Fine-Tuning an LLM to Detect and Explain SYN DDoS Attacks

Ganesh Nanduru and Christopher Johnson

Department of Computer Science, University of Virginia

May 6, 2024

Abstract

We present PCAPLM to classify malicious network traffic and explain its results in natural language better than LLaMA2-chat, its base model. PCAPLM is designed to injest network data in the form of PCAP files and explain why specific packet flows may contain malicious traffic.

PCAPLM performs better than LaMA2-chat at classifying malicous network traffic across all relevant classification metrics. It can also give relatively intuitive explanations for its classifications. We believe that PCAPLM is a step toward a highly-explainable network traffic classifier.

1 Introduction

LLMs are large, pre-trained neural networks. Their size is usually measured in billions or tens of billions of parameters and their versatility allows them to be fine-tuned to a wide-range of applications. In recent years, transformer-based Large Language Models (LLMs) have grown massively in popularity due in part to their flexibility as general-purpose models. For example, OpenAI's ChatGPT and GPT-4 are used not only in natural language processing, but have been fine-tuned by Microsoft to power their Copilot suite.

Using LLMs for cybersecurity-related work is not a new field of study. Because LLMs use self-attestation to learn dependencies between tokens in large contexts, they are often used to uphold the NIST Cybersecurity Framework by scanning social media posts, filtering web content to effectively serve as a firewall from URLs that fit a certain category, or perhaps most promisingly, to scan system logs in real-time [MHM+24].

However, their use for detecting network threats at the Network Layer (L3) and Transport Layer (L4) is not yet well-explored. Research in this area is usually limited to detecting threats given real-time network data with results similar to those of traditional network scanners and algorithmic threat detection. Though this level of success is impressive in its own right, we believe it fails to leverage the strengths of an LLM.

In this paper we present PCAPLM, an LLM-based threat detection platform we have optimised to leverage the LLMs ability to explain *why* a given stream of packets is likely indicative of a particular attack.

2 Related Work

Work in the L3/L4 network analysis space is very limited, but there are some examples of projects that have achieved success.

Several researchers at the Technology Innovation Institute in Abu Dhabi, UAE, produced SecurityBERT, an LLM designed to detect network threats in Internet of Things (IoT) networks. SecurityBERT leverages a custom encoder for preserving privacy over the network and is designed to run with limited resources on an IoT device. They achieved 98% overall accuracy across 14 attack vectors using only 16.7MB of memory. While SecurityBERT proved to us that LLMs can be highly accurate real-time threat detection platforms, it does not attempt to explain its decision, and we did not want to build just another threat classifier. [FNT+24]

Shariq Murtuza at the Jaypee Institute of Information Technology in Noida, India designed Sentinel, an LLM designed to analyze packets and assign them a threat level. Sentinel has not yet been implemented, and we do not intend to classify at the per-packet level. [Mur24]

Fatos Xhafa at the Technical University of Catalonia, Barcelona, Spain outlines a highly effective network intrusion detection platform, but focuses on log scanning and anomaly detection. [GLMT24]

3 Experimental Setup

To assess the effectiveness of LLMs for detection of SYN DDoS attacks, we fine-tune Meta AI's LLaMA-2-chat on serialized packet flows from CICDDoS2019 [TMS+23][SLHG19].

3.1 Model selection

We select LLaMA-2-chat as our pretrained model due to its state-of-the-art performance in commonsense reasoning, world knowledge, and natural language comprehension. We use the version fine-tuned for chat capabilities to address our goal of making the DDoS attack detection system explainable. Of the published model weights (7B, 13B, 34B, 70B), we select the most lightweight model to accommodate the time and memory constraints of the project. This 7B model requires 14GB of memory to run. We use the HuggingFace library to load the model, set hyperparameters, and fine-tune [WDS+20].

3.1.1 Quantization

To further speed up fine-tuning and decrease memory consumption, we apply 4-bit model quantization, scaling the model from fp16 (half-precision float) to nf4 (normal 4-bit float). Research has shown LLM quantization can immensely decrease resource consumption while performing on par with full-precision model weights for short-context natural language tasks [LNW+24]. Quantizing LLaMA-2 reduces its memory requirements to under 2GB.

3.1.2 Hyperparameters

To freeze LLaMA-2's pre-trained weights for more effective and efficient fine-tuning, we use LoRA with an alpha of 16, a dropout probability of 0.1, and rank 8 [HSW+21].

We fine-tune with a weight decay of $1e-3$ and a learning rate of $2e-5$. Due to the potential for feature overrepresentation in the first few data flows seen by the model, we shuffle the dataset and introduce a warmup ratio of $3e-2$, which decreases the learning rate during the initial stage of fine-tuning. To address the potential for gradient explosion during fine-tuning, we set a max. gradient norm of 0.3.

We fine-tune for a single epoch to avoid overfitting on the training set and to complete fine-tuning within our time constraints. We fine-tune with a batch size of 4, and train the model for 12 hours on 4 A100 GPUs.

3.2 Dataset and preprocessing

We used the Canadian Institute for Cybersecurity's DDoS evaluation dataset. The CIC DDoS evaluation dataset contains simulated network data over the course of two days and leveraged 19 state-of-the-art DDoS attack vectors and normal simulated user traffic to generate the data.

The CIC DDoS evaluation dataset contains raw packet capture (PCAP) files and labeled "packet flows" generated using CICFlowMeter-V3, a network traffic flow generator and analyzer. Rather than working with raw PCAP files, which are excessively large and require a lot of resource-intensive processing, we opted to use the pre-generated and labeled packet flows. Due to time constraints, we decided to limit ourselves to SYN flood DDoS attacks.

The features The CIC DDoS dataset provides are in Figure 1.

CIC DDoS Evaluation dataset features

Feature No	Feature name	Feature No	Feature name	Feature No	Feature name	Feature No	Feature name
1	Flow ID	23	Flow IAT Mean	45	Min Packet Length	67	Bwd Avg Packets/Bulk
2	Source IP	24	Flow IAT Std	46	Max Packet Length	68	Bwd Avg Bulk Rate
3	Source Port	25	Flow IAT Max	47	Packet Length Mean	69	Subflow Fwd Packets
4	Destination IP	26	Flow IAT Min	48	Packet Length Std	70	Subflow Fwd Bytes
5	Destination Port	27	Fwd IAT Total	49	Packet Length Variance	71	Subflow Bwd Packets
6	Protocol	28	Fwd IAT Mean	50	FIN Flag Count	72	Subflow Bwd Bytes
7	Timestamp	29	Fwd IAT Std	51	SYN Flag Count	73	Init_Win_bytes_forward
8	Flow Duration	30	Fwd IAT Max	52	RST Flag Count	74	Init_Win_bytes_backward
9	Total Fwd Packets	31	Fwd IAT Min	53	PSH Flag Count	75	act_data_pkt_fwd
10	Total Backward Packets	32	Bwd IAT Total	54	ACK Flag Count	76	min_seg_size_forward
11	Total Length of Fwd Packets	33	Bwd IAT Mean	55	URG Flag Count	77	Active Mean
12	Total Length of Bwd Packets	34	Bwd IAT Std	56	CWE Flag Count	78	Active Std
13	Fwd Packet Length Max	35	Bwd IAT Max	57	ECE Flag Count	79	Active Max
14	Fwd Packet Length Min	36	Bwd IAT Min	58	Down/Up Ratio	80	Active Min
15	Fwd Packet Length Mean	37	Fwd PSH Flags	59	Average Packet Size	81	Idle Mean
16	Fwd Packet Length Std	38	Bwd PSH Flags	60	Avg Fwd Segment Size	82	Idle Std
17	Bwd Packet Length Max	39	Fwd URG Flags	61	Avg Bwd Segment Size	83	Idle Max
18	Bwd Packet Length Min	40	Bwd URG Flags	62	Fwd Header Length	84	Idle Min
19	Bwd Packet Length Mean	41	Fwd Header Length	63	Fwd Avg Bytes/Bulk	85	SimillarHTTP
20	Bwd Packet Length Std	42	Bwd Header Length	64	Fwd Avg Packets/Bulk	86	Inbound
21	Flow Bytes/s	43	Fwd Packets/s	65	Fwd Avg Bulk Rate	87	Label
22	Flow Packets/s	44	Bwd Packets/s	66	Bwd Avg Bytes/Bulk		

Figure 1: CIC DDoS evaluation dataset features

3.2.1 Preprocessing

LLaMA-2-chat, like many LLMs, requires structured prompts / responses for fine-tuning. In order to properly fine-tune LLaMA, we had to convert the tabular packet flow data generated from the CIC dataset using CICFlowMeter-V3 into natural language suitable for consumption by LLaMA.

We began with a simple approach - take the SYN flood data, which contains both SYN flood packet flows and benign packet flows, and create a prompt by effectively asking the LLM: `Given the following information, is this packet flow a SYN flood?` and appending the tabular data.

To our surprise, this seemingly naïve approach was somewhat successful. After fine-tuning, PCAPLM was able to give relatively accurate explanations of why a packet flow looked like a SYN flood.

3.2.2 Dataset Imbalance

After our initial fine-tuning, we observe the model exhibits heavy bias toward identifying packet flows as SYN attacks, consistently outputting: `Yes, this is a SYN flood`. To investigate this behavior, we analyzed the dataset to discover that 99.12% of the data was labeled as a SYN flood.

We decided to supplement the Benign packet flows from the SYN-specific dataset with Benign packet flows from the other 25GB of the dataset. This approach allowed us to get closer to a 10% split of Benign/SYN flood packet flows. Though this split is not representative of real-time network data during a SYN attack, because the model handles each flow on a case-by-case basis, we do not anticipate that this will limit the model's ability to accurately classify SYN floods.

3.3 Evaluation

We evaluate models on a test set constructed from 100 sampled packets from CICDDoS2019. To assess different rates of prediction, we test on datasets with different distributions of Syn/benign packet flow ratios. The test size is relatively low due to the longer inference time required for LLM chatbots, compared to typical neural networks.

To validate the outputs of the models, we must account for surface-form competition in our post-processing [HWS+22]. While typical neural networks can simply output a binary 1/0 depending on which class they recognize the input to belong to, chatbots respond in a more verbose manner. We cannot simply query for the strings "benign" or "SYN" in model outputs due to responses such as, "It is unlikely that this packet flow is a SYN attack." Therefore, we carefully develop a set of substrings

to validate output, and if none of the substrings are found, our evaluation script requests human annotation for the test sample.

For each test run, we report a confusion matrix of false/true positives and negatives. We use these results to calculate accuracy, precision, recall, and f1-score for each model evaluation.

4 Results

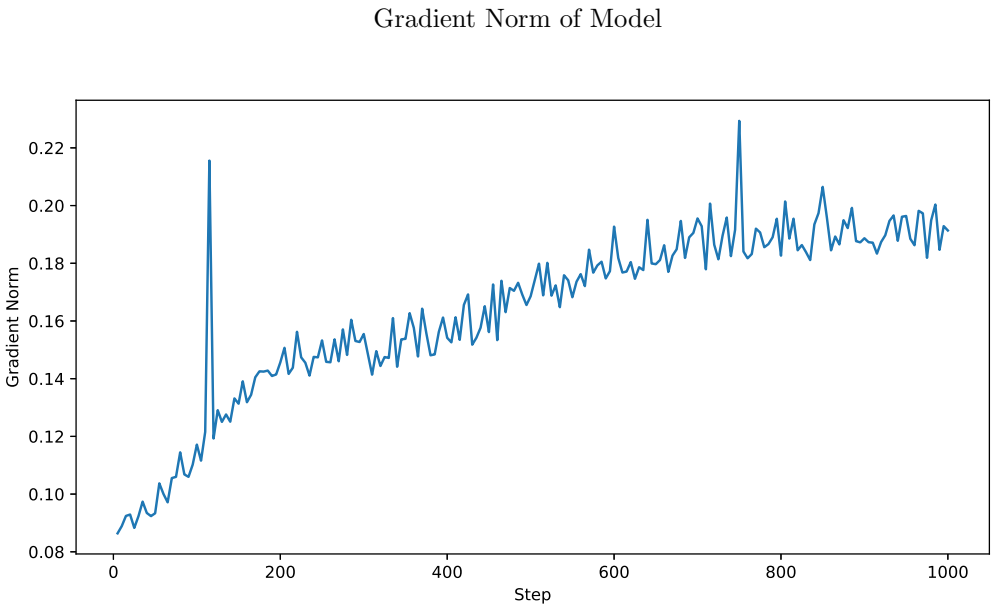

Figure 2: This graph shows the change in gradient norm over each training step during the model's fine-tuning.

We present training metrics of gradient norm and training loss recorded during fine-tuning of the initial model. In Fig. 2, we observe the gradient norm remains under 0.22, which indicates the model's loss function did not encounter any problems with exploding gradient during training. Otherwise, we would notice gradient clipping occuring at our max. gradient norm of 0.3. In Fig. 3, we observe a typical convergence of an LLM on a consistent text corpus, starting at a higher loss and gradually decreasing.

	LLaMA2-chat	PCAPLM-50	PCAPLM-100	PCAPLM-150
Accuracy	14.0	87.0	82.0	80.0
Precision	N/A†	96.2	97.2	98.5
Recall	0.00	88.4	81.4	77.9
F1-Score	N/A†	92.1	88.6	87.0

Table 1: Evaluation metrics of all models. †LLaMA2-chat had 0 true positives, leaving precision and F1 incalculable.

Table 1 shows the accuracy, precision, recall, and F1-score across three different checkpionts of PCAPLM and compares them to LLaMA2-chat, the base model. As fine-tuning progressed to later checkpoints (PCAPLM-100/150), the model improved in precision, but had a lower accuracy, recall, and F1-score, which is indicative of overfitting.

PCAPLM also significantly improves model explainability. Figure 4 and Figure 5 show the model output on the same SYN-flood packet flow for LLaMA2-chat and PCPAPLM, respectively. The LLaMA2-chat output in Figure 4 has what appears to be a valid explanation, but upon closer inspection, contains redundant (`SYN packets are present` and `SYN flag is set`) and erroneous (`No`

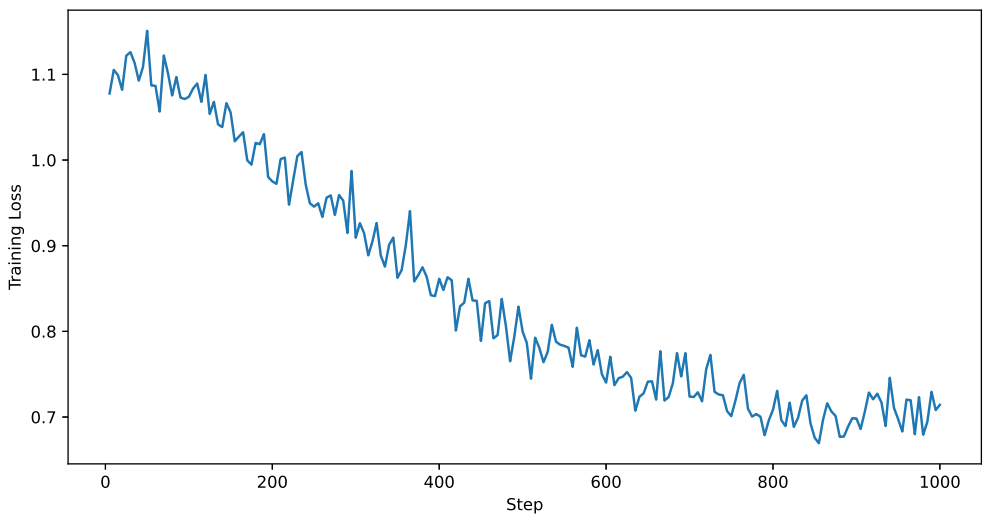

Figure 3: This graph shows the change in training loss over each training step during the model's fine-tuning.

LLaMA2-chat model output for a SYN flood

```
1. SYN packets are present: The packet flow contains SYN packets, which is a characteristic of a SYN attack.
2. SYN flag is set: The SYN flag is set in the packet flow, indicating that the SYN packets are being sent.
3. No ACK packets: There are no ACK packets present in the packet flow, which is consistent with a SYN attack as the attacker is
4. High packet rate: The packet rate is high, which is consistent with a SYN attack as the attacker is sending a large number of
5. No RST packets: There are no RST packets present in the packet flow, which is consistent with a SYN attack as the attacker is
6. No PSYNC packets: There are no PSYNC packets present in the packet flow, which is consistent with a SYN attack as the attacke
7. Low packet length: The packet length is low, which is consistent with a SYN attack as the attacker is sending small packets to
8. No fragmentation: There is no fragmentation present in the packet flow, which is consistent with a SYN attack as the attacker
```

Figure 4: LLaMA2-chat model output for a SYN flood

`ACK packets`) information. The PCAPLM output in Figure 5 gives valid, clearly outlined reasoning for why the packet flow appears to be a SYN flood.

We observe in Figure 6 that LLaMA2-chat simply guesses every packet flow is benign. PCAPLM, however, performs with greater accuracy at the cost of occasionally detecting false positives and false negatives.

5 Discussion and Limitations

PCAPLM shows immediate growth on the CIC dataset upon the first 50 steps of fine-tuning when compared to the base LLaMA2-chat model. However, we observe in Figure 1 that as fine-tuning continues, precision increases at the cost of accuracy and recall. Precision is the proportion of positive guesses that are correct, meaning the more fine-tuned checkpoints are more likely to be correct when they guess a packet flow is a SYN attack, but this comes at the cost of missing some SYN floods as false negatives.

PCAPLM represents the utility LLMs can provide to the network analysis space. It leverages the natural language generation capabilities of LLaMA to create accurate descriptions of why certain PCAP flows are likely SYN floods, something that no other network analyzer, LLM or otherwise, can currently accomplish.

However, PCAPLM does have some limitations relative to other network vulnerability classifiers. Its most obvious limitations are self-imposed restrictions we placed on the model in order to work under time constraints.

PCAPLM model output for a SYN flood

```
1. High packet flow: The packet flow rate is very high, with 12.0 fwd_packets and 2.0 bwd_packets.
2. Low latency: The average latency of the packets is 103908486.0, which is very high compared to n
3. High packet length: The average packet length is 0.0, which is unusual for network traffic. This
4. Presence of SYN headers: The presence of SYN headers in the packets is a clear indication of a S
5. Lack of urgent packets: There are no urgent packets ( packets with the bit set) in the flow, whi
6. Low byte/s rate: The byte/s rate is 0.0, which is very low compared to normal network traffic. T
7. High fwd_IAT_Mean: The forward IAT (Initial Action Time) mean is 9446226.0, which is very high c
8. High fwd_IAT_Std: The forward IAT standard deviation is 11974133.847526291, which is very high c
9. High bwd_IAT_Mean: The backward IAT mean is 0.0, which is very low compared to normal network tr
10. Low bwd_IAT_Std: The backward IAT standard deviation is 0.0, which is very low compared to norm
```

Figure 5: PCAPLM model output for a SYN flood

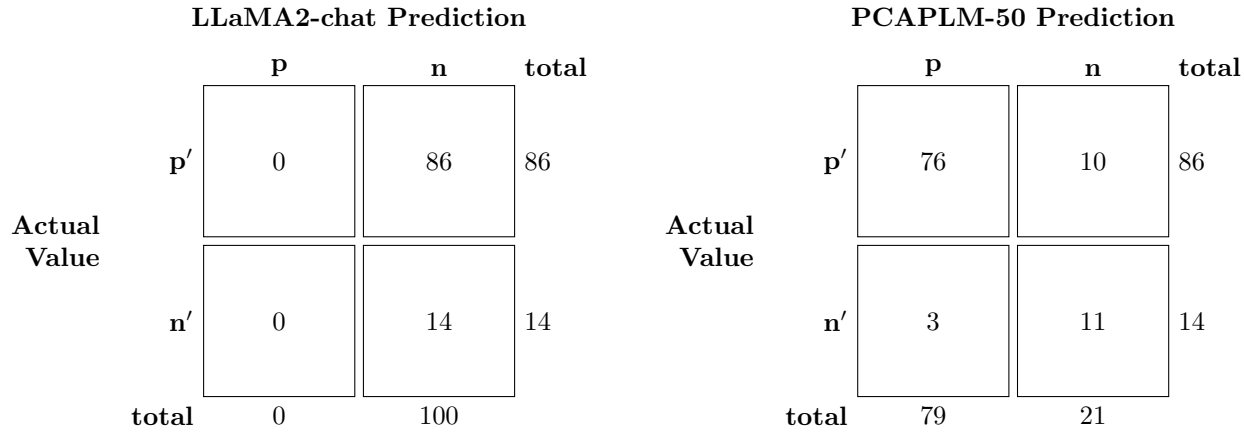

Figure 6: Confusion matrices for LLaMA2-chat and PCAPLM-50 predictions. Positive (p) entries correspond to SYN attacks and negative (n) entries correspond to benign packet flows.

First, it is limited to only DDoS attacks. We limited it to this space due to the lack of availability of large, open-source PCAP datasets. The Canadian Institute for Cybersecurity was the only one large enough that we could comfortably train an LLM on it without running over multiple epochs and overfitting.

Second, within that space, it is limited to only SYN attacks. The CIC dataset provides ample data for other DDoS attack vectors. However, given the time constraints, we decided it would be more useful to create a model that could better explain one type of DDoS attack vector than one that provided poor explanations of several attack types.

Lastly, some of the packet flow descriptions are inaccurate. For the most part, on flows the model is able to classify as a SYN packet, it is able to inspect the packet flow and create a good explanation. For example, if the packet flow has 14 forward (sent) packets, 14 SYN flags, and 2 backward (received), the model can describe why this is unlikely regular user data. However, for benign packets incorrectly classified as SYN floods, the model will generate a similar response with data that contradicts its own logic.

We argue that these limitations are not architectural, and are rather self-imposed or otherwise required because of our current implementation. Solutions to these issues will be discussed in Section 6.1.

6 Conclusion and Related Work

The results of this study underscore the efficacy of training on a modest text corpus for a targeted classification objective, yielding a proficient network traffic analyzer prioritizing comprehensive result explanations. While PCAPLM demonstrates superior performance compared to LLaMA2-chat, further refinements and advancements remain imperative.

6.1 Future Work

To expand on this project, we intend to do three things:

1. Generate our own data. We intend to use ns-3 to simulate a network to give us fine-grained control over the data we use. This way, we can generate our own packet flows and create arbitrary data distributions for any attack vector we choose.

2. Classify more attack vectors. If we are generating our own data, we can also create malicious traffic for more attack vectors than just DDoS / SYN flood.

3. Better annotations. Currently, we simply classify as `Yes, this is a SYN flood.` or `No, it is benign.` We then give the model a large enough output window that it generates explanations on its own. However, if we trained with better explanations of why something is classified a certain way, we believe we can create better explanations.

Combining these methods should allow us to create a much more versatile network with significantly better explanations, but will require much more work that was possible given the time frame we allotted.

7 Member Contributions

Ganesh Nanduru - Ganesh has significantly more experience in working with Large Language Models than Christopher. As such, he did all of the model training and fine-tuning, and a significant amount of model debugging.

Christopher Johnson - Christopher has significantly more experience in the cyber domain, specifically with DDoS attacks and analyzing network traffic via PCAP files. He wrote a significant amount of the data preprocessing code and used his subject matter expertise to aid in model debugging as applicable.

Both members contributed equally to the presentation and paper.

References

[FNT+24] Mohamed Amine Ferrag, Mthandazo Ndhlovu, Norbert Tihanyi, Lucas C. Cordeiro, Merouane Debbah, Thierry Lestable, and Narinderjit Singh Thandi. Revolutionizing cyber threat detection with large language models: A privacy-preserving bert-based lightweight model for iot/iiot devices, 2024.

[GLMT24] Oscar G. Lira, Alberto Marroquin, and Marco Antonio To. Harnessing the advanced capabilities of llm for adaptive intrusion detection systems. In Leonard Barolli, editor, *Advanced Information Networking and Applications*, pages 453–464, Cham, 2024. Springer Nature Switzerland.

[HSW+21] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[HWS+22] Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right, 2022.

[LNW+24] Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Evaluating quantized large language models, 2024.

[MHM+24] Farzad Nourmohammadzadeh Motlagh, Mehrdad Hajizadeh, Mehryar Majd, Pejman Najafi, Feng Cheng, and Christoph Meinel. Large language models in cybersecurity: State-of-the-art, 2024.

[Mur24] Shariq Murtuza. Sentinels of the stream: Unleashing large language models for dynamic packet classification in software defined networks – position paper, 2024.

[SLHG19] Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali A. Ghorbani. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8, 2019.

[TMS+23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[WDS+20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing, 2020.