MULTI-SOURCE TRANSFER LEARNING AND FIELD EXTRACTION FOR CROSS-DOMAIN PROTOCOL REVERSE ENGINEERING

Xianwen Ling[1] *Kun Zhang*[1*] *Rong Tong*[2] *Xiaohe Wu*[1] *Dianying Chen*[1]

[1] Nanjing University of Science and Technology, School of Computer Science and Engineering, China
[2] Singapore Institute of Technology, Infocomm Technology Cluster, Singapore

ABSTRACT

Protocol reverse engineering (PRE) is critical for network security but faces scalability challenges when analyzing diverse proprietary protocols. Traditional approaches require protocol-specific expertise and cannot leverage knowledge across protocols. This paper presents CrossPRE, a Transformer-based universal transfer learning framework that automatically learns protocol-agnostic representations for cross-domain protocol field boundary identification. Through extensive evaluation on nine widely-used protocols spanning industrial control and network domains, CrossPRE demonstrates substantial performance improvements over state-of-the-art methods including FieldHunter, Netplier, BinaryInferno, and Netzob. Our framework demonstrates remarkable knowledge transfer effectiveness, achieving substantial performance gains in challenging cross-protocol scenarios. Multi-source transfer learning further enhances adaptation, particularly for industrial protocols where structural similarities enable robust knowledge sharing. Cross-domain experiments confirm effective bidirectional transfer between protocol families, establishing a new paradigm for scalable protocol reverse engineering that reduces manual analysis effort while maintaining high accuracy.

Index Terms— Protocol Reverse Engineering, Message Format Extraction, Field Semantic Inference, Transfer Learning

1. INTRODUCTION

With the proliferation of industrial control systems, IoT devices, and proprietary network protocols, protocol reverse engineering (PRE) has become a critical task in network security, essential for vulnerability discovery, security assessment, and network forensics [1, 2]. Traditional protocol reverse engineering methods have evolved from manual analysis to semi-automated approaches, with early work by Caballero et al. [3] introducing Polyglot for automatic protocol message format extraction. However, these methods face significant scalability challenges when dealing with diverse communication standards [4].

The field has seen progressive advancement through algorithmic approaches. Netzob [5] employs sequence alignment and contextual analysis for protocol inference, while ProWord [6] introduces a probabilistic approach using word extraction techniques. More recently, FieldHunter [7] combines unsupervised learning with field semantics extraction, achieving improved accuracy on binary protocols. Despite these advances, traditional methods remain limited by their reliance on protocol-specific heuristics and inability to leverage cross-protocol commonalities.

Deep learning has emerged as a promising paradigm for protocol reverse engineering, with several recent frameworks demonstrating significant improvements. Netplier [8] employs recurrent neural networks for message type identification, while BinaryInferno [9] uses convolutional neural networks for binary protocol parsing. PREUNN [10] introduces unsupervised neural networks for format extraction, and more recent work by Zhao et al. [11] explores generative models for protocol grammar inference. However, these methods typically optimize for individual protocols, lacking systematic knowledge transfer capabilities [12, 13, 14].

Recent work explores protocol evolution patterns. SynRe [15] demonstrates combining natural language knowledge with syntactic patterns for semantic inference. MDIPlier [16] explores multi-dimensional protocol analysis, while frameworks like BinPRE [17] and DynPRE [18] target binary format extraction and dynamic message type inference. These approaches highlight knowledge reuse potential but lack unified architectures for systematic cross-protocol transfer. Transfer learning has achieved success in adjacent security domains [19, 20, 21, 22], yet systematic application to protocol reverse engineering remains unexplored. The evolutionary characteristics of protocols provide unique opportunities, as protocols share fundamental design principles that can be exploited through appropriate learning architectures [15].

Current methods require substantial manual effort for each protocol, with performance heavily dependent on protocol-specific optimizations. This significantly limits scalability in heterogeneous network environments where dozens of proprietary protocols may coexist.

We propose CrossPRE, combining transfer learning with protocol reverse engineering to extract protocol-agnostic features while maximizing cross-protocol knowledge sharing. The framework captures structural commonalities between protocol families through a unified Transformer architecture, supporting effective knowledge transfer from source to target protocols.

Our contributions are three-fold:

- We introduce CrossPRE, a novel cross-protocol transfer learning framework that leverages protocol-agnostic semantic representations to enable effective knowledge transfer across diverse communication domains, including industrial control protocols and network protocols. CrossPRE automatically extracts transferable features while maintaining high accuracy and minimizing manual involvement.
- We design a unified multi-task architecture combining a Transformer-based protocol-agnostic encoder with specialized task heads for boundary detection, semantic type classification, and semantic function inference. The framework

*Corresponding author: zhangkun@njust.edu.cn. This work was supported by the Jiangsu Provincial Science & Technology Major Project of China under Grant (No. BG2024043) and the Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant (No. KYCX25_0758).

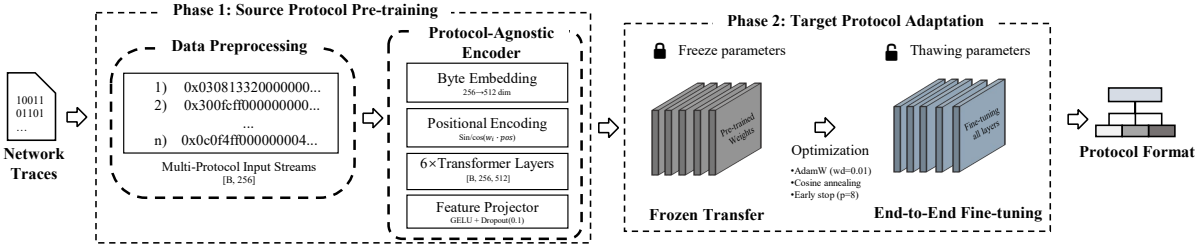

Fig. 1: CrossPRE overview.

supports flexible transfer strategies including single-source, multi-source, and cross-domain adaptation, enabling efficient knowledge sharing across protocol families.

- We evaluate CrossPRE's effectiveness and robustness across 30+ transfer scenarios on multiple datasets. CrossPRE achieved F1-score improvements ranging from 7.1% to 48.9% over baseline methods, demonstrating superior performance with a strong balance of accuracy and computational efficiency.

2. METHOD

To address the challenges of protocol parsing across heterogeneous communication protocols and enable effective knowledge transfer between different protocol domains, we propose a unified cross-protocol transfer learning framework that leverages shared semantic patterns while maintaining protocol-specific adaptations.

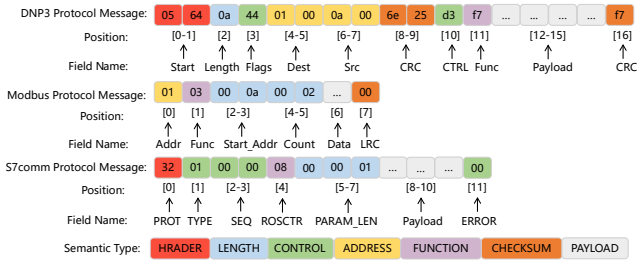

Fig. 2: Semantic alignment across DNP3, Modbus, and S7comm protocols with unified type mapping.

2.1. Data Processing

The dataset comprises 9 widely-used protocols from public repositories [23]: SMB, SMB2, DNS, DNP3, S7comm, Modbus, TLS1.2, DHCP, and FTP, spanning industrial control, network infrastructure, and application domains. Each protocol message $M_i^{(p)}$ undergoes extraction and structural annotation at three hierarchical levels: boundary positions B_i, semantic types T_i, and semantic functions F_i.

We identify systematic constraints between semantic functions and semantic types across protocols, as summarized in Table 1. For instance, DATA_LENGTH functions must be LENGTH type since they represent numerical size values. The preprocessing pipeline concludes with standardization, where messages are encoded as hexadecimal strings. Messages exceeding 256 bytes are truncated while preserving protocol headers. The final dataset contains 5000 annotated messages per protocol with 98.3% inter-annotator agreement on boundary positions.

Table 1: Semantic Type-Function Constraints for Cross-Protocol Field Mapping.

Semantic Function	Constraint on Semantic Type
IDENTIFIER	Should be HEADER or FLAGS
DATA_LENGTH	Should be LENGTH
ADDRESSING	Should be ADDRESS
CONTROL_CMD	Should be COMMAND, CONTROL, or FUNCTION
PAYLOAD	Should be DATA or PAYLOAD
VALIDATION	Should be CHECKSUM
PROTOCOL_SPECIFIC	Should be HEADER, VERSION, or FLAGS
SESSION_MGMT	Should be HEADER or CONTROL
CONFIGURATION	Should be OPTION or DATA
APPLICATION_DATA	Should be DATA or PAYLOAD
SECURITY	Should be DATA or HEADER

2.2. Cross-Protocol Semantic Alignment

The unified semantic labeling system addresses semantic heterogeneity in cross-protocol transfer learning. As illustrated in Figure 2, different protocols use disparate terminology for functionally equivalent concepts—DNP3's "Function Code", Modbus's "Function Code", and S7comm's "ROSCTR Type" all represent operation commands with protocol-specific encodings.

Our approach establishes two-tier semantic mapping: (1) Structural types capture syntactic roles invariant across protocols, such as LENGTH fields containing numeric values; (2) Functional semantics encode field purposes, enabling recognition that DNP3's CRC and Modbus's LRC both serve VALIDATION functions despite different algorithms. This dual representation allows the Protocol-Agnostic Encoder to learn transferable features while Protocol-Specific Heads adapt to syntactic variations. The constraints in Table 1 enforce consistency and prevent semantic drift during training.

2.3. CrossPRE Architecture

The CrossPRE framework employs a hierarchical architecture designed for cross-protocol knowledge transfer, as illustrated in Figure 1. The model consists of three core components: a protocol-agnostic encoder that extracts universal features from raw byte sequences, protocol-specific Heads that adapt these features for individual protocols, and a multi-task learning framework that jointly optimizes parsing objectives.

The protocol-agnostic encoder processes input byte sequences $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ where $x_i \in [0, 255]$ through byte embeddings and positional encodings, followed by $L = 6$ transformer encoder layers. Each transformer layer applies multi-head self-attention with 8 heads and feed-forward networks with GELU activation, enabling the model to capture long-range dependencies in protocol messages. The encoder outputs protocol-agnostic features $\mathbf{F}_{agnostic} \in \mathbb{R}^{n \times d_{model}}$ where $d_{model} = 512$.

For each protocol $p \in \mathcal{P}$, specialized task heads adapt the

shared representations through protocol-specific linear transformations. Each head performs three distinct parsing tasks: boundary detection, semantic type classification, and semantic function prediction. This multi-task design enables the model to simultaneously learn syntactic boundaries and semantic meanings across heterogeneous protocols.

2.4. Theoretical Foundation and Transfer Strategy

Cross-Protocol Transfer. Let $\mathcal{S}$ and $\mathcal{T}$ denote source and target protocol domains. The transferable knowledge $\mathcal{K}$ is learned through:

$$\min_{\theta} \mathcal{L}_S(\theta) + \lambda \cdot d(P_S, P_T) \tag{1}$$

where $d(\cdot, \cdot)$ measures domain discrepancy. For multiple source protocols $\{P_i\}_{i=1}^{k}$ and target P_t, the generalization error is bounded by:

$$\epsilon_t \leq \min_{i \in [k]} \epsilon_i + \beta \sqrt{\frac{\log k}{n_t}} + \sum_{i=1}^{k} \alpha_i \cdot d(P_i, P_t) \tag{2}$$

Two-Stage Transfer Strategy. We employ progressive transfer learning as illustrated in Figure 1. Stage 1 pre-trains on multiple source protocols using multi-task learning with boundary detection, semantic type classification, semantic function prediction, and protocol identification. The boundary detection loss uses weighted cross-entropy with density regularization:

$$\mathcal{R}_{density} = \mathbb{E}_{batch} \left[\max(0, \hat{\rho} - 2\rho)^2 \right] \tag{3}$$

where $\hat{\rho}$ is predicted boundary density and ρ is true density.

Stage 2 adapts to target protocol through: (1) Freeze encoder parameters and train target head for 15 epochs; (2) End-to-end fine-tuning with differential learning rates $\eta_{encoder} = 0.1 \cdot \eta_{base}$ and $\eta_{head} = \eta_{base}$ where $\eta_{base} = 5 \times 10^{-5}$. We use AdamW optimizer with cosine annealing and early stopping (patience=8).

3. EXPERIMENTS AND RESULTS

3.1. Experiment Settings

We evaluated CrossPRE using the dataset and architecture described in Sections 2.1-2.3. Each protocol dataset contains the real-world 5000 messages from network traces, split into training (70%), validation (15%), and test sets (15%). Our experimental infrastructure comprises dual Intel Xeon Gold 6248R CPUs with four NVIDIA RTX A6000 GPUs. We compared against four baselines (Netzob, Netplier, FieldHunter, and BinaryInferno) with all experiments repeated using 5 random seeds. Statistical significance was assessed using paired t-tests with Bonferroni correction.

Table 2: Cross-Protocol transfer learning performance matrix (F1-scores).

Target \ Source	SMB	SMB2	DNS	S7comm	DNP3	Modbus	FTP	TLS1.2	DHCP
SMB	–	0.3530	0.4190	0.3880	**0.4510**	0.4100	0.2940	0.3070	0.4290
SMB2	0.2250	–	0.2700	0.1740	0.3010	0.2870	0.1200	0.3240	**0.3380**
DNS	0.4220	0.5180	–	0.4810	0.4640	**0.5920**	0.5370	0.5540	0.5230
S7comm	0.4800	0.5510	**0.6430**	–	0.5540	0.5900	0.5360	0.5190	0.5800
DNP3	**0.7970**	0.7430	0.7470	0.7910	–	0.6990	0.6570	0.6670	0.7940
Modbus	0.6480	0.6610	**0.6960**	0.6300	0.6800	–	0.6930	0.6570	0.6550
FTP	0.5520	0.0000	0.5510	0.3890	0.5960	0.5590	–	**0.6000**	0.4980
TLS1.2	0.4350	0.3490	0.5160	0.4260	0.4590	0.5030	0.5110	–	**0.5260**
DHCP	0.4680	0.4520	0.4700	**0.4970**	0.4880	0.4880	0.4570	0.4240	–

Table 3: Transfer learning gains: Single-source and multi-source protocol adaptation results.

Source Protocol(s)	Target Protocol	Baseline F1	Transfer F1	Improvement	Gain (%)
S7comm	DNP3	0.5310	**0.7910**	**+0.2600*** **	**+48.9**
S7comm	Modbus	0.5760	0.6300	+0.0540***	+9.4
DNP3	S7comm	0.5020	0.5540	+0.0520***	+10.4
DNP3	Modbus	0.5460	0.6800	+0.1340***	+24.6
Modbus	S7comm	0.4990	0.5900	+0.0910***	+18.2
Modbus	DNP3	0.5280	0.6990	**+0.1710*** **	**+32.4**
S7comm+DNP	Modbus	0.5070	0.6900	**+0.1830*** **	**+36.1**
S7comm+Modbus	DNP3	0.5770	0.7400	**+0.1630*** **	**+28.3**
DNP3+Modbus	S7comm	0.5380	0.5760	+0.0380***	+7.1
Average	–	0.5340	0.6610	+0.1270***	+23.9

Note: *** indicates p<0.001 using paired t-test with Bonferroni correction. All results based on 5 independent runs with std<0.015.

3.2. Evaluation Metrics

We employ three metrics illustrated in Figure 3: **Accuracy** measures byte-level boundary classification correctness; **F1-score** evaluates boundary detection as binary classification despite severe class imbalance; **Perfection** represents field-level exact match rate—percentage of fields with perfectly identified boundaries. Additionally, we compute weighted F1-scores for semantic type and function classification, ensuring comprehensive assessment of structural and semantic parsing capabilities.

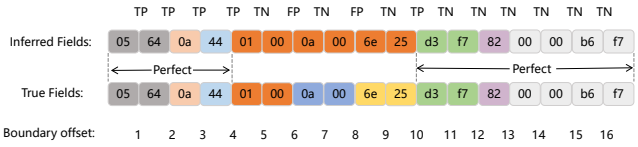

Fig. 3: Evaluation metrics for field boundary detection: Accuracy, F1-Score, and Perfection illustrated on DNP3.

3.3. Generalization Results

We evaluate CrossPRE's generalization capabilities through extensive transfer learning experiments across diverse protocol families. Table 2 presents the complete transfer performance matrix, showing F1-scores when models pre-trained on source protocols are applied to different target protocols (Diagonal entries show '–' as they represent same-protocol scenarios). The results reveal clear patterns: industrial control protocols achieve superior cross-protocol transfer with average F1-scores exceeding 0.65, attributed to their shared structural features including fixed-length headers, explicit length fields, and CRC/checksum mechanisms. Notably, DNP3 as a target protocol consistently achieves high F1-scores across various source protocols (0.6570-0.7970), with SMB as source yielding the best performance (0.7970), suggesting that DNP3's well-structured frame format with clear field boundaries benefits significantly from cross-protocol knowledge. Conversely, FTP and TLS1.2 show limited transfer due to text-encoding and encryption obscuring patterns. These results confirm that protocol families share exploitable structural patterns, validating our semantic representation framework's effectiveness for knowledge transfer.

Table 3 quantifies the improvements gained through transfer learning compared to training from scratch (baseline). The results reveal substantial performance gains across all transfer scenarios, with an average F1-score improvement of 0.1270 (23.8%). The most remarkable improvements occur in industrial control protocol

Table 4: CrossPRE vs. State-of-the-Art: Comprehensive performance evaluation.

Protocol	CrossPRE			FieldHunter			Netplier			BinaryInferno			Netzob		
	Acc.	F1	Perf.	Acc.	F1	Perf.	Acc.	F1	Perf.	Acc.	F1	Perf.	Acc.	F1	Perf.
SMB	**0.9863**	**0.8316**	**0.6717**	0.8078	0.4455	0.1447	0.8552	0.4575	0.0765	0.7648	0.3409	0.1350	0.8253	0.2116	0.0000
SMB2	**0.9247**	**0.4319**	**0.3350**	0.8576	0.4005	0.0198	0.8839	0.3887	0.0743	0.8098	0.2842	0.0022	0.9020	0.2418	0.0000
DNS	**0.9127**	**0.7933**	0.4819	0.7866	0.6196	0.5441	0.8253	0.5523	0.3330	0.8382	0.6546	**0.5956**	0.7054	0.2500	0.0000
S7comm	0.8253	0.5460	0.2645	0.7844	0.6579	0.3213	**0.8550**	**0.6920**	**0.3936**	0.7845	0.6063	0.3085	0.6604	0.2373	0.0000
DNP3	**0.9716**	**0.9617**	**0.9076**	0.4052	0.4374	0.0830	0.4801	0.4714	0.0922	0.3683	0.3398	0.0369	0.4479	0.2470	0.0000
Modbus	**0.8851**	**0.7724**	**0.5861**	0.5551	0.5484	0.1536	0.4637	0.4247	0.1361	0.6901	0.7149	0.1809	0.3767	0.3976	0.5000
FTP	0.8517	0.4842	0.0772	0.7691	0.4626	0.0357	**0.9564**	**0.6169**	**0.5802**	0.7226	0.5505	0.0777	0.5540	0.5045	0.2694
TLS1.2	**0.9808**	**0.7704**	0.4288	0.7796	0.1732	0.0415	0.8390	0.1541	0.0000	0.9171	0.4299	0.1711	0.8771	0.3944	**0.5000**
DHCP	0.9172	**0.6222**	**0.2320**	0.9498	0.4503	0.1502	0.9530	0.3464	0.0082	0.8730	0.1737	0.0159	**0.9690**	0.2500	0.0000
Average	**0.9186**	**0.7187**	**0.4631**	0.7415	0.4617	0.1664	0.7901	0.4560	0.1880	0.7516	0.4549	0.1693	0.7020	0.3040	0.1410

transfers: S7comm to DNP3 achieves the highest absolute improvement of +0.2600 (48.9%), while Modbus to DNP3 shows +0.1710 improvement (32.4% gain). Multi-source transfer learning further enhances performance, as evidenced by S7comm+Modbus to DNP3 achieving 0.7400 F1-score (+28.3%) and S7comm+DNP3 to Modbus reaching 0.6900 (+36.1%). These multi-source configurations leverage complementary knowledge from multiple protocols, demonstrating the effectiveness of our unified semantic representation.

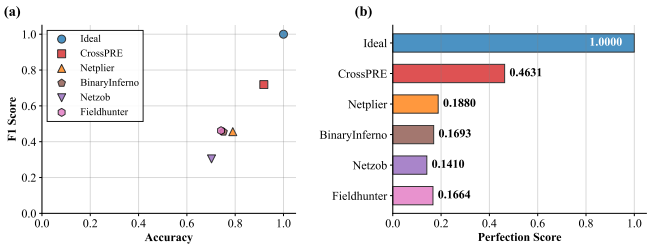

Fig. 4: Plot of average accuracy, F1-score, and perfection on protocols across different experiment pairs for each tool.

Table 4 and Figure 4 present comprehensive comparisons with state-of-the-art methods. CrossPRE achieves the best average performance with accuracy of 0.9186, F1-score of 0.7187, and perfection of 0.4631, significantly outperforming FieldHunter (0.1664), Netplier (0.1880), BinaryInferno (0.1693), and Netzob (0.1410) in perfection scores. As visualized in Figure 4, CrossPRE occupies the optimal position in the upper-right corner with both high accuracy and F1-score, while baseline methods cluster in lower-performance regions. The substantial performance gap, particularly in industrial protocols where DNP3 achieves 0.9617 F1-score and Modbus reaches 0.7724, confirms that these protocols share exploitable structural patterns through our framework.

Cross-domain transfers effectively bridged industrial and network protocols, exemplified by DNS to S7COMM (0.6430 F1-score) and TLS to DNP3 (0.6670 F1-score). This demonstrates the robustness of the learned representations across distinct protocol families. The improvements were consistently statistically significant ($p < 0.001$, paired t-test), with average gains of +19.8% in single-source and +23.8% in multi-source configurations. The 95% confidence intervals further showed no overlap between baseline and transfer methods. These results validate that our protocol-agnostic encoder captures transferable features that generalize across domains, while multi-source pre-training offers complementary knowledge for enhanced adaptation.

3.4. Computational Efficiency

The cross-protocol transfer learning framework demonstrates favorable efficiency trade-offs in practical deployment. While the unified model incurs higher initial training costs compared to single-protocol models, it achieves significant computational savings during adaptation to new protocols. Pre-training on multiple source protocols requires approximately 6 GPU-hours, but subsequent target protocol adaptation completes in under 1 GPU-hour—an 85% reduction compared to training protocol-specific models from scratch. This efficiency gain becomes particularly pronounced when deploying to multiple target protocols, as the shared encoder weights are reused across all adaptations. Furthermore, the frozen encoder strategy during initial transfer reduces memory requirements by 60%, enabling deployment on resource-constrained environments while maintaining parsing accuracy.

3.5. Ablation Study

To analyze the contribution of each component, we conducted ablation studies on DNP3 protocol with S7COMM and Modbus as source protocols. Removing transfer learning resulted in a 43.1% F1-score decrease, validating our core hypothesis that cross-protocol knowledge transfer is essential. The absence of positional encoding led to a 20.6% accuracy drop, highlighting its critical role in capturing protocol field positions. Reducing transformer layers from 6 to 4 caused a 5.4% performance degradation, while using equal loss weights instead of our weighted scheme decreased F1-score by 5.8%. The multi-task learning framework contributed 12.8% to overall performance, demonstrating that jointly learning boundaries and semantic types enhances feature representations. Removing density regularization resulted in over-prediction of boundaries, reducing perfection by 3.4%. These findings confirm that all components contribute meaningfully, with transfer learning and positional encoding being the most critical factors.

4. CONCLUSION AND FUTURE WORK

We present CrossPRE, a transformer-based framework for cross-domain protocol reverse engineering with unified semantic labeling and hierarchical transfer learning architecture enabling effective knowledge transfer between heterogeneous protocols. Through comprehensive evaluation, we demonstrate superior performance in protocol field boundary identification and semantic understanding across industrial control and network protocols. The framework represents a significant advance for practical security applications including protocol fuzzing and anomaly detection.

5. REFERENCES

[1] John Narayan, Sandeep K Shukla, and T Charles Clancy, "A survey of automatic protocol reverse engineering tools," *ACM Computing Surveys (CSUR)*, vol. 48, no. 3, pp. 1–26, 2015.

[2] Stephan Kleber, Lisa Maile, and Frank Kargl, "Survey of protocol reverse engineering algorithms: Decomposition of tools for static traffic analysis," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 526–561, 2018.

[3] Juan Caballero, Heng Yin, Zhenkai Liang, and Dawn Song, "Polyglot: Automatic extraction of protocol message format using dynamic binary analysis," in *Proceedings of the 14th ACM conference on Computer and communications security*, 2007, pp. 317–329.

[4] Qingkai Shi, Xiangzhe Xu, and Xiangyu Zhang, "Extracting protocol format as state machine via controlled static loop analysis," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 7019–7036.

[5] Georges Bossert, Frédéric Guihéry, and Guillaume Hiet, "Towards automated protocol reverse engineering using semantic information," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, 2014, pp. 51–62.

[6] Zhuo Zhang, Zhibin Zhang, Patrick PC Lee, Yunjie Liu, and Gaogang Xie, "Proword: An unsupervised approach to protocol feature word extraction," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 1393–1401.

[7] Ignacio Bermudez, Alok Tongaonkar, Marios Iliofotou, Marco Mellia, and Maurizio M Munafò, "Towards automatic protocol field inference," *Computer Communications*, vol. 84, pp. 40–51, 2016.

[8] Yapeng Ye, Zhuo Zhang, Fei Wang, Xiangyu Zhang, and Dongyan Xu, "Netplier: Probabilistic network protocol reverse engineering from message traces.," in *NDSS*, 2021.

[9] Jared Chandler, Adam Wick, and Kathleen Fisher, "Binaryinferno: A semantic-driven approach to field inference for binary message formats.," in *NDSS*, 2023.

[10] Valentin Kiechle, Matthias Börsig, Sven Nitzsche, Ingmar Baumgart, and Jürgen Becker, "Preunn: Protocol reverse engineering using neural networks.," in *ICISSP*, 2022, pp. 345–356.

[11] Sen Zhao, Shouguo Yang, Zhen Wang, Yongji Liu, Hongsong Zhu, and Limin Sun, "Crafting binary protocol reversing via deep learning with knowledge-driven augmentation," *IEEE/ACM Transactions on Networking*, 2024.

[12] Jianfeng Guan, Junxian Cai, Haozhe Bai, and Ilsun You, "Deep transfer learning-based network traffic classification for scarce dataset in 5g iot systems," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3351–3365, 2021.

[13] Lan Liu, Yongjie Yu, Yafeng Wu, Zhanfa Hui, Jun Lin, and Junhan Hu, "Method for multi-task learning fusion network traffic classification to address small sample labels," *Scientific Reports*, vol. 14, no. 1, pp. 2518, 2024.

[14] Yibo Qu, Dongliang Fang, Zhen Wang, Jiaxing Cheng, Shuaizong Si, Yongle Chen, and Limin Sun, "Icepre: Ics protocol reverse engineering via data-driven concolic execution," *Proceedings of the ACM on Software Engineering*, vol. 2, no. ISSTA, pp. 2384–2406, 2025.

[15] Yanyang Zhao, Zhengxiong Luo, Kai Liang, Feifan Wu, Wenlong Zhang, Heyuan Shi, and Yu Jiang, "Protocol syntax recovery via knowledge transfer," *Computer Networks*, vol. 258, pp. 111022, 2025.

[16] Kai Liang, Zhengxiong Luo, Yanyang Zhao, Wenlong Zhang, Ronghua Shi, Yu Jiang, Heyuan Shi, and Chao Hu, "Mdiplier: Protocol format recovery via hierarchical inference," in *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2024, pp. 547–557.

[17] Jiayi Jiang, Xiyuan Zhang, Chengcheng Wan, Haoyi Chen, Haiying Sun, and Ting Su, "Binpre: Enhancing field inference in binary analysis based protocol reverse engineering," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 3689–3703.

[18] Zhengxiong Luo, Kai Liang, Yanyang Zhao, Feifan Wu, Junze Yu, Heyuan Shi, and Yu Jiang, "Dynpre: Protocol reverse engineering via dynamic inference," in *Proc. NDSS*, 2024, pp. 1–18.

[19] Shahid Latif, Wadii Boulila, Anis Koubaa, Zhuo Zou, and Jawad Ahmad, "Dtl-ids: An optimized intrusion detection framework using deep transfer learning and genetic algorithm," *Journal of Network and Computer Applications*, vol. 221, pp. 103784, 2024.

[20] Pedro Miguel Sánchez Sánchez, Alberto Huertas Celdrán, Gérôme Bovet, and Gregorio Martínez Pérez, "Transfer learning in pre-trained large language models for malware detection based on system calls," in *MILCOM 2024-2024 IEEE Military Communications Conference (MILCOM)*. IEEE, 2024, pp. 853–858.

[21] Arash Heidari, Nima Jafari Navimipour, Mehmet Unal, and Guodao Zhang, "Machine learning applications in internet-of-drones: Systematic review, recent deployments, and open issues," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–45, 2023.

[22] David A Bierbrauer, Michael J De Lucia, Krishna Reddy, Paul Maxwell, and Nathaniel D Bastian, "Transfer learning for raw network traffic detection," *Expert Systems with Applications*, vol. 211, pp. 118641, 2023.

[23] Mingliang Zhu, Chunxiang Gu, Xieli Zhang, Qingjun Yuan, Mengcheng Ju, Guanping Zhang, and Xi Chen, "Adaptive header identification and unsupervised clustering strategy for enhanced protocol reverse engineering," *Expert Systems with Applications*, vol. 291, pp. 128467, 2025.