Impact of Data Extraction Methods on Pseudo-Label Quality in Malicious Behavior Detection of Encrypted Network Traffic*

Chaeeun Won, Yeasul Kim and Hwankuk Kim†

Kookmin University, Seoul, Republic of Korea
{chaeeun4583, gimsul80, rinyfeel}@kookmin.ac.kr

Abstract

The rapid growth of encrypted network traffic has increasingly exposed the limitations of traditional payload-based security inspection techniques, highlighting the need for alternative approaches such as machine learning-based encrypted traffic analysis. However, these approaches generally rely on a large volume of labeled data to achieve robust performance, creating a major bottleneck in practical deployment. To mitigate this dependency, semi-supervised learning—particularly pseudo-labeling—has emerged as a promising solution. While prior research has primarily emphasized comparisons between algorithms, there has been limited investigation into how feature extraction strategies influence the quality of pseudo-labels. This study conducts a quantitative assessment of pseudo-label quality across six feature extraction strategies: Packet, Flow, Encryption, Packet+Encryption, Flow+Encryption, and Total (Packet+Flow+Encryption). Experimental findings reveal that both the Total and Flow+Encryption approaches achieved up to 18% higher pseudo-label quality than other extraction strategies. Importantly, the Flow+Encryption configuration produced accuracy and F1-scores comparable to the Total approach while attaining the highest coverage, indicating that high-quality pseudo-labels can be generated efficiently using a smaller number of features

keyword: semi-supervised learning, pseudo-labeling, encrypted traffic, network security, feature extraction.

1 Introduction

With the increasing emphasis on privacy protection, the proportion of encrypted network traffic has continued to rise worldwide. According to SonicWall's 2024 Cyber Threat Report, cyberattacks leveraging encrypted communications in the Asia-Pacific region surged by 117% compared with the previous year [1]. As encrypted traffic becomes dominant, the limitations of traditional payload-based analysis methods have become increasingly evident. In response, machine learning–based traffic analysis has gained significant attention; however, these approaches require extensive labeled datasets, which are often costly and difficult to obtain.

* Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. 26, December 16-18, 2025, Sapporo, Japan. © The copyright of this paper remains with the author(s).

† Corresponding author

To address the scarcity of labeled data, semi-supervised learning techniques, particularly pseudo-labeling, have been proposed to leverage both labeled and unlabeled data during training. Despite extensive research on algorithmic performance, little attention has been paid to how feature extraction strategies influence the reliability and quality of pseudo-labels. To fill this gap, the present study performs a quantitative analysis of pseudo-label quality under various traffic feature extraction schemes, thereby elucidating the relationship between feature design and pseudo-label performance.

The key contributions of this study are summarized as follows:

1. It presents an empirical investigation into how different data extraction strategies affect the quality of pseudo-labels in encrypted traffic analysis.
2. It characterizes the general performance trends of multiple pseudo-labeling techniques under varying extraction configurations.
3. It proposes a set of effective data extraction strategies that balance detection accuracy, privacy sensitivity, and computational resource constraints.

The remainder of this paper is organized as follows. Section 2 reviews theoretical background and related work. Section 3 details the experimental design and methodology. Section 4 presents the results and comparative analysis. Section 5 discusses and interprets the key findings. Finally, Section 6 concludes the study and outlines directions for future research.

2 Background and Related Work

2.1 Overview of Semi-Supervised Learning and Pseudo-labeling

Semi-supervised learning (SSL) refers to a learning paradigm that enhances model generalization by jointly utilizing labeled and unlabeled data [2]. Van Engelen *et al.* (2020) categorized SSL methods into four types based on how unlabeled data are utilized: wrapper-based, inductive, intrinsically semi-supervised, and transductive approaches [3]. Among these, pseudo-labeling, which belongs to the wrapper-based category, is one of the most representative SSL approaches. It assigns the model's predictions on unlabeled samples as temporary labels (pseudo-labels) and retrains the model together with labeled data.

Patrick Kage *et al.* (2024) classified pseudo-labeling methods into confidence-based, augmentation-based, and iterative learning categories [4], while Mahmood *et al.* (2023) identified their proposed method as a similarity-based pseudo-labeling approach [5]. Building on these prior classifications, this study redefines pseudo-labeling techniques into five categories according to the primary design factor for improving final classification performance: confidence-based, consistency-based, similarity-based, iterative-based, and hybrid methods.

All of these methods incorporate consistency regularization to some extent and use a confidence threshold to determine the final pseudo-labels. However, each category focuses on a different mechanism to enhance model performance, as summarized below.

- **Confidence-based methods** focus on dynamically adjusting the confidence threshold for pse udo-label assignment. A representative example is FlexMatch (Bowen Z. et al., 2021), which applies class-wise adaptive thresholds to mitigate label distribution imbalance [6].
- **Consistency-based methods** encourage prediction consistency when applying different augm entations to the same input. The UDA (Unsupervised Data Augmentation) method (Qizhe X. *et al.*, 2019) applies both *weak* augmentations (e.g., noise injection, color shifting) and *strong* augmentations (e.g., back translation, flipping), minimizing the discrepancy between the two predictions through a consistency loss, thereby improving generalization [7].
- **Similarity-based methods** assign pseudo-labels based on feature-level similarity between lab

eled and unlabeled samples. ProtoMatch (Ziyu C. *et al.*, 2023) calculates class-wise prototype s by averaging feature representations of labeled samples and compares them with the feature representations of unlabeled data to determine pseudo-labels [8].

- **Iterative-based methods** improve model performance by repeatedly generating pseudo-label s and updating the model through a teacher–student framework. The MPL (Meta Pseudo Labe ls) method (Hieu P. *et al.*, 2020) uses feedback from the student model—specifically its traini ng loss—to refine the teacher model's parameters iteratively, thus enhancing final performanc e [9].
- **Hybrid methods** integrate multiple pseudo-labeling principles to achieve complementary effe cts. ReMixMatch (David B. *et al.*, 2019), for instance, combines confidence filtering with con sistency regularization, jointly improving pseudo-label quality and classification performance. [10]

2.2 Network Traffic Extraction Units

Network traffic can be represented and analyzed at different granularities, most commonly at the packet and flow levels. Papadogiannaki *et al.* (2021) emphasized that both packet-level and flow-level representations serve as fundamental extraction units that remain effective even under encryption [11]. In packet-level extraction, features such as packet size, direction, and inter-arrival time are directly derived from each individual packet. In contrast, flow-level extraction aggregates multiple packets belonging to the same communication flow to compute statistical descriptors such as average length, duration, and variance. Building upon these conventional approaches, this study incorporates cryptographic metadata extraction, enabling the analysis of encrypted traffic through attributes derived from cryptographic protocols. The proposed framework evaluates and compares pseudo-label quality across all combinations of these extraction strategies.

2.3 Related Work

The performance of intrusion detection models has been shown to vary considerably depending on the data extraction unit employed. Yang et al. (2022) compared packet- and flow-level representations within a reinforcement learning–based intrusion detection framework and confirmed that the choice of extraction granularity significantly influences model outcomes [12].

Despite such evidence, most pseudo-labeling–based approaches for encrypted traffic analysis have focused on enhancing detection accuracy within a single extraction domain, without exploring how different feature representations affect pseudo-label reliability. At the flow level, Yuan et al. (2024) achieved approximately a 9–10% improvement in detection accuracy by leveraging statistical characteristics of aggregated flows [13]. Likewise, Lin et al. (2022) reported over 98% detection accuracy using temporal flow-based features [14]. Conversely, at the packet level, Iliyasu et al. (2019) demonstrated that their semi-supervised model attained 89% accuracy even with only 10% labeled data, outperforming CNN and MLP baselines by 4–12% [15].

While these studies collectively demonstrate that extraction granularity can substantially influence learning performance, none have systematically examined how extraction strategy affects the quality of pseudo-labels—a critical factor determining the final accuracy of semi-supervised frameworks. Accordingly, the present study conducts a comparative evaluation of pseudo-label quality across six feature extraction strategies, aiming to clarify the relationship between extraction-level design and pseudo-label generation reliability.

3 Materials and Methods

3.1 Experimental Design

This study was designed to evaluate the impact of feature extraction strategies on pseudo-label quality in encrypted network traffic classification. To achieve this, pseudo-labels generated under six different extraction methods were assessed using three key metrics: accuracy, F1-score, and coverage. The pseudo-labeling techniques were categorized into five groups—confidence-based, consistency-based, similarity-based, iterative-based, and hybrid—and representative algorithms from each category were selected for experimentation, namely FlexMatch, UDA, ProtoMatch, MPL, and ReMixMatch.

Two experiments were conducted to analyze the overall performance trends and to evaluate the generalizability of the results. In the first experiment, a dataset combining four attack types was constructed for each feature extraction strategy, and the pseudo-label quality was compared across five pseudo-labeling methods. To emphasize the effect of feature extraction, the results were primarily reported as averaged values, while individual method results were additionally referenced to confirm whether consistent trends were observed across different pseudo-labeling algorithms. In the second experiment, datasets were constructed separately for each attack type to determine whether the observed influence of extraction strategies remained consistent across varying attack behaviors. As in the first experiment, the analysis mainly focused on average performance metrics to isolate the impact of feature extraction methods from algorithm-specific variations.

The overall experimental procedure comprised three main stages:

1. **Data Preparation:** TLS packets were extracted from raw PCAP files and processed accordi ng to six feature extraction configurations—Packet, Flow, Encryption, Flow+Encryption, Pa cket+Encryption, and Total (Packet+Flow+Encryption).

2. **Pseudo-label Generation:** Pseudo-labels were generated for each extraction method using fi ve semi-supervised learning algorithms: FlexMatch, UDA, ProtoMatch, MPL, and ReMixMa tch.

3. **Quality Evaluation:** The generated pseudo-labels were evaluated in terms of accuracy, F1-s core, and coverage. To minimize bias arising from method-specific fluctuations, average val ues across the five pseudo-labeling methods were used for final comparison.

3.2 Dataset and Preprocessing

This study utilized the CICIoT-2023 dataset, released by the *Canadian Institute for Cybersecurity* in 2023 [16]. The dataset consists of normal traffic collected from IoT environments and 32 categories of attack traffic. Among them, four representative attack types—SQL Injection, Browser Hijacking, DoS-HTTP-Flood, and Dictionary Brute-Force—were selected for experimentation. This attack types correspond to major threats listed in the OWASP Top 10 (2021) and exhibit distinct differences in their mechanisms, traffic patterns, and temporal characteristics. Such diversity makes them suitable for analyzing how different combinations of packet-, flow-, and encryption-level features affect pseudo-labeling quality.

The experimental dataset was configured under fixed ratio conditions of labeled:unlabeled = 1:9 and benign:attack = 5:5, meaning that class imbalance within the attack samples was not separately addressed. The data composition for each experiment is summarized as follows:

– **Experiment 1 (combined-attacks):** 20,000 benign and 20,000 attack samples (5,000 each for SQL Injection, Browser Hijacking, DoS-HTTP-Flood, and Dictionary Brute-Force).

– **Experiment 2 (per-attack):** 5,000 benign and 5,000 attack samples for each attack type.

Preprocessing. The preprocessing procedure applied in this study is summarized in Table 1.

Values labeled as 'unknown' represent fields that were not extracted because they were unused in the corresponding packet. To distinguish between fields that existed but were assigned to a value of zero and fields that were entirely absent, all instances labeled as *'unknown'* were recorded as zero while introducing a separate feature flag to mark the original *'unknown'* state. For count-type features, however, missing fields were considered equivalent to zero occurrences; therefore, no additional flag feature was created.

Method	Feature name	Explain
'unknown' = 0	server_extensions_cnt, client_extensions_cnt	Replace 'unknown' values with 0
Add 'is_unknown' feature	server_ttl, client_ttl, server_extensions, client_extensions	Replace 'unknown' values with 0 and add an 'is_unknown' feature
Label encoding	IP ID, ciphersuite	Map identical category values to the same integer
Hex to dec	SSL/TLS version,	Convert hexadecimal values to decimal
Int to float	ACK_mean, SYN_mean, FIN_mean, PSH_mean	Convert integer values to float type

Table 1: Preprocessing Method

Selected Features. The final set of selected features is listed in Table 2.

Feature category	Feature name	Explain
Packet level	Server_ttl	Server→client direction TTL value
	Server_ttl_is_unknown	Server→client direction TTL value extracted as unknown
	Client_ttl	Client→server direction TTL value
	Client_ttl_is_unknown	Client→server direction TTL value extracted as unknown
	Ip_len	IP header and payload total length
	Ip_id	Identifier for distinguishing fragments when reassembling original IP datagrams
	Ip_checksum	Checksum
Flow level	packet_size	Average packet size
	IAT	Average packet interval time
	ack_mean	Average ACK flag value within the session
	syn_mean	Average SYN flag value within the session
	fin_mean	Average FIN flag value within the session
	psh_mean	Average PSH flag value within the session
	app_data_size	Average application data length
Encryption level	SSL/TLS_version	TLS encryption version value (hexadecimal)
	client_extensions	List of extensions included in the Client hello
	client_extensions_cnt	Values extracted as unknown from the list of extensions included in the Client hello
	server_extensions	Number of extensions included in the Client hello

3.3 Experimental Environment and Model Configuration

The experiments were conducted on Windows 11 using a WSL2 (Ubuntu 20.04) environment equipped with an NVIDIA RTX A6000 GPU. A one-dimensional convolutional neural network (1D-CNN) was adopted to effectively capture the sequential characteristics of network traffic data.

Model Architecture. Conv1D → BatchNorm → ReLU → Conv1d → BatchNorm → ReLU → AdaptiveAvgPool1d → Flatten → Fully Connected

Hyperparameters. The hyperparameters used in the experiments are summarized in Table 3.

Hyperparameter	Setting Value
Batch size	32
Iteration	5,000
Threshold	0.7
Optimization function	Adam
Loss function	Cross-entropy

Table 3: Hyperparameter Settings

Data Augmentation Techniques. The data augmentation techniques used in the experiments are summarized in Table 4.

Feature category	Feature name	Weak Augment	Strong Augment
Packet level	Server_ttl	Among three variables (server_ttl, ip_len, ip_checksum), one variable was randomly selected and a noise of ±1 was added.	Among the same three variables, one variable was randomly selected, and for approximately 30% of the batch size, a noise of ±10 was added.
	Ip_len		
Packet+Encryption level	Ip_checksum		
Flow level	packet_size	Among three variables (packet_size, IAT, tls_app_data_size), one variable was randomly selected and a noise of ±0.01–0.1 was added.	Among the same three variables, one variable was randomly selected, and for approximately 30% of the batch size, the value was replaced with 0.
Flow+Encryption level	IAT		
Flow+Packet+ Encryption level	app_data_size		
Encryption level	client_extensions_cnt	Among two variables (client_extensions_cnt, server_extensions_cnt), one variable was randomly selected and a noise of ±1 was added.	Among the same two variables, one variable was randomly selected, and for approximately 30% of the batch size, a noise of ±10 was added.
	server_extensions_cnt		

Table 4: Data Augmentation Techniques

3.4 Performance Evaluation Metrics

The quality of the generated pseudo-labels was evaluated using the following three metrics:

1. Accuracy: The ratio of generated pseudo-labels matching the actual labels (ground truth data), used to evaluate label correctness.

2. F1-score: The harmonic mean of precision and recall, adopted to check bias toward specific classes.

3. Coverage: The proportion of unlabeled data assigned to pseudo-labels above a threshold, used to assess labeling scope and usability.

4 Experimental Results

4.1 Overall Performance of Feature Extraction Methods

Feature Extraction Categories	Average Coverage (±95% CI)	Average Accuracy (±95% CI)	Average F1-score (±95% CI)
Flow level	90.05%(±0.114)	0.714(±0.019)	0.718(±0.011)
Packet level	82.94%(±0.294)	0.692(±0.081)	0.654(±0.084)
Encryption level	85.30%(±0.152)	0.676(±0.052)	0.642(±0.141)
Flow+Encryption level	92.33%(±0.080)	0.730(±0.040)	0.754(±0.041)

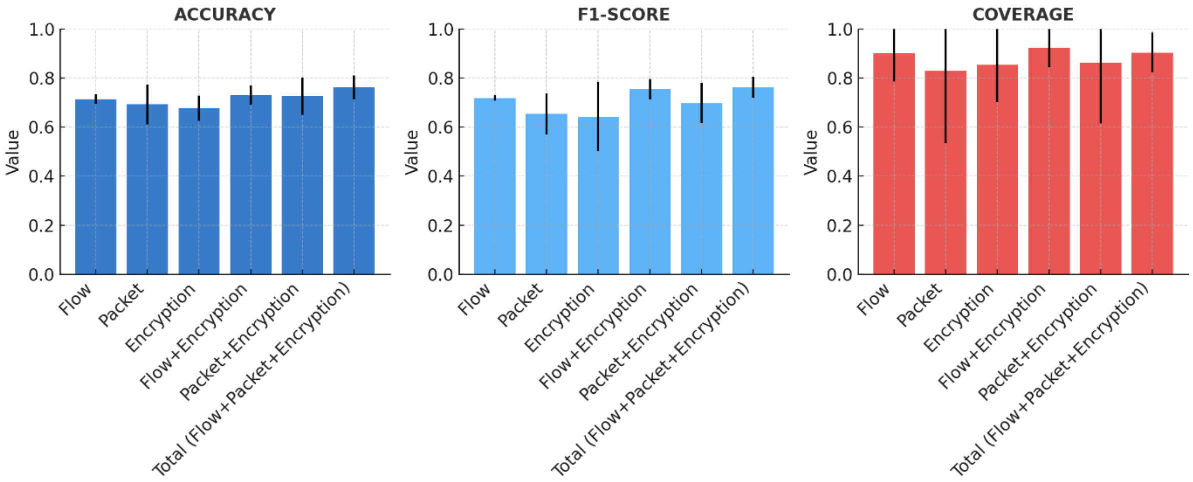

Figure 1: Pseudo-label quality (Accuracy, F1-score, Coverage) across extraction methods.

Table 5 presents the average performance of five pseudo-labeling methods across six feature extraction strategies. Figure 1 displays the Accuracy, F1-score, and Coverage results for each extraction strategy.

Total extraction method achieved the best results, maintaining over 90% coverage while attaining an accuracy of 0.76 and an F1-score of 0.76. This represents an improvement of approximately 12% in accuracy and 18% in F1-score compared to other strategies.

Excluding Total extraction method, Flow+Encryption achieved the highest performance. Notably, its coverage reached 92.33%, surpassing the holistic extraction method, while its accuracy and F1-score differed by only 1–4% compared to Total extraction results. This demonstrates that Flow+Encryption provided similar performance with fewer features.

4.2 Method-wise Performance of Feature Extraction Methods

Feature Extraction Categories	Pseudo-Labeling Method	Coverage	Accuracy	F1-score
Flow level	Confidence	91.94%	0.71	0.71
	Consistency	67.28%	0.75	0.73
	Similarity	97.88%	0.71	0.72
	Iterative	94.15%	0.71	0.73
	Hybrid	998.98%	0.69	0.70
Packet level	Confidence	92.57%	0.62	0.61
	Consistency	23.16%	0.72	0.59
	Similarity	100%	0.66	0.60
	Iterative	99.97%	0.84	0.82
	Hybrid	98.98%	0.62	0.65
Encryption level	Confidence	99.99%	0.64	0.67
	Consistency	60.45%	0.78	0.34
	Similarity	100%	0.66	0.72
	Iterative	97.06%	0.66	0.72
	Hybrid	98.98%	0.64	0.67
Flow+Encryption level	Confidence	93.07%	0.73	0.79
	Consistency	78.56%	0.76	0.74
	Similarity	99.85%	0.73	0.74
	Iterative	99.72%	0.73	0.74
	Hybrid	98.97%	0.64	0.66
Packet+Encryption level	Confidence	98.61%	0.62	0.62
	Consistency	35.85%	0.77	0.60
	Similarity	98.73%	0.69	0.72
	Iterative	99.89%	0.85	0.83
	Hybrid	98.97%	0.70	0.71
Total (Flow+Packet+Encryption)	Confidence	95.19%	0.73	0.73
	Consistency	78.12%	0.77	0.77
	Similarity	99.95%	0.73	0.75
	Iterative	99.26%	0.85	0.84
	Hybrid	98.98%	0.71	0.72

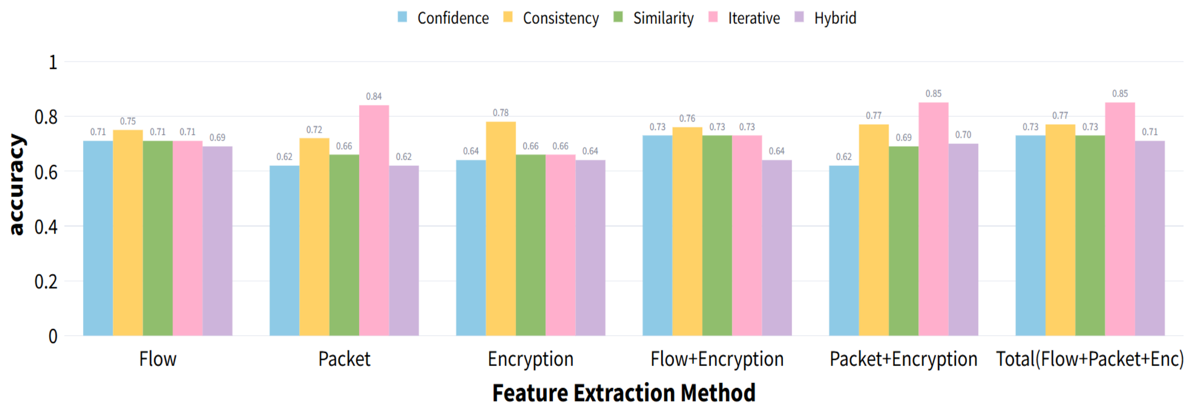

Figure 2: Accuracy of pseudo-labels across feature extraction methods by labeling technique

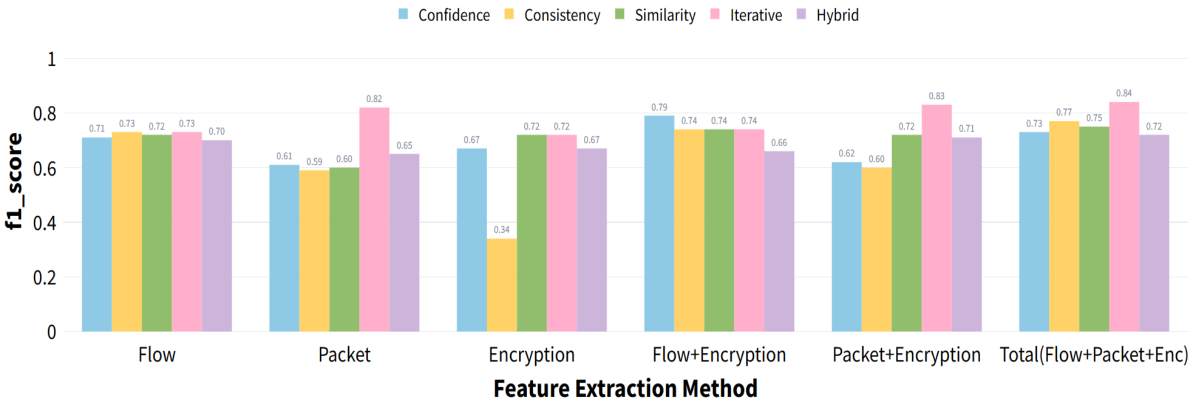

Figure 3: F1-score of pseudo-labels across feature extraction methods by labeling technique

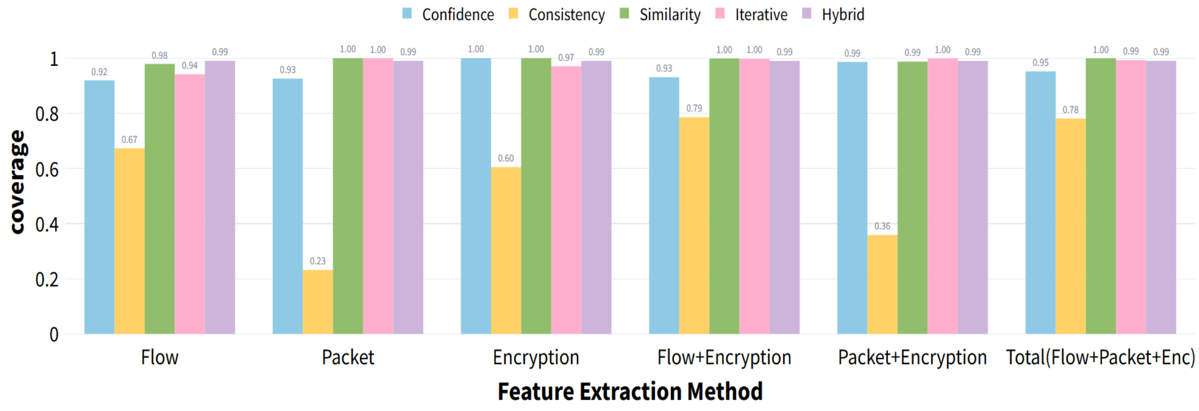

Figure 4: Coverage of pseudo-labels across feature extraction methods by labeling technique

While Section 4.1 presented the average results across the five pseudo-labeling methods, Section 4.2 reports the detailed performance of each method. Table 6 summarizes the performance of pseudo labels generated by five pseudo labeling techniques based on their extraction methods. Figures 2, 3, and 4 visually represent the results summarized in the table. Consistent with the average results, the Total strategy achieved the highest pseudo-label quality, recording an accuracy of 0.85 and an F1-score of 0.84, which were up to 10% higher in accuracy and 12% higher in F1-score compared to the other strategies. The Flow+Encryption strategy followed, showing a narrow margin of 1–3% from Total and thus demonstrating comparable performance. In the case of the Packet+Encryption method, performance varied across methods. In certain methods (Similarity, Iterative, and Hybrid), the results were again within a 1–3% margin of Total, confirming performance comparable to Flow+Encryption. Nevertheless, in the Consistency and Confidence methods, performance differences of approximately 15–22% were observed, indicating greater variability compared to Flow+Encryption.Attack-wise Performance of Feature Extraction Methods

4.3 Performance Analysis by Attack Type

Feature Extraction Categories	Attack Method	Coverage	Accuracy	F1-score
Flow level	Browser Hijacking	94.43%	0.71	0.73
	HTTP-Flood	89.59%	0.64	0.67
	Dictionary Brute Force	94.70%	0.71	0.73
	SQL Injection	96.55%	0.78	0.79
Packet level	Browser Hijacking	91.51%	0.69	0.71
	HTTP-Flood	89.40%	0.68	0.65
	Dictionary Brute Force	95.19%	0.73	0.73
	SQL Injection	91.70%	0.73	0.72
Encryption level	Browser Hijacking	88.18%	0.73	0.76
	HTTP-Flood	80.64%	0.61	0.65
	Dictionary Brute Force	86.46%	0.69	0.61
	SQL Injection	88.54%	0.74	0.73
Flow+Encryption level	Browser Hijacking	96.54%	0.76	0.77
	HTTP-Flood	85.96%	0.68	0.69
	Dictionary Brute Force	95.39%	0.73	0.73
	SQL Injection	99.46%	0.80	0.81
Packet+Encryption level	Browser Hijacking	94.21%	0.77	0.77
	HTTP-Flood	89.48%	0.74	0.73
	Dictionary Brute Force	95.44%	0.77	0.77
	SQL Injection	93.84%	0.77	0.77

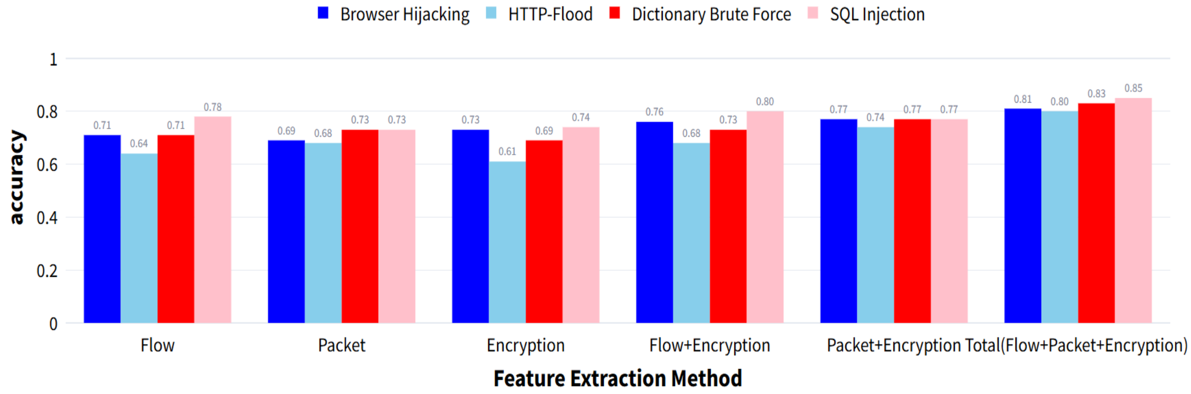

Figure 5: Accuracy of pseudo-labels across feature extraction methods by Attack technique

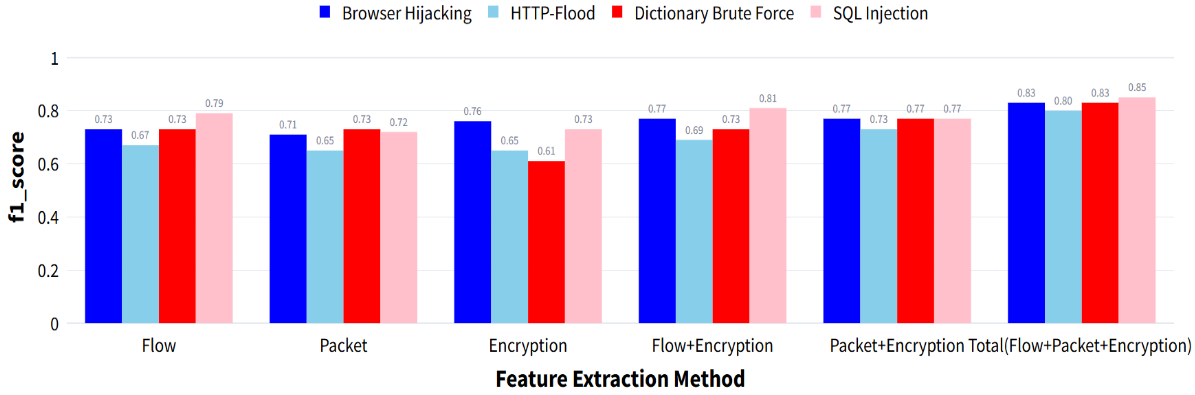

Figure 6: F1-score of pseudo-labels across feature extraction methods by Attack technique

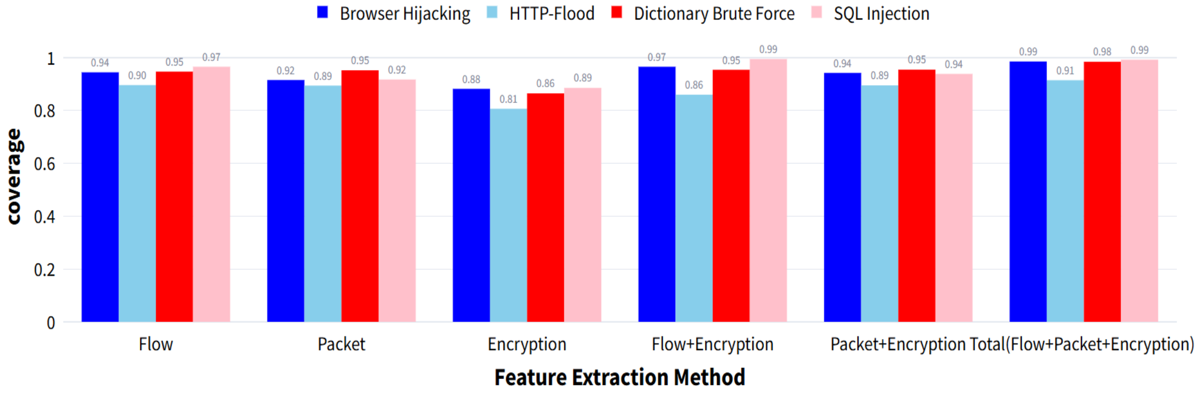

Figure 7: Coverage of pseudo-labels across feature extraction methods by Attack technique

Section 4.3 presents experiments conducted for each attack type to verify whether the overall trend identified in the combined analysis also appears consistently at the individual attack level. Table 7 summarizes the pseudo-label quality according to the extraction method for each attack type, while Figures 5, 6, and 7 visualize these results. Across all attack types, the Total extraction method consistently achieved the highest performance, with an accuracy of 0.81 and an F1-score of 0.83, confirming the same trend observed in the integrated analysis. Similarly, the Flow + Encryption method demonstrated comparable performance, with only a 2–4% difference from the Total extraction results, reaffirming its robustness and stability across different attack scenarios.

4.4 Interpretation and Analysis of Results

Experimental findings indicate that the Flow + Encryption extraction strategy yielded the most efficient pseudo-label generation, achieving high quality with fewer features. Although the Total (Flow + Packet + Encryption) method recorded the highest overall performance, the Flow + Encryption approach showed only a marginal difference of 1–3% in accuracy and F1-score. This result can be attributed to the complementary characteristics of flow and encryption features: flow features aggregate statistical patterns over packets sharing the same five-tuple, enabling consistent temporal learning, while encryption features provide protocol-level information—such as cipher suites and TLS extension metadata—that enhance session discrimination. Together, these properties contribute to stable and reliable pseudo-label generation. Both the Flow and Flow + Encryption methods maintained F1-scores within the 0.70 range with minimal variation, supporting their robustness. In contrast, packet-level features exhibited high-frequency variability caused by external factors such as transmission delay and packet loss, which may have led to instability and potential overfitting during training. The Packet and Packet + Encryption extraction strategies showed wider fluctuations, with accuracy values ranging from 0.62 to 0.84 and F1-scores from 0.59 to 0.83.

5 Discussion

This study examined how different data extraction strategies affect pseudo-label quality. The preceding experimental results clearly demonstrate that the choice of extraction strategy has a substantial impact on the performance of pseudo-labeling. However, several limitations remain in this work.

First, this study was conducted using a single dataset (CICIoT-2023), a single model structure (1D-CNN), and fixed data ratios for labeled-to-unlabeled (1:9) and normal-to-attack (5:5) samples. To mitigate these constraints, we evaluated five attack types (four individual attacks and one combined scenario) and five pseudo-labeling methods (Confidence-, Consistency-, Similarity-, Iterative-, and Hybrid-based) to identify consistent trends independent of specific attacks or algorithms. Nevertheless, as all experiments were performed within the same dataset, model architecture, and ratio settings, the generalizability of the findings to other datasets or model types remains limited. Future research should validate the generality of these results under diverse configurations of labeled–unlabeled and normal–attack ratios, using heterogeneous datasets such as QUIC- or VPN-based encrypted traffic, and employing various model architectures such as LSTM and Transformer.

Second, the use of a static dataset prevented this study from accounting for concept drift, i.e., temporal variations in traffic distributions observed in real-world networks. As user behavior and attack characteristics may change between the training and inference phases, pseudo-label reliability could degrade over time. Therefore, future studies should investigate how temporal shifts in data distribution influence pseudo-label confidence and model robustness.

Third, this study evaluated six combinations of feature extraction methods based on packet-, flow-, and encryption-level information. Consequently, other possible extraction paradigms—such as burst-level, time-window, and connection-level feature generation—were not examined. Further work should explore these complementary approaches to assess their influence on pseudo-label quality.

Fourth, this research primarily focused on the quantitative comparison of pseudo-label quality across different feature extraction strategies. Although Section 4.4 provided an interpretation of the efficiency of the Flow + Encryption method and the instability of packet-level features, this interpretation was inferential and lacked quantitative validation. Additionally, some performance fluctuations were observed among the pseudo-labeling methods, particularly in the Consistency-based approach (UDA). However, since the primary objective of this study was to analyze the effect of data extraction methods on pseudo-label generation, a detailed investigation of the internal mechanisms and performance variations across individual pseudo-labeling methods was not within the study's scope. Future research should address this limitation through feature-importance analysis and augmentation-strength experiments to provide a more granular understanding of these behaviors.

6 Conclusion and Future Work

This study analyzed how the performance of semi-supervised pseudo-labeling techniques varies depending on data extraction strategies in encrypted network traffic. Six combinations of flow-, packet-, and encryption-level metadata were evaluated. Experimental results showed that the Total (Flow + Packet + Encryption) extraction method achieved the highest pseudo-label quality, while the Flow + Encryption combination produced comparable results with fewer features, demonstrating its practical applicability. These findings indicate that pseudo-labeling performance is influenced not only by the choice of algorithm but also by the granularity of data extraction.

Future research should address the following limitations of this study:

1. Generalization across datasets — The observed trends should be validated using diverse datase ts (e.g., QUIC-based traffic, VPN traffic) to ensure robustness in various encrypted environme nts.
2. Broader feature extraction perspectives — Pseudo-label quality should be evaluated from mult iple angles by introducing additional feature extraction approaches such as burst-level, time-wi ndow, connection-level, and application handshake–based methods.
3. Model generalizability — The generality of the findings should be verified by applying differe nt model architectures, including LSTM, Transformer, and Attention-based models.
4. Feature contribution and resource trade-offs — Future work should include SHAP-based featu re importance analysis to quantify the contribution of each feature group and evaluate computa tional complexity and GPU resource usage to identify trade-offs between performance and res ource efficiency.

In conclusion, this study provides a systematic analysis of how data extraction granularity affects pseudo-label quality in encrypted traffic environments. The findings offer valuable insights for selecting optimal feature extraction strategies in machine learning–based encrypted traffic analysis and contribute to the design of more efficient and scalable security analytics frameworks.

Acknowledgement. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2023-00235509, Development of security monitoring technology-based network behavior against encrypted cyber threats in ICT convergence environment).

References

[1] SonicWall. (2024). *2024 SonicWall Cyber Threat Report*. Retrieved from https://www.sonicwall.com/resources/white-papers/2024-sonicwall-cyber-threat-report.

[2] Chapelle, O., Schölkopf, B., & Zien, A. (Eds.). (2006). *Semi-Supervised Learning*. MIT Press. [Reviewed in IEEE Transactions on Neural Networks, 20(3), 542]. doi:10.1109/TNN.2009.2015974.

[3] van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning, 109*(3), 373–440. https://doi.org/10.1007/s10994-019-05855-6.

[4] Kage, P., Rothenberger, J. C., Andreadis, P., & Diochnos, D. I. (2024). A Review of Pseudo-Labeling for Computer Vision. *arXiv preprint.* https://doi.org/10.48550/arXiv.2408.07221.

[5] Mahmood, M. J., Raj, P., Agarwal, D., Kumari, S., & Sing, P. (2023). SPLAL: Similarity-based pseudo-labeling with alignment loss for semi-supervised medical image classification. *arXiv preprint*. https://arxiv.org/abs/2307.04610.

[6] Zhang, B., Wang, Y., Hou, W., Wu, H., Wang, J., Okumura, M., & Shinozaki, T. (2022). FlexMatch: Boosting semi-supervised learning with curriculum pseudo labeling. *arXiv preprint.* https://arxiv.org/abs/2110.

[7] Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., & Le, Q. V. (2020). Unsupervised data augmentation for consistency training. *arXiv preprint.* https://arxiv.org/abs/1904.12848.

[8] Cheng, Z., Wang, X., & Li, J. (2023). ProMatch: Semi-supervised learning with prototype consistency. *Mathematics, 11*(16), 3537. https://doi.org/10.3390/math11163537.

[9] Pham, H., Dai, Z., Xie, Q., Luong, M.-T., & Le, Q. V. (2021). Meta pseudo labels. *arXiv preprint.* Retrieved from https://arxiv.org/abs/2003.10580.

[10] Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., & Raffel, C. (2020). ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint*. Retrieved from https://arxiv.org/abs/1911.09785

[11] Papadogiannaki, E., & Ioannidis, S. (2021). A survey on encrypted network traffic analysis applications, techniques, and countermeasures. *ACM Computing Surveys, 54*(6), Article 123, 35 pages. https://doi.org/10.1145/3457904.

[12] Yang, B., Arshad, M. H., & Zhao, Q. (2022). Packet-level and flow-level network intrusion detection based on reinforcement learning and adversarial training. *Algorithms, 15*(12), 453. https://doi.org/10.3390/a15120453.

[13] Yuan, Y., Huang, Y., & Wang, J. (2025). AnomalyAID: Reliable interpretation for semi-supervised network anomaly detection. *arXiv preprint*. Retrieved from https://arxiv.org/abs/2411.11293.

[14] Lin, K., Xu, X., & Xiao, F. (2022). MFFusion: A multi-level features fusion model for malicious traffic detection based on deep learning. *Computer Networks, 202*, 108658. https://doi.org/10.1016/j.comnet.2021.108658.

[15] Iliyasu, A. S., & Deng, H. (2020). Semi-supervised encrypted traffic classification with deep convolutional generative adversarial networks. *IEEE Access, 8*, 118–126. https://doi.org/10.1109/ACCESS.2019.2962106.

[16] Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., & Ghorbani, A. A. (2023). CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environments. *Sensors* (submitted).