NetBench: A Large-Scale and Comprehensive Network Traffic Benchmark Dataset for Foundation Models

Chen Qian[1*], Xiaochang Li[1*], Qineng Wang[2], Gang Zhou[1], Huajie Shao[1]
[1] Department of Computer Science, William & Mary
[2] Independent Researcher
[1] {cqian03, xli59, gzhou, hshao}@wm.edu, [2] wangqineng73@gmail.com

***Abstract*—In computer networking, network traffic refers to the amount of data transmitted in the form of packets between internetworked computers or Cyber-Physical Systems. Monitoring and analyzing network traffic is crucial for ensuring the performance, security, and reliability of a network. However, a significant challenge in network traffic analysis is to process diverse data packets including both ciphertext and plaintext. While many methods have been adopted to analyze network traffic, they often rely on different datasets for performance evaluation. This inconsistency results in substantial manual data processing efforts and unfair comparisons. Moreover, some data processing methods may cause data leakage due to improper separation of training and testing data. To address these issues, we introduce the `NetBench`, a large-scale and comprehensive benchmark dataset for assessing machine learning models, especially foundation models, in both network traffic classification and generation tasks. `NetBench` is built upon seven publicly available datasets and encompasses a broad spectrum of 20 tasks, including 15 classification tasks and 5 generation tasks. Furthermore, we evaluate eight State-Of-The-Art (SOTA) classification models (including two foundation models) and two generative models using our benchmark. The results show that foundation models significantly outperform the traditional deep learning methods in traffic classification. We believe `NetBench` will facilitate fair comparisons among various approaches and advance the development of foundation models for network traffic. Our benchmark is available at https://github.com/WM-JayLab/NetBench.**

***Index Terms*—Benchmark Dataset, Network Traffic, Foundation Models**

I. Introduction

In the domain of computer networking, network traffic [1] is the amount of data transmitted in the form of packets between interconnected computers or systems. Generally, a network packet is composed of two parts: a header containing meta features and a commonly encrypted payload, as shown in Fig. 1. Analyzing network traffic is critical for enhancing network security and management. However, it is challenging to analyze network traffic due to the diversity of packet types, such as TCP and UDP, as well as the presence of both encrypted and unencrypted data.

Over the past decades, machine learning-based methods have been extensively developed for network traffic analysis. Earlier studies [8]–[12] primarily assessed the proposed methods on a single dataset encompassing at most 2 classification tasks. Despite achieving notable performance, these studies suffer from disparate data processing pipelines and insufficiently comprehensive evaluations. To address this issue, some recent works [13]–[15] have evaluated performance across multiple datasets and a broader range of classification tasks. For instance, the foundation model ET-BERT [14] was pre-trained on 7 datasets to learn intrinsic representations and then fine-tuned for 5 downstream tasks. However, the data processing techniques in existing works are often customized for themselves. Moreover, some studies randomly split the extracted packets from the same flow into training and testing dataset for packet-level evaluation, which may cause data leakage since the packets from same flow are strongly correlated. Additionally, certain research practices [16], [17] only focused on traffic generation for network simulation rather than traffic classification. In summary, there is a lack of evaluation against a unified and comprehensive network benchmark that encompasses both network traffic classification and generation, making it hard to fairly compare performance.

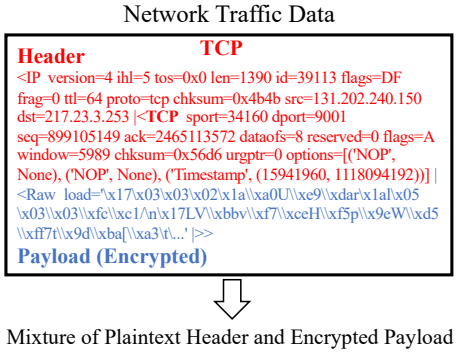

Fig. 1. The format of network traffic data, which are the mixture of plaintext header and encrypted payload. This mixture makes it hard to directly process with text tokenizers for model training.

To address existing limitations, we propose `NetBench`, a large-scale and comprehensive network traffic benchmark covering 7 distinct datasets, featuring 15 classification tasks and 5 generation tasks, as illustrated in Table I. For the classification benchmark, we construct 15 tasks across 7 datasets, including ISCXVPN 2016 [2], ISCXTor 2016 [3], USTC-TFC 2016 [4], Cross-Platform [5] dataset containing both Android

[*] Both authors contributed equally to this research.

979-8-3503-6345-6/24/$31.00 ©2024 IEEE
DOI 10.1109/FMSys62467.2024.00008

TABLE I
THE STATISTICAL INFORMATION OF 7 DIFFERENT DATASETS, INCLUDING 15 TRAFFIC CLASSIFICATION TASKS AND 5 TRAFFIC GENERATION TASKS.

Dataset	#Flow	#Packet	Task	#Label
ISCXVPN 2016 [2]	311,390	1,040,354	1 - VPN Detection 2 - VPN Service Detection 3 - VPN Application Classification	2 6 17
ISCXTor 2016 [3]	55,523	1,163,495	4 - Tor Detection 5 - Tor Service Detection	2 7
USTC-TFC 2016 [4]	489,139	4,564,519	6 - Malware Detection 7 - Application Classification	2 20
Cross Platform (Android) 2020 [5]	66,346	1,358,292	8 - Application Classification 9 - Country Detection	212 3
Cross Platform (iOS) 2020 [5]	34,912	971,762	10 - Application Classification 11 - Country Detection	196 3
CIRA-CIC-DoHBrw 2020 [6]	831,497	32,962,034	12 - DoH Attack Detection 13 - DoH Query Method Classification	2 5
CIC IoT Dataset 2023 [7]	1,163,495	27,738,736	14 - IoT Attack Detection 15 - IoT Attack Method Detection	2 7
Consecutive Packets from above 7 datasets	-	12,786,490	16 - Source IP Address Generation 17 - Destination IP Address Generation 18 - Source Port Number Generation 19 - Destination Port Number Generation 20 - Packet Length Generation	-
Total	2,952,302	69,799,192	-	-

and IOS applications, CIRA-CIC-DoHBrw 2020 [6], and CIC IoT Dataset 2023 [7]. The specific tasks for each dataset are outlined in table I. For the generation benchmark, we devise five tasks tailored to essential header fields [16] at the packet level, including IP addresses (Source/Destination), port number (Source/Destination) and packet length over the same 7 datasets above. This benchmark can help create valid network packets for network simulation.

To prevent data leakage, we first split the original network traffic data into training, validation, and testing sets, from which we extract flows and packets. Then, we anonymize sensitive header fields and employ a unified hexadecimal encoding to standardize different data formats. Moreover, our benchmark provides both flow-level and packet-level evaluations for each dataset, accommodating various input types. To the best of our knowledge, `NetBench` is the first network traffic benchmark that covers a wide range of tasks through a unified data processing method. Furthermore, we evaluate eight SOTA models (including two foundation models, ET-BERT [14] and YaTC [15]) on 15 classification tasks and two generative models on 5 generation tasks using our benchmark. We believe that our benchmark has laid a solid foundation for evaluating network traffic models fairly, which will significantly contribute to the development of foundation models for network traffic.

In summary, the main contributions of this work are threefold:

- **Comprehensive Network Traffic Benchmark.** We first create a large-scale and comprehensive network traffic benchmark from 7 distinct datasets with 20 tasks. Our benchmark offers diverse data types for foundation model training and a wide range of downstream tasks for fair evaluation in both network traffic classification and generation.
- **Unified Data Processing.** We introduce a uniform hexadecimal encoding method to process diverse network traffic data with anonymization in sensitive header fields. Our data processing method unifies data pre-processing, data standardization, and data segmentation, which can offer a fair evaluation opportunity and save human labor.
- **Benchmarking SOTA Methods.** We compare the performance of eight SOTA classifiers (including two foundation models) and two generative models using our benchmark. The evaluation results demonstrate the superiority of foundation models over other approaches in traffic classification.

II. RELATED WORK

Evaluation on Single Dataset. Most earlier machine learning-based studies have been evaluated using a single dataset focused on no more than two tasks. For example, Fs-net [8] assessed its performance on a campus network dataset [9] with 18 classes by extracting statistical features like packet length as model input. Similarly, BiLSTM_ATTN [10] implemented traffic segmentation and unwanted information removal for data processing before evaluation on the ISCXVPN-detection dataset [2]. DataNet [11] employed balanced subsets, byte vectorization, and normalization in its preprocessing steps, but its testing was confined to the ISCX VPN-application task. In addition, DeepPacket [12] built upon earlier efforts [10], [11] to eliminate irrelevant packets for evaluation on the ISCXVPN dataset. STAN [17] applied normalization and one-hot encoding for pre-processing before it was evaluated on a selected subset from UGR'16 [18] dataset across 2 generation tasks. While these approaches have shown promising results, their evaluations were restricted to single dataset and a limited range of tasks. Moreover, the variance in their data processing techniques could result in biased comparisons.

TABLE II
COMPARISON OF NETWORK TRAFFIC METHODS IN TERMS OF DATASET UTILIZATION AND TASK DIVERSITY. OUR BENCHMARK ENCOMPASSES A TOTAL OF 20 TASKS ACROSS 7 DATASETS.

Method	#Employed Dataset(s)	#Dataset Size (flows)	#Covered Task(s)
Fs-net	1	956K	1
BiLSTM_ATTN	1	47K	2
Datanet	1	206K	1
DeepPacket	1	206K	2
STAN	1	348K	2
TSCRNN	3	420K	3
YaTC	5	360K	4
ET-Bert	**7**	376K	5
NetShare	6	600K	13
NetBench	**7**	**2952K**	**20**

Evaluation on Multiple datasets. To achieve a more comprehensive evaluation, recent works have assessed their models on more datasets covering a variety of tasks. For instance, TSCRNN [13] was evaluated on 3 public datasets following the pre-processing pipelines established in [10], [11]. More recently, Transformer-based foundation models like ET-BERT [14] pre-trained on 6 public and 1 collected dataset to learn network traffic representations and then fine-tuned for 5 classification tasks. However, the construction of their packet-level evaluation datasets through random sampling raises the potential issue of data leakage, as highly correlated packets from the same flow could appear in both the training and testing datasets. Additionally, the foundation model YaTC [15] pre-trained on 4 datasets for better representation and fine-tuned on 5 public datasets to test its generalization ability. Nevertheless, it faces a similar risk of data leakage for packet-level evaluation due to its use of a data processing methodology akin to ET-BERT. Meanwhile, NetShare [16] evaluated its generative performance across six datasets, although it is not designed for traffic classification tasks.

In summary, while current approaches have been assessed across various datasets and tasks, the comprehensiveness and fairness of these evaluations remain questionable due to the variance in data processing approaches. To our best knowledge, there has yet to be an establishment of a large-scale and comprehensive benchmark that enables a fair comparison across different models.

III. NETBENCH

In this section, we create a large-scale and comprehensive benchmark dataset named NetBench, designed to standardize the evaluation of network traffic analysis models. Our benchmark contains 7 datasets with 20 tasks, including 15 classification tasks and 5 generation tasks.

A. Data Preparation

To create a large-scale and comprehensive benchmark, we first collect raw data samples from 7 publicly available datasets, as detailed in table I. Then, our next step is to convert them into a standardized format as model input. Fig. 2 shows the proposed pipeline of data preparation, which consists of three steps: Data Pre-Processing, Data Standardization, and Data Segmentation.

- **Data Pre-Processing.** Network traffic data encompasses flows of network packets containing privacy-sensitive information. As network traffic packets are captured and saved in Packet Capture (PCAP) files, we firstly segments PCAP files into distinct set for training, validation, and testing. This strategy ensures that high-correlated packets from the same flow will not be simultaneously present in both training and testing data. This effectively reduces the risk of data leakage. Subsequently, we extract flows containing multiple packets from these PCAP files according to distinct combinations of IP addresses, port numbers and protocols [19]. To protect data privacy, we further anonymize each packet by masking the source/destination IP addresses and port numbers (replaced with 0). This not only preserves data integrity but also protects data privacy.
- **Data Standardization.** Since data packets include both plaintext and ciphertext, we need to convert anonymized flows into a hexadecimal format, thereby unifying both header and payloads' formats and reducing the complexity to process network traffic data. This standardization offers a fair and uniform basis for model comparison, benchmarking more accurately among different tasks in network traffic classification and generation. Following the same methodology in [14], we employ the WordPiece algorithm to segment the hexadecimal data into 4-digit blocks, incorporating specific symbols (</s> for sequence ends, <head> for header separation, and <pkt> for packet demarcation) to standardize the dataset format.
- **Data Segmentation.** To capture the underlying characteristics of network traffic, we offer evaluation on each dataset at both flow level and packet level. Flow-level data is ideal for high-level analysis, allowing for the identification of trends, such as increasing or decreasing volumes of data over time, shifts in the types of traffic, and overall patterns of network use. Conversely, packet-level data enables a detailed examination of packet content and specific network traffic patterns. In addition to classification, we extract IP addresses, port numbers, and packet length for packet-level data as additional labels for generation tasks. By enabling evaluation at both levels, we can comprehensively evaluate the performance of different models in both classification and generation tasks.

B. Tasks in NetBench

Utilizing the described pipeline on seven collected traffic datasets, we construct a benchmark dataset for network traffic called NetBench. This dataset encompasses a total of 20 downstream tasks, including 15 traffic classification and 5 traffic generation tasks. Table I provides a detailed summary of NetBench. To our best knowledge, NetBench is the largest and most comprehensive benchmark in network traffic domain. It offers significant benefits for the development

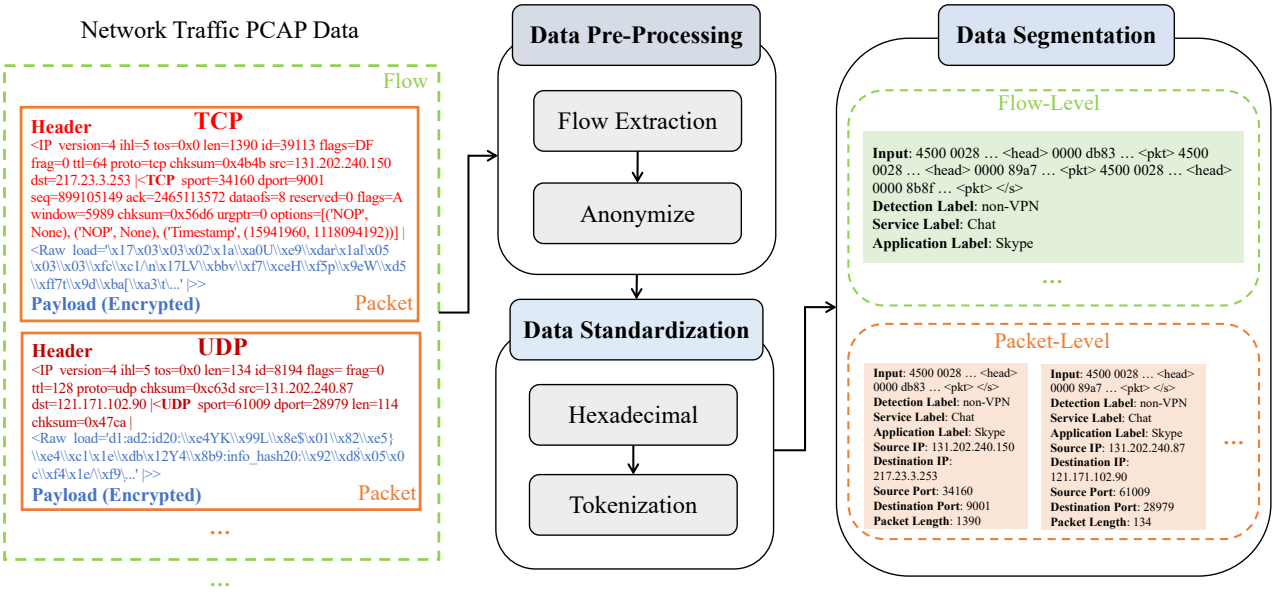

Fig. 2. The overall pipeline of data preparation, consisting of three parts: Data Pre-Processing, Data Standardization, and Data Segmentation. Firstly, we convert flows extracted from network traffic PCAP files into a hexadecimal format. Then, the WordPiece algorithm is employed to segment the hexadecimal data into 4-digit blocks, incorporating specific symbols (`</s>` for sequence ends, `<head>` for header separation, and `<pkt>` for packet demarcation). Lastly, we create two different types of dataset based on flow level and packet level.

of foundational models and enables fair assessment of their performance.

To better understand different tasks, we give two representative examples about traffic classification and generation as follows.

Classification Tasks: We utilize the ISCXVPN 2016 [2] dataset as an illustrative example. This dataset showcases the emulation of real-world internet behaviors by setting up user accounts to interact with each other on popular applications in VPN-routed or non-VPN scenarios. It involves capturing both VPN-routed and non-VPN internet sessions. There are three tasks: VPN Detection (Task 1) aims to ascertain the use of VPNs; VPN Service Detection (Task 2) seeks to identify 6 specific services, namely P2P, Streaming, Email, Chat, VoIP, and File Transfer; and VPN Application Detection (Task 3) strives to differentiate 17 distinct applications, including Facebook, Skype, YouTube, and more. Similarly, the remaining 12 tasks from other 6 datasets perform different types of classifications.

Generation Tasks: The goal of this task is to generate five pivotal header fields: the source IP address (Task 16), destination IP address (Task 17), source port (Task 18), destination port (Task 19), and the packet length (Task 20) [16]. This generation process entails the synthesis of 5 header fields to facilitate the creation of network traffic packets that closely mirrors real-world scenarios. Such generations are crucial to the evaluation of innovative networking hardware and software solutions [16], [20].

IV. Experiments

In this section, we perform a fair and holistic evaluation of SOTA models on the `NetBench` in both network traffic classification and generation tasks.

A. Experimental Settings

SOTA Models. We assess the performance of 8 major SOTA open-source models on `NetBench` to ensure a comprehensive and unbiased comparison. Specifically, they include 7 classification models, DataNet [11], Fs-net [8], BiLSTM_ATTN [10], DeepPacket [12], TSCRNN [13], ET-BERT [14], YaTC [15], as well as two competitive generative models, STAN [17] and Netshare [16]. Note that, ET-BERT and YaTC are foundation models which could be pre-trained and then fine-tuned on different classification tasks. The detailed experimental settings of each model are described below. For Fs-net, we follow its setting that only evaluate its performance in classification tasks at flow-level. For DataNet, DeepPacket, TSCRNN, and BiLSTM_ATTN, we assess their performance on all classification tasks at both flow level and packet level. Regarding ET-Bert and YaTC, we employ the released pre-trained weights and fine-tune them with the entirety of prepared training dataset. For generative models, we following the same setting in prior work [16] to randomly sample consecutive packets from each dataset, ensuring a consistent evaluation for future studies.

Dataset. We split our benchmark into training, validation, and testing data with a ratio of 8:1:1 respectively.

Evaluation metrics. For assessing SOTA models in traffic understanding tasks, we employ two principal metrics: Accuracy (AC) and Macro F1 Score (F1). To evaluate the traffic generation performance, we utilize the Jensen-Shannon Divergence (JSD) and Total Variation Distance (TVD). JSD quantifies the similarity of two probability distributions, indicating their shared information, while TVD measures the largest difference between two probability distributions, capturing the maximum discrepancy.

TABLE III
COMPARISON RESULTS OF TRAFFIC CLASSIFICATION TASKS FROM TASKS 1 TO 8 (FLOW: FLOW-LEVEL, PKT: PACKET-LEVEL, AC: ACCURACY, F1: F1-SCORE). FOUNDATION MODELS LIKE ET-BERT AND YATC OUTPERFORM OTHER TRADITIONAL DEEP LEARNING METHODS.

Method	Task 1		Task 2		Task 3		Task 4		Task 5		Task 6		Task 7		Task 8	
	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1
Datanet (flow)	0.9406	0.4847	0.6918	0.1363	0.3397	0.0298	0.9987	0.4997	0.4981	0.0950	0.6374	0.3893	0.2096	0.0173	0.0945	0.0153
Datanet (pkt)	0.8836	0.4691	0.5993	0.1249	0.3198	0.0285	0.9961	0.4990	0.3196	0.0692	0.6440	0.3917	0.0817	0.0076	0.0207	0.0002
Fs-net (flow)	0.9258	0.4807	0.2930	0.3367	0.2109	0.3088	0.9976	0.8327	0.8203	0.6354	0.3711	0.2801	0.8203	0.8455	0.0708	0.0411
BiLSTM_ATTN (flow)	0.9406	0.4847	0.0057	0.0313	0.3373	0.0484	0.9996	0.9166	0.8833	0.5554	0.6374	0.3893	0.0256	0.0052	0.0045	0.0004
BiLSTM_ATTN (pkt)	0.8836	0.4691	0.0303	0.0098	0.3198	0.0285	0.9961	0.4990	0.0722	0.0192	0.6440	0.3917	0.0329	0.0032	0.0003	0.0000
DeepPacket (flow)	0.9408	0.4948	0.6918	0.1363	0.3397	0.0298	0.9998	0.9666	0.4981	0.0950	0.6374	0.3893	0.3110	0.0908	0.0484	0.0004
DeepPacket (pkt)	0.8836	0.4691	0.5993	0.1249	0.3198	0.0285	0.0039	0.0039	0.4727	0.0917	0.6440	0.3917	0.2638	0.0209	0.0238	0.0002
TSCRNN (flow)	0.9406	0.4847	0.6918	0.1363	0.3397	0.0298	0.9987	0.7664	0.4470	0.1340	0.6374	0.3893	0.0839	0.0108	0.0243	0.0024
TSCRNN (pkt)	0.8836	0.4691	0.0303	0.0098	0.3198	0.0285	0.9961	0.4990	0.0722	0.0192	0.6440	0.3917	0.0329	0.0032	0.0037	0.0000
ET-BERT (flow)	0.9964	0.9838	0.7462	0.7110	0.5253	0.6667	**1.0000**	**1.0000**	0.9571	0.8029	**1.0000**	**1.0000**	0.9786	0.9820	0.8463	0.6770
ET-BERT (pkt)	0.9902	0.9758	0.7653	0.7631	0.5972	0.6956	**1.0000**	0.9978	0.8469	0.5685	**1.0000**	**1.0000**	0.9539	0.9530	**0.9687**	**0.8753**
YaTC (flow)	0.9974	0.9880	0.7805	0.7083	0.5991	0.7090	**1.0000**	**1.0000**	**0.9739**	**0.8512**	**1.0000**	**1.0000**	**0.9936**	**0.9949**	0.9161	0.8228
YaTC (pkt)	**0.9984**	**0.9961**	**0.8073**	**0.8260**	**0.6458**	**0.7837**	0.9998	0.9896	0.9601	0.8297	**1.0000**	**1.0000**	0.9850	0.9874	0.9519	0.8462

TABLE IV
COMPARISON RESULTS OF TRAFFIC CLASSIFICATION TASKS FROM TASKS 9 TO 15 (FLOW: FLOW-LEVEL, PKT: PACKET-LEVEL, AC: ACCURACY, F1: F1-SCORE). FOUNDATION MODELS LIKE ET-BERT AND YATC OUTPERFORM OTHER TRADITIONAL DEEP LEARNING METHODS.

Method	Task 9		Task 10		Task 11		Task 12		Task 13		Task 14		Task 15	
	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1	AC	F1
Datanet (flow)	0.7795	0.2920	0.0481	0.0005	0.7717	0.7781	0.8261	0.4524	0.2928	0.0906	0.9768	0.4941	0.0250	0.0081
Datanet (pkt)	0.8112	0.2986	0.0429	0.0004	0.3788	0.1832	0.8187	0.4502	0.2907	0.0901	0.0078	0.0078	0.0044	0.0015
Fs-net (flow)	0.5273	0.2314	0.1094	0.0638	0.1758	0.0997	0.4023	0.2869	0.1855	0.0626	0.9023	0.7369	0.6680	0.5481
BiLSTM_ATTN (flow)	0.1152	0.0689	0.0029	0.0001	0.3860	0.1857	0.8261	0.4524	0.2609	0.0952	0.9768	0.4941	0.0579	0.0466
BiLSTM_ATTN (pkt)	0.0988	0.0600	0.0005	0.0000	0.3788	0.1832	0.8187	0.4502	0.3851	0.1658	0.9922	0.4980	0.0044	0.0015
DeepPacket (flow)	0.7795	0.2920	0.0481	0.0005	0.3660	0.1786	0.8261	0.4524	0.5332	0.1391	0.0232	0.0227	0.3435	0.0852
DeepPacket (pkt)	0.8112	0.2986	0.0429	0.0004	0.3788	0.1832	0.8187	0.4502	0.2907	0.0901	0.0078	0.0078	0.0044	0.0015
TSCRNN (flow)	0.7795	0.2920	0.0284	0.0051	0.3860	0.1857	0.8261	0.4524	0.2928	0.0906	0.9768	0.4941	0.0250	0.0081
TSCRNN (pkt)	0.0988	0.0600	0.0005	0.0000	0.3788	0.1832	0.8187	0.4502	0.2907	0.0901	0.9922	0.4980	0.0044	0.0015
ET-BERT (flow)	0.9762	0.9418	0.7705	0.7426	**0.9881**	**0.9866**	**1.0000**	**1.0000**	**0.9998**	0.9980	0.9881	**0.8535**	**0.8809**	**0.8329**
ET-BERT (pkt)	0.9911	0.9785	0.9435	0.9313	0.9736	0.9740	0.9970	0.9948	0.9701	0.9316	**0.9957**	0.8526	0.8222	0.7804
YaTC (flow)	0.9927	0.9782	0.7531	0.6957	0.9736	0.9705	**1.0000**	**1.0000**	0.9990	**0.9989**	0.9825	0.8356	0.8618	0.7316
YaTC (pkt)	**0.9981**	**0.9933**	**0.9576**	**0.9407**	0.9868	**0.9866**	**1.0000**	**1.0000**	0.9682	0.7499	0.9936	0.8440	0.8639	0.6254

B. Evaluation Results

Classification Tasks. Table III and IV illustrate the comparison results of the 8 SOTA models for traffic classification using `NetBench`. From the evaluation results, we can conclude with three main insights: (1) Traditional deep learning methods, such as DataNet, DeepPacket, Fs-net, TSCRNN and BiLSTM_ATTN, exhibit limitations in generalizing to new tasks, with a noticeable tendency to bias classifications towards dominant classes. This is evidenced by the f1-score that is consistently below 0.5 with low recall score due to data imbalance. (2) Foundation models like ET-BERT and YaTC significantly outperform these traditional approaches, showcasing their superior prediction accuracy and generalization ability. (3) Although a flow with multiple packets inherently contains richer information compared to a single packet, foundation models trained at flow level do not surpass those trained on packet-level in tasks 3, 8, and 10. This is attributed to the constraints on the input length for foundation models as the input flows exceeding a specified length will be truncated.

Generation Tasks. We also compare the generation performance of two open-sourced generative models, STAN and NetShare, as shown in Table V. It can be observed that NetShare performs well in generating IP addresses and port numbers, while STAN achieves better performance on packet length generation. The main reason is that NetShare designs bitwise encoding [16] for IP addresses and IP2Vec encoding [21] for port numbers, which helps generating these fields accurately. Conversely, STAN integrates both CNN and mixture density neural layers to capture joint distribution of packet length effectively, explaining its superior performance in generating packet lengths.

V. CONCLUSION

We introduced `NetBench`, a large-scale and comprehensive benchmark for network traffic analysis, which addressed the critical issue of fair evaluation and comparison among different methods. Characterized by its comprehensive design, `NetBench` included a total of 20 evaluation tasks across 7 datasets through a unified data processing approach. Furthermore, we evaluated some SOTA models on our benchmark.

TABLE V
EVALUATION RESULTS OF GENERATION TASKS (TASK 16 - 20) USING STAN AND NETSHARE. NETSHARE EXHIBITS GOOD PERFORMANCE IN GENERATING IP ADDRESSES AND PORT NUMBERS ON MOST DATASETS WHILE STAN PERFORMS BETTER IN GENERATING PACKET LENGTH.

Dataset	Method	16 - Source IP		17 - Destination IP		18 - Source Port		19 - Destination Port		20 - Packet Length	
		JSD	TVD	JSD	TVD	JSD	TVD	JSD	TVD	JSD	TVD
ISCXVPN 2016	STAN	**0.1130**	0.4107	**0.0850**	**0.3407**	**0.0186**	**0.1220**	**0.0247**	**0.1791**	**0.0767**	**0.3405**
	NetShare	0.1218	**0.3513**	0.1269	0.3603	0.1622	0.4966	0.1494	0.4755	0.5451	0.9011
ISCXTor 2016	STAN	0.2089	0.5505	0.2132	0.5615	0.2163	0.5658	0.1998	0.5225	**0.2173**	**0.5665**
	NetShare	**0.0168**	**0.0880**	**0.0338**	**0.1263**	**0.0736**	**0.2190**	**0.0711**	**0.2269**	0.3091	0.7004
USTC-TFC 2016	STAN	0.4421	0.8134	0.4226	0.7855	0.4070	0.7829	0.4767	0.8515	**0.3235**	**0.6952**
	NetShare	**0.0363**	**0.2148**	**0.0445**	**0.2205**	**0.2268**	**0.5899**	**0.1144**	**0.4091**	0.4908	0.8598
Cross Platform (Android) 2020	STAN	0.2446	0.5834	0.4743	0.8498	0.4089	0.7850	0.3727	0.7479	**0.2196**	**0.5845**
	NetShare	**0.0572**	**0.2909**	**0.0155**	**0.1177**	**0.0520**	**0.2177**	**0.0323**	**0.2216**	0.4266	0.8008
Cross Platform (iOS) 2020	STAN	0.3092	0.6537	0.5368	0.9093	0.4594	0.8391	0.4393	0.8181	**0.2538**	**0.6002**
	NetShare	**0.0588**	**0.2665**	**0.0261**	**0.1717**	**0.0499**	**0.2176**	**0.0217**	**0.1592**	0.3981	0.7748
CIRA-CIC-DoHBrw 2020	STAN	0.3710	0.7382	0.2823	0.5867	0.2351	0.5336	**0.0960**	**0.2511**	**0.0313**	**0.0875**
	NetShare	**0.3548**	**0.7113**	**0.2481**	**0.5828**	**0.0359**	**0.1749**	0.0963	0.3605	0.4806	0.8576
CIC IoT Dataset 2023	STAN	0.3094	0.6537	0.5370	0.9093	0.4596	0.8391	0.4394	0.8182	**0.3101**	**0.6548**
	NetShare	**0.0452**	**0.2598**	**0.0111**	**0.0817**	**0.0666**	**0.2636**	**0.0370**	**0.2126**	0.4199	0.7957

The experimental results underscored the benefits of foundation models, which demonstrated superior performance across a wide range of classification tasks. These observations highlighted the significant potential of foundation models to revolutionize network traffic analysis. For traffic generation, very few foundation models have investigated in this field. This remains to explore in the future as foundation models have the potential to excel in generation tasks as they do in classification [22]. Besides, longer input length is needed for foundation models to better understand the increasingly complex network traffic. In short, future research could leverage `NetBench` to train foundation models in a convenient and straightforward manner for a variety of downstream network traffic classification and generation tasks. We believe that our benchmark will promote advancements in foundation models for network traffic through fair and comprehensive comparisons.

REFERENCES

[1] T. P. Oliveira, J. S. Barbar, and A. S. Soares, "Computer network traffic prediction: a comparison between traditional and deep learning neural networks," *International Journal of Big Data Intelligence*, vol. 3, no. 1, pp. 28–37, 2016.

[2] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related," in *ICISSP*, 2016, pp. 407–414.

[3] A. Habibi Lashkari., G. Draper Gil., M. S. I. Mamun., and A. A. Ghorbani., "Characterization of tor traffic using time based features," in *ICISSP*, INSTICC. SciTePress, 2017, pp. 253–262.

[4] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *ICOIN*. IEEE, 2017, pp. 712–717.

[5] T. Van Ede, R. Bortolameotti, A. Continella, J. Ren, D. J. Dubois *et al.*, "Flowprint: Semi-supervised mobile-app fingerprinting on encrypted network traffic," in *NDSS*, vol. 27, 2020.

[6] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. H. Lashkari, "Detection of doh tunnels using time-series classification of encrypted traffic," in *2020 Intl Conf on Cyber Science and Technology Congress*. IEEE, 2020, pp. 63–70.

[7] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, no. 13, 2023.

[8] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "Fs-net: A flow sequence network for encrypted traffic classification," in *INFOCOM*. IEEE, 2019.

[9] C. Liu, Z. Cao, G. Xiong, G. Gou, S.-M. Yiu, and L. He, "Mampf: Encrypted traffic classification based on multi-attribute markov probability fingerprints," in *IWQoS*. IEEE, 2018, pp. 1–10.

[10] H. Yao, C. Liu, P. Zhang, S. Wu, C. Jiang, and S. Yu, "Identification of encrypted traffic through attention mechanism based long short term memory," *IEEE Transactions on Big Data*, vol. 8, no. 1, 2019.

[11] P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: Deep learning based encrypted network traffic classification in sdn home gateway," *IEEE Access*, vol. 6, pp. 55 380–55 391, 2018.

[12] M. Lotfollahi, M. Jafari Siavoshani *et al.*, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.

[13] K. Lin, X. Xu, and H. Gao, "Tscrnn: A novel classification scheme of encrypted traffic based on flow spatiotemporal features for efficient management of iiot," *Computer Networks*, vol. 190, p. 107974, 2021.

[14] X. Lin, G. Xiong, G. Gou *et al.*, "Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification," in *WWW*, 2022, pp. 633–642.

[15] R. Zhao, M. Zhan, X. Deng *et al.*, "Yet another traffic classifier: a masked autoencoder based traffic transformer with multi-level flow representation," in *AAAI*, vol. 37, no. 4, 2023, pp. 5420–5427.

[16] Y. Yin, Z. Lin, M. Jin, G. Fanti, and V. Sekar, "Practical gan-based synthetic ip header trace generation using netshare," in *Proceedings of the ACM SIGCOMM 2022 Conference*, 2022, pp. 458–472.

[17] S. Xu, M. Marwah, M. Arlitt, and N. Ramakrishnan, "Stan: Synthetic network traffic generation with generative neural models," in *International Workshop on Deployable Machine Learning for Security Defense*. Springer, 2021, pp. 3–29.

[18] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, "Ugr '16: A new dataset for the evaluation of cyclostationarity-based network idss," *Computers & Security*, vol. 73, pp. 411–424, 2018.

[19] Netresec. (2024) Splitcap - a fast pcap file splitter. [Online]. Available: https://www.netresec.com/?page=SplitCap

[20] W. Zhang, X. Meng, and Y. Zhang, "Dual-track protocol reverse analysis based on share learning," in *INFOCOM*. IEEE, 2022, pp. 51–60.

[21] M. Ring, A. Dallmann, D. Landes, and A. Hotho, "Ip2vec: Learning similarities between ip addresses," in *ICDMW*. IEEE, 2017.

[22] Q. Wang, C. Qian, X. Li, Z. Yao, and H. Shao, "Lens: A foundation model for network traffic in cybersecurity," *arXiv e-prints*, pp. arXiv–2402, 2024.