcal{T}$ followed by an MLP to produce the final classification output:

$$\hat{\boldsymbol{y}} = \text{Softmax}(\text{MLP}(\mathcal{T}(X_t))) \tag{9}$$

Training leverages cross-entropy loss, smoothing loss, and anomaly localization techniques to optimize the GNN model for network traffic event recognition and anomaly detection tasks following (Yun et al., 2025).

4 Experiments

4.1 Implementation details

For converting tabular data into NL expressions, we used LLaMA 3 (Touvron et al., 2023). For KG generation, we employed an automated framework powered by GPT-4o (Achiam et al., 2023). The PACKETCLIP packet encoder was implemented using the *ET-BERT* pre-trained encoder (Lin et al., 2022), while the text encoder relied on RoBERTa (Liu, 2019). For optimization, we adopted the Adam (Kingma, 2014) optimizer, configured with a learning rate of 5.0×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.8$, and $\epsilon = 1.0 \times 10^{-6}$. Within the hierarchical GNN model, we ensured a consistent dimensionality of $D_{m_i,l} = 8$ for mission m_i at hierarchy l. For the short-term temporal model, an internal dimensionality of 128 was employed, alongside 8 attention heads and the hyperparameter A was set to 30. The training process was conducted over 3,000 steps, utilizing a mini-batch size of 128 samples for each step.

4.2 Datasets

We utilized the ACI-IoT-2023 dataset (Bastian et al., 2023), a comprehensive IoT cybersecurity dataset containing 3,157,430 labeled benign and malicious traffics, including threats like malware, DoS, and botnets. Using an LLM-based paraphrasing method (Figure 2), we generated diverse payload-text pairs to enhance semantic representation.

For PACKETCLIP pretraining evaluated in Section 4.4, the data were categorized into 10 distinct classes: *Benign*, *OS Scan*, *Vulnerability Scan*, *Port Scan*, *ICMP Flood*, *Slowloris*, *SYN Flood*, *UDP Flood*, *DNS Flood*, and *Dictionary Attack*. Notably, 95.31% of the dataset comprises benign traffic, while the class distribution for anomalous categories is detailed in Figure 5. We randomly split the dataset into 80% for training and 20% for testing.

For GNN-based reasoning classification in Section 4.5, we consolidated the dataset into broader categories: *Benign*, *DoS*, *Reconnaissance*, and *Brute Force*. Each sample was treated as a time-series packet sequence by attaching timestamps and sorting chronologically, enabling the GNN models to capture temporal structures akin to those in video analysis. Since the joint embedding learned by PACKETCLIP is optimized for obtaining NL representations rather than direct anomaly classification, we again partitioned the data into 80% for training and 20% for testing. Note that all evaluation metrics reported in subsequent sections are calculated on the held-out test data.

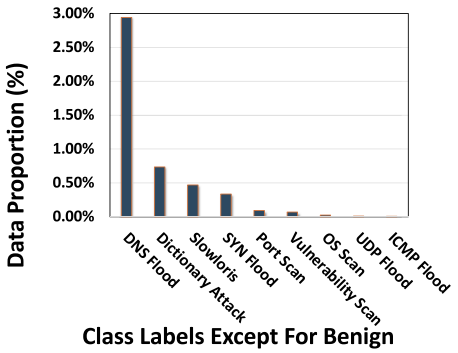

FIGURE 5
Proportion of Class labels.

4.3 Visualization of LLM generated cyberattack semantics

Figure 6 shows the textual explanations generated from the ACI-IoT dataset, emphasizing the most frequent terms used to describe network events. The lower bar graph highlights common words and phrases, such as "attack," "network," and "security," which encapsulate key cybersecurity themes. Above, word clouds visually represent mission-specific vocabularies, showing the terms that form the nodes of corresponding KGs. Together, these visualizations illustrate PACKETCLIP's ability to generate contextually relevant explanations, providing enriched semantic insights into various cyber incidents.

4.4 PacketCLIP semantic classification performance

Baselines: To establish a baseline, we fine-tuned the *ET-BERT* (Lin et al., 2022) packet classifier on the ACI-IoT-2023 dataset, demonstrating the effectiveness of leveraging NL-based semantics for improved classification performance. Additionally, we performed an ablation study to evaluate the contribution of the SSL head in PACKETCLIP. Specifically, we examined three configurations: PACKETCLIP without any SSL head, with a single SSL head applied only to the packet encoder, and with SSL heads applied to both the text and packet encoders. We did not include an analysis of applying a single SSL head to the text encoder because the primary goal of PACKETCLIP's contrastive learning is to align the non-interpretable packet modality with the NL modality. Applying an SSL head solely to the text encoder could potentially degrade valuable NL information, which is counterintuitive to our objective.

Evaluation metrics: For zero-shot performance, we use both macro-averaged Top-1 and Top-5 accuracy, defined as

$$\begin{aligned}\text{Top-1}_{\text{macro}} &= \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i:\,y_i=c}\mathbb{I}\left[y_i=\hat{y}_i^{(1)}\right],\\ \text{Top-5}_{\text{macro}} &= \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i:\,y_i=c}\mathbb{I}\left[y_i\in\{\hat{y}_i^{(1)},\ldots,\hat{y}_i^{(5)}\}\right],\end{aligned} \tag{10}$$

where $C = 10$ is the number of classes, N_c the number of instances in class c, y_i the ground-truth label for instance i, and $\hat{y}_i^{(k)}$ its k-th ranked prediction. For comparison, since the baseline methods can only output a single classification result, we mainly compare with macro top-1 accuracy.

Table 1 shows the performance comparison between each baselines that PACKETCLIP methods perform better than *ET-BERT* fine-tuning in traffic classification. While *ET-BERT* demonstrates moderate accuracy, PACKETCLIP using only packet information

TABLE 1 Comparison of models highlighting PACKETCLIP configurations achieving highest macro top-1 accuracy and macro top-5 accuracy.

Method	top-1	top-5
ET-BERT (Lin et al., 2022)	0.730	-
PACKETCLIP (No SSL Head)	0.001	0.955
PACKETCLIP (SSL Head Only on Packet Encoder)	0.831	**0.991**
PACKETCLIP (SSL Head on both Encoder)	**0.856**	0.961

Bold values indicate the highest performance in both metrics.

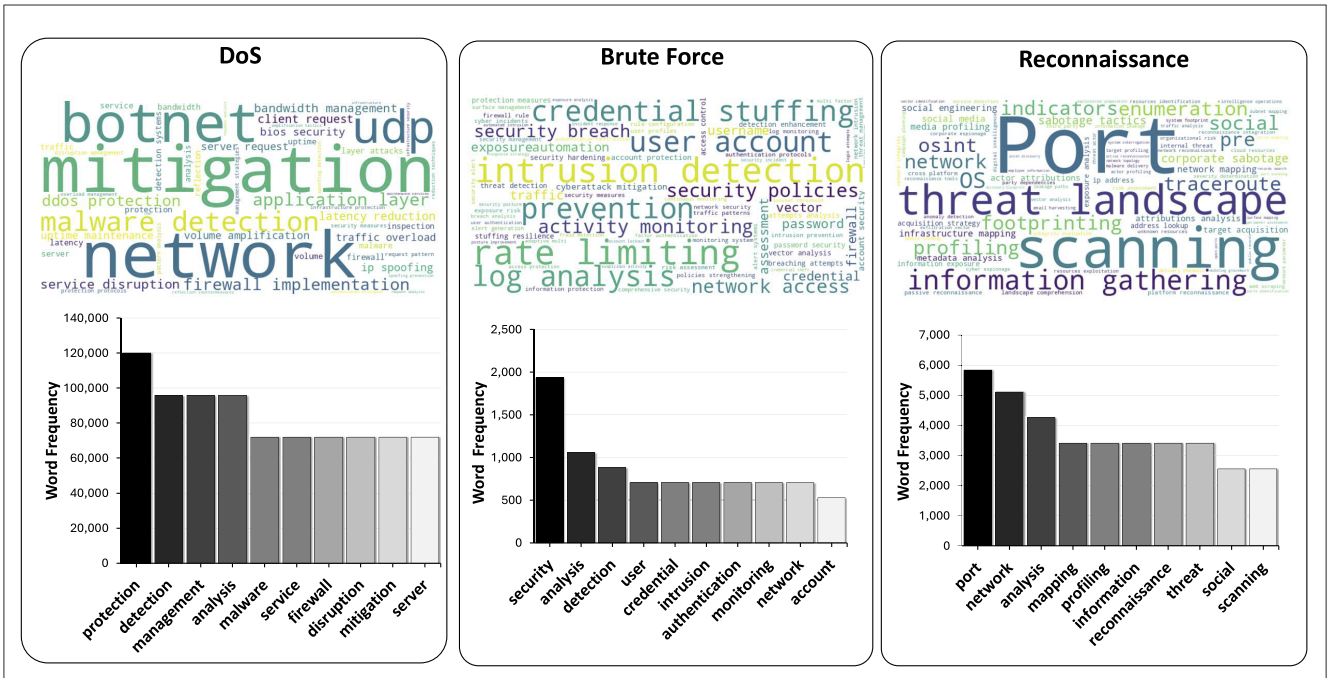

FIGURE 6
Word clouds and top 10 frequent vocabularies for DoS, Brute Force, and Reconnaissance missions from ACI-IoT-2023, highlighting key terms like "botnet," "credential," and "port scanning" for respective categories.

achieves a noticeable improvement, with nearly perfect reliability when considering multiple predictions. PACKETCLIP, when incorporating both packet data and contextual information, further enhances its ability to make accurate top predictions but slightly reduces its broader predictive range. Overall, PACKETCLIP offers superior accuracy, especially when combining packet and contextual details, making it a more effective method for precise traffic classification in network management. At the same time, PACKETCLIP's performance comparison in Figure 7 highlights the critical role of the SSL head. When incorporated, the SSL head significantly boosts both macro top-1 and macro top-5 accuracy, demonstrating its ability to enhance classification reliability and precision. Without only having one SSL head in packet Encoder, the performance drops notably in the middle of the training, underlining its importance in leveraging SSL heads on both encoders effectively. This comparison underscores the value of the SSL head in extracting meaningful features from both packet data and contextual information, enabling more accurate predictions in traffic classification. The results clearly establish the SSL head as a crucial component for achieving superior classification performance in PACKETCLIP.

4.5 Evaluation on hierarchical GNN reasoning

Baselines: The ACI-IoT-2023 dataset, utilized in our experiments, has been previously explored in works such as AIS-NIDS (Farrukh et al., 2024). AIS-NIDS introduced a novel approach involving serialized RGB image transformations for packet-level feature extraction and employed basic machine learning models, including XGBoost and LightGBM, for intrusion detection. However, AIS-NIDS relies on closed-set classifiers and lacks publicly available code for its CNN preprocessing pipeline, posing challenges for reproducibility and adaptation to alternative methods. To address these limitations, we adopt baselines that incorporate the PACKETCLIP packet feature encoder in conjunction with various machine learning models. Specifically, we evaluate the following configurations: PACKETCLIP + XGBoost, PACKETCLIP + LightGBM, and PACKETCLIP (packet) + Deep Neural Network (DNN). Furthermore, to assess the performance of an external baseline, we fine-tuned *ET-BERT* (Lin et al., 2022) on the same dataset for comparative analysis.

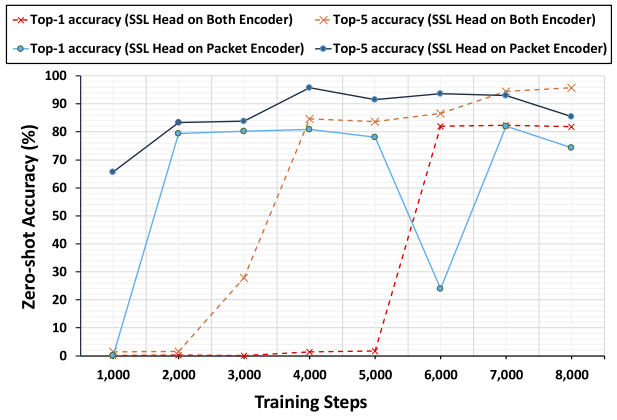

FIGURE 7
Zero-shot accuracy change during training shows a trade-off: SSL on both encoders improves faster but is less stable, while SSL only on the packet encoder progresses slower but is more stable.

Evaluation metrics : To evaluate our method, following the convention of previous research (Yun et al., 2025; Bhavsar et al., 2023; Ajagbe et al., 2024), We adopted the Area Under the Receiver Operating Characteristic Curve (ROC AUC) as our evaluation metric, which offers a robust measure of performance across all classification thresholds. AUC is particularly suitable for cybersecurity tasks, given the highly imbalanced nature of datasets, where attack instances are far fewer than benign traffic. By focusing on ranking instances correctly, AUC ensures a comprehensive evaluation of anomaly detection performance under varying conditions.

We present the result of our approach in Table 2, showing an average AUC score gain of more than **11.6**% compared to baseline methods. This substantial improvement highlights significant advancements in intrusion detection, combining semantic reasoning, interpretability, and advanced classification capabilities. By employing PACKETCLIP alongside hierarchical GNN reasoning, we provide a robust and innovative solution tailored for real-time anomaly detection in IoT networks, demonstrating the potential for enhanced semantic understanding and improved adaptability compared to traditional methods.

4.6 Hardware efficiency analysis during training

The computational efficiency of training hierarchical GNN reasoning module in PACKETCLIP is compared against *ET-BERT* (Lin et al., 2022), TFE-GNN (Zhang et al., 2023), and CLE-TFE (Zhang et al., 2024) in terms of FLOPs and parameter count. These baselines are chosen due to their proven effectiveness in encrypted traffic detection tasks. GNN reasoning module using the joint embedding vectors from PACKETCLIP demonstrates a significant improvement, achieving a 107M reduction in training parameters compared to fine-tuning *ET-BERT*, as shown in Figure 8b. This reduction highlights the module's streamlined architecture, which effectively aggregates semantic information through hierarchical message passing while minimizing parameterization. Moreover, as shown in Figure 8a, the FLOPs of the GNN reasoning module are approximately one-thirtieth of *ET-BERT*, while still delivering competitive performance in traffic anomaly detection tasks. These results emphasize the training scalability of the hierarchical GNN module using PACKETCLIP joint embeddings, particularly for resource-constrained environments like IoT networks, where computational overhead is a critical concern. By incorporating GNN reasoning using PACKETCLIP balances performance and efficiency, making it highly suitable for real-time intrusion detection applications.

TABLE 2 AUC scores for individual classes and their mean AUC are compared across models.

Model	Benign	DoS	Reconnaissance	Brute force	Average
ET-BERT Fine-Tuning (Lin et al., 2022)	0.717	0.731	0.752	0.784	0.746
PacketCLIP + XGBoost	0.522	0.500	0.477	0.458	0.489
PacketCLIP + LightGBM	0.961	0.544	0.974	0.567	0.761
PacketCLIP + DNN	0.978	0.724	0.996	0.623	0.830
PacketCLIP + GNN Reasoning	**0.996**	**0.930**	**0.999**	**0.909**	**0.946**

The results highlight the performance superiority of PacketCLIP with GNN Reasoning, achieving the highest scores in all categories. Bold values indicate the highest performance in both metrics.

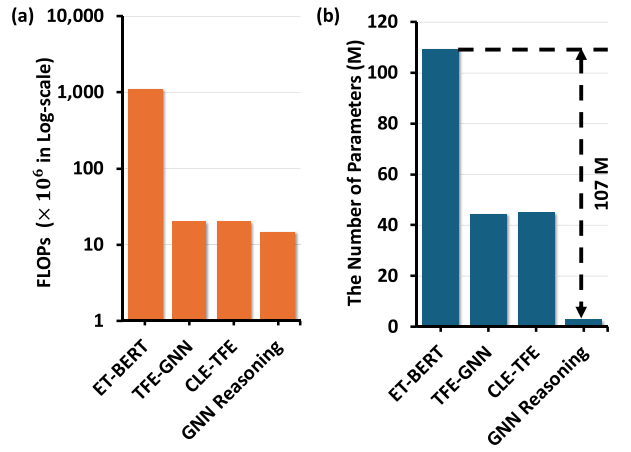

FIGURE 8
(a) Comparison of models based on FLOPs (×10^6 Log-scale) and **(b)** on the number of parameters (M) for training.

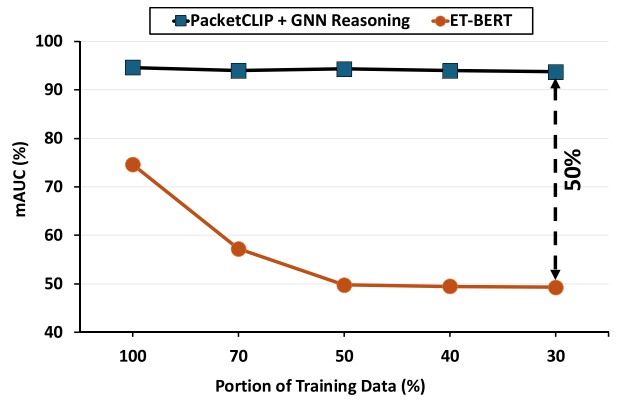

FIGURE 9
Robustness against Data Scarcity Analysis by mAUC Comparison of PacketCLIP + GNN Reasoning and *ET-BERT* with Varying Training Data (100%, 70%, 50%, 40%, 30%).

4.7 GNN reasoning robustness of scarce data availability

To assess the robustness of PacketCLIP + GNN Reasoning under varying levels of data availability, we conducted experiments using the ACI-IoT-2023 dataset, selecting *ET-BERT* (Lin et al., 2022) as a baseline for comparison. Both models were trained on three different proportions of the training data: 100%, 70%, 50%, 40%, and 30%, while the test set remained consistent across all experiments to ensure a fair and controlled evaluation. The mAUC, again, served as the primary performance metric.

Figure 9 presents the results of this experiment. PacketCLIP + GNN Reasoning consistently achieved high mAUC scores (∼ 95%) across all training data splits, demonstrating strong generalization even with limited data. In contrast, *ET-BERT* exhibited notable performance degradation, with mAUC dropping from ∼ 70% at 100% training data to ∼ 50% at 30%. These findings emphasize the robustness of PacketCLIP + GNN Reasoning, making it well-suited for scenarios with constrained training data.

5 Discussions and limitations

In this paper, we introduce PacketCLIP, a novel framework that aligns encrypted packet data and natural language explanations within a shared vector space through contrastive pre-training. Our results show that PacketCLIP achieves up to a 12% improvement in zero-shot detection accuracy on novel examples of known classes compared to static, one-size-fits-all graph-based methods. Instead of generating explicit human-readable labels, PacketCLIP enables querying natural language expressions in a manner similar to CLIP, providing flexible interpretability for detected events. Furthermore, we demonstrate that a downstream hierarchical GNN reasoning module, leveraging the PacketCLIP joint embedding space, can be trained to detect traffic anomalies with fewer parameters, enhancing efficiency in adapting to new cyberattacks.

Despite these strengths, PacketCLIP also presents several limitations that highlight opportunities for future work. First, the quality of the generated knowledge graphs is fundamentally dependent on the underlying language model: outdated or domain-misaligned LLMs may introduce irrelevant or redundant concepts, adding noise to the GNN and slowing inference. Second, although we demonstrate a reduction in trainable parameters within the hierarchical GNN framework, the initial knowledge graph generation and packet embedding steps still require PacketCLIP inference, which imposes a hardware burden—particularly because loading the ET-BERT (Lin et al., 2022) is essential for PacketCLIP. Hardware acceleration of the PacketCLIP encoder therefore represents an important avenue for future improvement. Third, our explanation module currently uses a fixed set of templates, paraphrased by the LLM, but we do not perform explicit validation of these paraphrased textual expressions for each packet. As a result, this approach may introduce irrelevant words during pre-training

and struggle to handle rare or novel attack patterns, potentially leading to vulnerabilities from unintentional adversarial samples or insufficient template coverage. Finally, while our experiments focus on encrypted packet metadata, extending PACKETCLIP to other modalities—such as host logs or full packet inspections—would likely require new vocabulary prompts, retraining of the contrastive encoder, and careful consideration of privacy implications. Addressing these limitations will further strengthen PACKETCLIP's applicability across diverse operational settings and accelerate its adoption in production security systems.

6 Conclusions

We introduced PACKETCLIP, a multi-modal framework integrating packet-level data with NL semantics to advance encrypted traffic classification and intrusion detection. By combining contrastive pre-training and a downstream hierarchical GNN reasoning, PACKETCLIP demonstrates robustness in both performance and efficiency. PACKETCLIP itself achieved an 11.6% higher top-1 accuracy compared to baseline models, and the downstream GNN reasoning module consistently delivering superior mAUC scores of approximately 95%, even with only 30% of the training data. These results highlight the resilience of hierarchical GNN reasoning in PACKETCLIP to handle data scarcity and its ability to generalize effectively. Furthermore, hierarchical GNN module trained by PACKETCLIP joint embeddings reduces model size by 92% and computational requirements by 98%, making it highly efficient for real-time applications in resource-constrained environments like IoT networks. By providing interpretable semantic insights alongside robust anomaly detection, PACKETCLIP harmonizes advanced machine learning techniques and practical cybersecurity solutions, setting a strong foundation for future developments in multi-modal network security frameworks. Integrating NL semantics improves detection capabilities and offers a more intuitive understanding of network behaviors, crucial for cybersecurity professionals to diagnose and respond to threats effectively. Future work includes analyzing PACKETCLIP's versatility by applying it to a broader range of network security tasks and exploring its performance in diverse network environments and hardware acceleration for its encoders.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

RM: Writing – original draft, Validation, Methodology, Writing – review & editing. SY: Funding acquisition, Visualization, Formal analysis, Validation, Resources, Project administration, Data curation, Supervision, Software, Writing – review & editing, Methodology, Investigation, Conceptualization. SJ: Writing – review & editing. WH: Supervision, Writing – review & editing. YN: Writing – review & editing, Supervision, Project administration. IB: Writing – review & editing, Data curation. NB: Supervision, Writing – review & editing. MI: Funding acquisition, Conceptualization, Project administration, Supervision, Writing – review & editing.

Funding

The author(s) declare that financial support was received for the research and/or publication of this article. This work was supported in part by the U.S. Military Academy (USMA) under Cooperative Agreement No. W911NF-24-2-0200, as well as the U.S. Army Combat Capabilities Development Command (DEVCOM) C5ISR Center under Support Agreement No. USMA23011. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Military Academy, the U.S. Army, the U.S. Department of Defense, or the U.S. Government.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that Gen AI was used in the creation of this manuscript. Revising grammar and expressions.

Publisher's note

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., et al. (2023). Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ajagbe, S. A., Awotunde, J. B., and Florez, H. (2024). Intrusion detection: a comparison study of machine learning models using unbalanced dataset. *SN Comput. Sci.* 5:1028. doi: 10.1007/s42979-024-03369-0

Alrahis, L., Knechtel, J., and Sinanoglu, O. (2023). "Graph neural networks: a powerful and versatile tool for advancing design, reliability, and security of ICS," in

Proceedings of the 28th Asia and South Pacific Design Automation Conference, 83–90. doi: 10.1145/3566097.3568345

Bastian, N., Bierbrauer, D., McKenzie, M., and Nack, E. (2023). *ACI IoT Network Traffic Dataset 2023*. doi: 10.21227/qacj-3x32

Bhavsar, M., Roy, K., Kelly, J., and Olusola, O. (2023). Anomaly-based intrusion detection system for iot application. *Discover Internet Things* 3:5. doi: 10.1007/s43926-023-00034-5

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning* (PmLR), 1597–1607.

Dadkhah, S., Mahdikhani, H., Danso, P. K., Zohourian, A., Truong, K. A., and Ghorbani, A. A. (2022). "Towards the development of a realistic multidimensional IoT profiling dataset," in *2022 19th Annual International Conference on Privacy, Security Trust (PST)* (IEEE), 1–11. doi: 10.1109/PST55820.2022.9851966

Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., and Ghorbani, A. A. (2016). "Characterization of encrypted and vpn traffic using time-related features," in *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP)*, 407–414. doi: 10.5220/0005740704070414

Farrukh, Y. A., Wali, S., Khan, I., and Bastian, N. D. (2024). Ais-nids: an intelligent and self-sustaining network intrusion detection system. *Comput. Secur.* 144:103982. doi: 10.1016/j.cose.2024.103982

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* 3, 128–135. doi: 10.1016/S1364-6613(99)01294-2

Gupta, K., Ajanthan, T., Hengel, A., v. d., and Gould, S. (2022). Understanding and improving the role of projection head in self-supervised learning. *arXiv preprint arXiv:2212.11491*.

Hayes, J., and Danezis, G. (2016). "k-fingerprinting: a robust scalable website fingerprinting technique," in *25th USENIX Security Symposium (USENIX Security 16)* (Austin, TX. USENIX Association), 1187–1203.

Huoh, T.-L., Luo, Y., Li, P., and Zhang, T. (2022). Flow-based encrypted network traffic classification with graph neural networks. *IEEE Trans. Netw. Serv. Manag.* 20, 1224–1237. doi: 10.1109/TNSM.2022.3227500

Jayawardena, L., and Yapa, P. (2024). "Parameter efficient diverse paraphrase generation using sequence-level knowledge distillation," in *2024 5th International Conference on Advancements in Computational Sciences (ICACS)* (IEEE), 1–12. doi: 10.1109/ICACS60934.2024.10473289

Kingma, D. P. (2014). Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lin, X., Xiong, G., Gou, G., Li, Z., Shi, J., and Yu, J. (2022). "Et-bert: a contextualized datagram representation with pre-training transformers for encrypted traffic classification," in *Proceedings of the ACM Web Conference 2022*, 633–642. doi: 10.1145/3485447.3512217

Liu, Y. (2019). Roberta: a robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lu, H., Ma, Z., Li, X., Bi, S., He, X., and Wang, K. (2024). "Traffichd: efficient hyperdimensional computing for real-time network traffic analytics," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 1–6. doi: 10.1145/3649329.3657330

Meng, X., Srivastava, A., Arunachalam, A., Ray, A., Silva, P. H., Psiakis, R., et al. (2024). "NSPG: natural language processing-based security property generator for hardware security assurance," in *Proceedings of the 61st ACM/IEEE Design Automation Conference*, 1–6. doi: 10.1145/3649329.3656255

Moore, D., Keys, K., Koga, R., Lagache, E., and Claffy, K. C. (2001). "The $*CoralReef*$ software suite as a tool for system and network administrators," in *15th Systems Administration Conference (LISA 2001)*.

Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., and Ghorbani, A. A. (2023). Ciciot2023: a real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors* 23:5941. doi: 10.3390/s23135941

Panchenko, A., Lanze, F., Pennekamp, J., Engel, T., Zinnen, A., Henze, M., et al. (2016). "Website fingerprinting at internet scale," in *NDSS*. doi: 10.14722/ndss.2016.23477

Papadogiannaki, E., and Ioannidis, S. (2021). A survey on encrypted network traffic analysis applications, techniques, and countermeasures. *ACM Comput. Surv.* 54, 1–35. doi: 10.1145/3457904

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., et al. (2021). "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning* (PmLR), 8748–8763.

Taylor, V. F., Spolaor, R., Conti, M., and Martinovic, I. (2016). "Appscanner: automatic fingerprinting of smartphone apps from encrypted network traffic," in *2016 IEEE European Symposium on Security and Privacy (EuroSP)*, 439–454. doi: 10.1109/EuroSP.2016.40

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yun, S., Masukawa, R., Na, M., and Imani, M. (2025). "Missiongnn: hierarchical multimodal GNN-based weakly supervised video anomaly recognition with mission-specific knowledge graph generation," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (IEEE), 4736–4745. doi: 10.1109/WACV61041.2025.00464

Zaki, F., Afifi, F., Abd Razak, S., Gani, A., and Anuar, N. B. (2022). Grain: granular multi-label encrypted traffic classification using classifier chain. *Comput. Netw.* 213:109084. doi: 10.1016/j.comnet.2022.109084

Zhang, H., Xiao, X., Yu, L., Li, Q., Ling, Z., and Zhang, Y. (2024). One train for two tasks: An encrypted traffic classification framework using supervised contrastive learning. *arXiv preprint arXiv:2402.07501*.

Zhang, H., Yu, L., Xiao, X., Li, Q., Mercaldo, F., Luo, X., et al. (2023). "TFE-GNN: a temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification," in *Proceedings of the ACM Web Conference 2023*, 2066–2075. doi: 10.1145/3543507.3583227

Zhao, R., Zhan, M., Deng, X., Wang, Y., Wang, Y., Gui, G., et al. (2023). "Yet another traffic classifier: a masked autoencoder based traffic transformer with multi-level flow representation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 5420–5427. doi: 10.1609/aaai.v37i4.25674