When Students Chat: Router-Based Detection of Unauthorized LLM Usage in Encrypted Enterprise Networks

Gavin Crigger, Tao Groves, Matthew Lucio, Sebastian Wiktorowicz*
University of Virginia
Charlottesville, Virginia

ABSTRACT

Artificial intelligence companies have had a massive boom in the public market within the past 5 years, creating widespread availability for querying LLMs over the internet. This has impacted academic integrity within universities and information security in sensitive workplaces, thus we propose an approach to identifying LLM versus non-LLM traffic flows in encrypted enterprise networks. With data collected and various models trained for ChatGPT, Gemini, and Claude flow classification, we aim to provide network administrators with a methodology for connecting timestamped browser LLM usage with corresponding network users. Our models had accuracies between 85%-97% with notably lower recall rates ranging from 48%-75%. Future work and considerations should be geared towards more extensive data collection, fine-tuning classification models, and collaboration with network administrators to create a more complete and practical workflow.

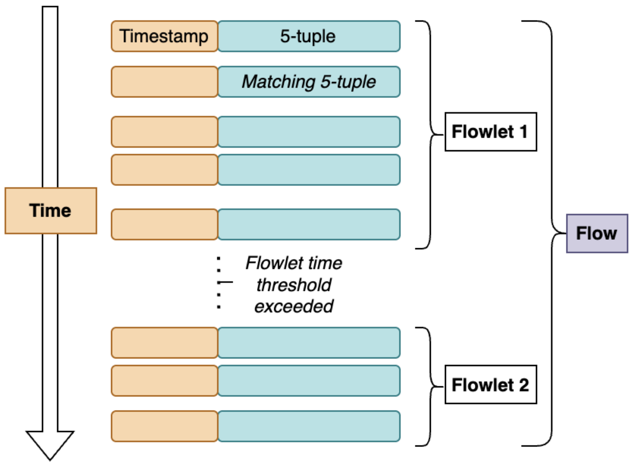

Figure 1: Classification of network traffic into flows and flowlets, used for further insight into flow-level behavior.

1 INTRODUCTION

With the exponential increase in usage of AI tools, instructors must come up with creative ways to continue testing their students properly. It is not difficult for a student to use generative AI to complete an assignment in violation of academic integrity guidelines. This cheating creates a norm for academic dishonesty, and this paper is intended to create a framework for LLM traffic detection that will aid in dissuading such dishonesty. This will mainly be done by accessing encrypted network packets securely and without exposing sensitive information about the affected user. Furthermore, research will specifically target sensitive workplace and school networks, such as the encrypted enterprise 'eduroam' network at the University of Virginia, and will thus be tailored to packet streams following such architecture standards. This research is for fingerprinting generalized application flows rather than specific applications, and thus we intend this work to extend to further efforts in classifying abstract traffic flows without compromising the privacy of encrypted networks.

2 BACKGROUND

The initial proposal for this research was intended to use entropy analysis to identify header meanings and any other proprietary LLM packet attributes that could be used for identification [10]. Because the research scope is constrained to identification over encrypted networks, we decided that entropy analysis would not be effective, as real-time analysis deals with encrypted packets and enforcing decryption violates enterprise privacy goals. Instead, we draw on research regarding traffic classification methods given encrypted packets. In particular, we refer to website fingerprinting via machine-learning models in developing our process.

Website fingerprinting is a well-researched and implemented process for identifying connections to applications via traffic flow analysis, application signatures, and classification models. Additionally, prior research has found success in using fingerprinting techniques for traffic classification even with Tor, demonstrating transference to our goal of

*Paper authored and research conducted as part of CS 7457 - Computer Networks at the University of Virginia School of Engineering and Applied Sciences, taught by Hyojoon Kim, PhD.

classification regardless of Tor or VPN usage [11]. These methods are often tailored to specific use cases, however, and are thus not entirely applicable to our goal of identifying any LLM usage - we instead focus on methodologies that approach general process identification (like streaming or browsing). Liu et al approach the problem of website fingerprinting over an encrypted network with trained Markov models and network traffic measurements to achieve impressively high true positive and low false positive rates [7]. We hope to abstract this workflow of collecting encrypted traffic data and training classification models to a more generalized approach of fingerprinting LLM versus non-LLM traffic.

Network packets can be classified into 5-tuples with timestamps regardless of encryption, where each 5-tuple contains src-IP, dst-IP, src-port, dst-port, and protocol. A traffic "flow" can be identified as all of the packets with identical 5-tuple characteristics and represents a connection (and all of the corresponding communication) between two end-hosts. When labeled by timestamp, flows can be partitioned according to a time threshold into "flowlets" that represent more fine-grained aspects of the connection, like individual messages as part of a larger conversation between two hosts, resulting in better insight on flow-level behaviors [8].

Our research is done under the assumption that encrypted enterprise networks operate with identifiable attributes that allow users to communicate with the network (and access the internet). We assume that users authenticate with the network before being granted access and that this authentication grants an identifiable feature that links users and their corresponding network traffic. The eduroam network, for example, uses a Chargeable-User-Identity attribute that can optionally be used to identify which user is responsible for traffic flows at the router [1]. This assumption allows us to further assume that successful classification of encrypted traffic flows extends to identification of the users responsible for those flows.

Due to the domain-agnostic nature of the proposed detection method, it is theoretically possible that the method could be applied to routing obfuscation technologies such as Virtual Private Networks (VPNs) or Tor Onion Routing (Tor). In fact, this could be considered a significant utility of such a method. However, from limited testing on this application it was determined that the presence of VPNs and Tor substantially complicates the process of detecting LLM usage purely from packet-level metadata, as both may significantly alter the exact kind of traffic features the proposed model relies on. VPNs often aggregate flows and restructure packet sizes, and Tor deliberately standardizes packets into fixed-length cells. Prior work has shown strong results detecting specific application traffic through a VPN [9], as well as characterizing encrypted traffic based purely on time-related features [3]. However, these methods involved significantly more complex classification models as well as additional flow features not measured by the proposed model, and were unable to achieve the same accuracy. Tor in particular requires extensive effort to demystify, and a user could configure their Tor client to make time-based analysis almost impossible at the cost of slower traffic speeds [6]. As such, LLM detection on obfuscated traffic has been omitted from this work. The primary benefit of the model as described, therefore, is the ability to detect a wide variety of LLM traffic. Even if a malicious network user hosted an LLM on their own server on a brand new domain, this would be flagged by the model.

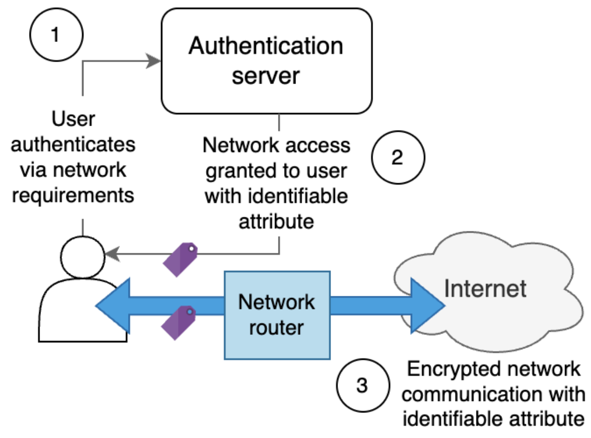
1
Authentication server
User authenticates via network requirements
Network access granted to user with identifiable attribute
2
Network router
Internet
3
Encrypted network communication with identifiable attribute

Figure 2: Assumed authentication flow users must follow before interacting with encrypted enterprise networks.

3 METHODS

3.1 Data Collection

Data collection was carried out on two separate networks: the IPv4 eduroam network at the University of Virginia (measured in more than one building) and an IPv6 home network. Upon beginning a Wireshark packet capture, we issued a "long query" request to the LLM being measured (Gemini, ChatGPT, or Claude all via browser). The individual collecting data would then identify the IP streaming the response via the Wireshark interface and use that IP to begin a tailored packet capture via a Python script that collected all traffic going to and from that IP address. Once this script was running, the LLM was queried many more times, and all resulting data exchange was parsed into a .txt file. Each text file is named according to the timestamp when it was taken, the IP address being captured, and the general query flow that took place - for example, capture_20251201_203821_2607-6bc0–10_128_CODINGQUERY.txt represents an IPv6 capture for address 2607-6bc0–10 with the packets corresponding to a single coding query from the LLM. Such naming practices aimed to identify any potentially anomalous classification results corresponding with measured behaviors.

3.2 Traffic Representation

The text files captured from LLM and non-LLM usage then had to be parsed to be used in classification models. Because the text files only had individual packet captures, it was useful to group these packet captures into flowlets, representing packets streamed consecutively. The methods for identifying flowlets are described in further detail in section 4.1.

The captures were parsed into flowlets in a .json file where each flowlet had a unique 5-tuple key, consisting of the src-IP, dst-IP, src-port, dst-port, and protocol (all falling in line with background on flow classification). For each of these 5-tuples, the features gathered included:

- Start Time
- End Time
- Duration
- Packet Count
- Total Bytes
- Inter Packet Time Mean
- Inter Packet Time Standard Deviation
- Packet Size Mean
- Packet Size Standard Deviation
- Inter-Packet Times
- Packet Sizes

Additionally, each of the flowlets contained a field for the traffic class it was part of (non-LLM or LLM), which served as the ground truth for the models as well as the name of the source file, which would help identify the particular LLM used. Each of the features associated with a particular flowlet key along with the traffic class was used to train the classification models.

3.3 Flowlet Classification

Three classifier types — Random Forest, XGBoost, and SVM — were trained for each of three LLMs (ChatGPT, Gemini, and Claude), resulting in nine total models. These were the primary classification approaches evaluated, though additional models may be trained to gather further results.

The extracted flowlet .json file was used to train models to classify flowlets associated with LLM usage. Each flowlet was labeled based on the LLM used during capture (ChatGPT, Gemini, or Claude) or as non-LLM traffic, using the source packet capture file described in Section 3.2. After labeling, the flowlets were input into the three models corresponding to the LLM label. For example, ChatGPT-labeled flowlets were used to train and evaluate the three ChatGPT models against the non-LLM flowlets.

The features used for classification included duration, packet count, total bytes, inter-packet time mean, inter-packet time standard deviation, packet size mean, and packet size standard deviation, similar to the features in the flowlet collection stage. The features were tested for correlation with the LLM and the results of the correlations and classification of the models are detailed in section 4.

4 RESULTS

This section details quantitative results from data collection, feature correlation metrics, and measurements from classification models on each use case.

4.1 Data

We collected 10 captures of non-LLM data and 40 captures of LLM data. Non-LLM data captures had significantly more flows/flowlets due to capturing all traffic on the end-host device as opposed to traffic to and from a single IP address. A more complete representation of the data we collected is shown when quantified as packets, flows, and flowlets. We collected 748382 LLM packets and 2058944 non-LLM packets, which correspond to the total number of lines for each capture type (each line stores information for one packet). This then split up into 45,368 LLM flowlets and 121,752 non-LLM flowlets partitioned by a time threshold of 0.1 seconds. Tables 1 and 2 further represent the distribution of our collected data.

Table 1: Packet and flowlet dataset summary

Category	Count	Notes
LLM packets	748,382	packets (rows) from LLM captures
Non-LLM packets	2,058,944	packets (rows) from non-LLM captures
LLM flowlets	45,368	derived after flowletization
Non-LLM flowlets	121,752	derived after flowletization
Time threshold	0.1 s	flowlet partition threshold

4.2 Feature Analysis

The correlation of features with the target classification in each of the models was initially identified through a heatmap, with key correlations shown for each of the models. In addition to the correlations, the results of each of the models are shown in their corresponding subsections.

4.2.1 ChatGPT. The most correlated feature for our ChatGPT models was the packet size mean. Features related to

Table 2: Flowlet distribution by label

Label	Flowlets	Percentage
ChatGPT	10,764	6.4%
Claude	16,090	9.6%
Gemini	18,514	11.1%
Non-LLM	121,752	72.9%
Other	0	0.0%

inter-packet time, like mean and standard deviation time between packets, were next highest for feature correlation with the target. Finally, flowlet duration (time between the first and last packets of a flowlet) also turned up positively correlated with the target label (albeit rather low). Figure 3 represents our full findings on feature correlation.

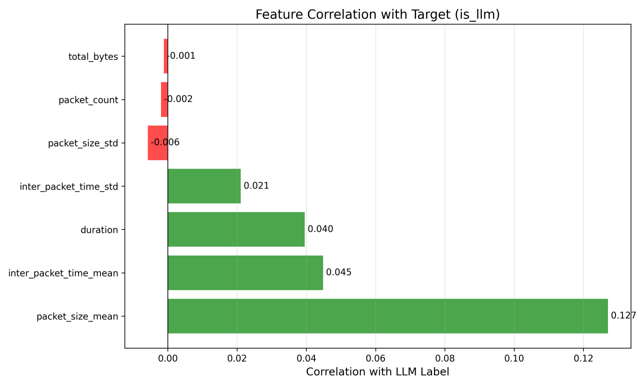

Figure 3: Flowlet feature correlation with target label for ChatGPT data.

The packet size for ChatGPT flowlets differed substantially from non-LLM flowlets as seen in Figure 4. Although the distribution range was similar between 0-1400 bytes, the ChatGPT flowlets saw a higher concentration of packets near the 1300 byte mark, as well as a higher concentration near the 200-400 byte mark. On the other hand, non-LLM packets tended to stay below 50 bytes each, demonstrating packet size as an important and distinct feature in classification.

4.2.2 Gemini. The most correlated features for the Gemini flowlets were also related to packet size as well as packet distribution. Specifically, the most correlated features were packet size standard deviation, inter-packet standard deviation, and inter-packet time mean. These 3 features each had a correlation coefficient of between 0.15-0.2 as shown in comparison to other features in Figure 5. Interestingly, this is much higher than the average correlation of ChatGPT

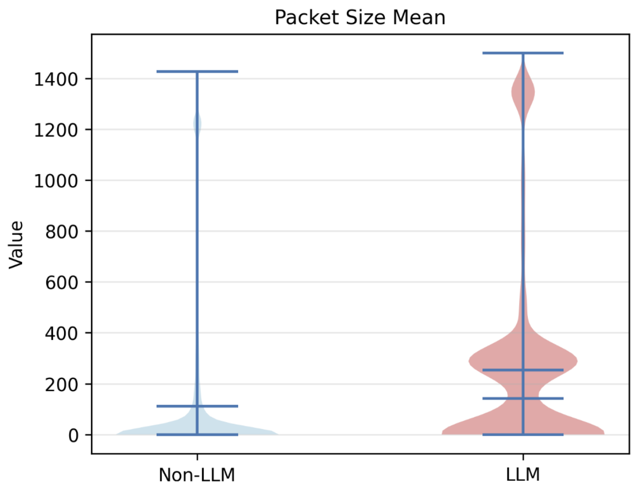

Figure 4: ChatGPT packet size mean distribution

features (the most correlated feature, mean packet size, only having a correlation coefficient of 0.127).

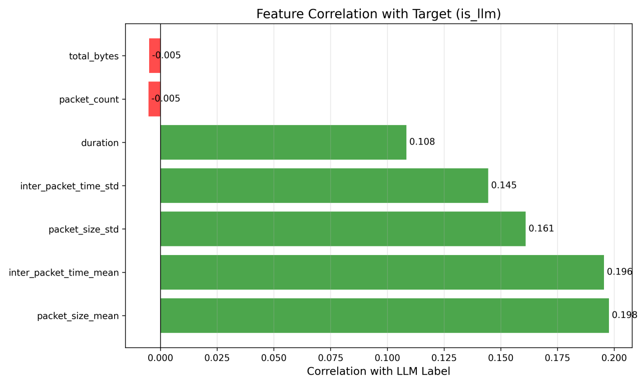

Figure 5: Feature correlation with Gemini flowlets.

Gemini flowlets also have a correlation with packet size mean, where the distribution of the LLM packet sizes are different than that of the non-LLM packet sizes (see Figure 6). The packet sizes of the non-LLM are concentrated 30 bytes whereas the LLM packet sizes are concentrated around 50 and spread between 150 and 600 bytes, and thinner between 600-1400 bytes. This difference in distribution gives the model a good feature to identify LLM packets on.

4.2.3 Claude. Inter-packet time mean, inter-packet time standard deviation, and duration of each of the flowlets turned out to be most important for the Claude flowlets as illustrated in Figure 7. The correlations were 0.663, 0.554, and 0.453 respectively - these coefficients are not only higher than the ChatGPT features but also the Gemini features.

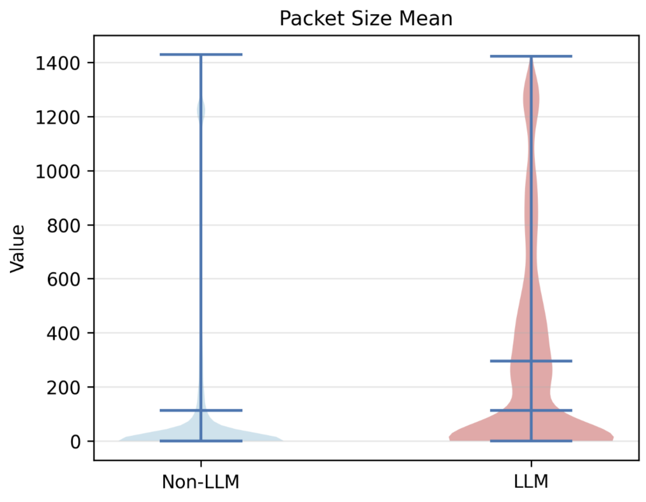

Figure 6: Gemini packet size mean distribution

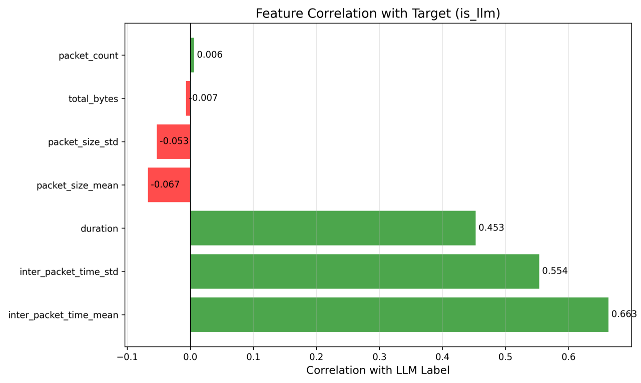

Figure 7: Feature correlation with Claude flowlets.

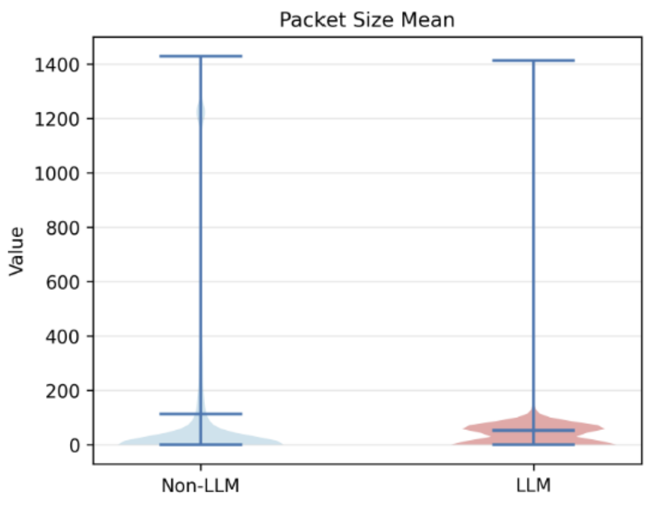

Figure 8: Claude packet size mean distribution

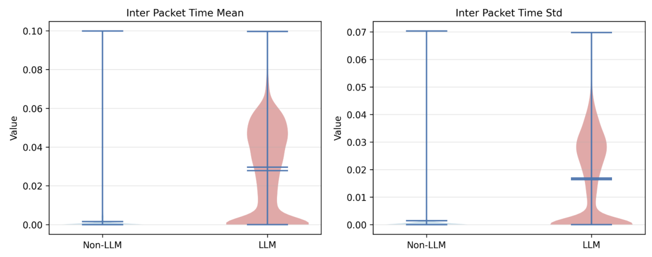

Figure 9: Claude Inter Packet Time Statistics

Figure 8 also shows that the packet size mean is similar for Claude packets in flowlets when compared to that of non-LLMs. This is different from ChatGPT and Gemini packets that saw the packet size differ from non-LLMs. In contrast, according to Figure 9, the inter-packet time mean and standard deviation were distinctly different for Claude packets compared to LLMs. The mean and standard deviation of inter-packet times for non-LLM packets were close to 0.00s, but the Claude packets were concentrated at 0.05s and distributed towards another concentration at 0.00s for the mean and concentrated at 0.03s for the standard deviation distributed towards another concentration towards 0.00s.

4.3 Classification Results

The results of the classification models were promising for our use-case of identifying unwanted LLM-usage. The LLM/non-LLM flowlet classification accuracy was high at 98, 88, and 97 percent respectively for the ChatGPT, Gemini, and Claude models. Detailed results for each of the models for all LLMs are shown in Table 3. Additionally, the precision for all classifiers was high, meaning that most of the flowlets flagged as LLM were indeed LLM flowlets.

Table 3: Model performance by provider

Provider	Model	Accuracy	Precision	Recall	F1
ChatGPT	Random Forest	0.9792	0.8559	0.5976	0.7038
ChatGPT	Svm	0.9730	0.7166	0.5732	0.6369
ChatGPT	Xgboost	0.9774	0.8121	0.5884	0.6824
Gemini	Random Forest	0.8778	0.9259	0.5651	0.7019
Gemini	Svm	0.8587	0.9227	0.4855	0.6362
Gemini	Xgboost	0.8818	0.9275	0.5810	0.7144
Claude	Random Forest	0.9672	0.9305	0.7522	0.8319
Claude	Svm	0.9533	0.8764	0.6607	0.7534
Claude	Xgboost	0.9694	0.9504	0.7557	0.8419

Although the accuracy and precision were high for these classifiers, the recall was lower, coming in at 59 percent, 58 percent, and 75 percent, respectively for ChatGPT, Gemini, and Claude. However, we interpret a tradeoff of high accuracy for lower recall as a good sign for our intended use case. This distinction means that most of the flowlets flagged for LLM usage will, in fact, correspond to LLM flowlets, leading to a high true positive rate and low false positive rate. This is beneficial to applications of the models as users will likely want to know definitively when an LLM is being used - misclassifying some flowlets in the process is not detrimental to the use case.

4.4 General Findings

By exporting SSL secrets and performing packet inspection in Wireshark and tshark, we successfully decrypted the TLS traffic to observe the underlying application-layer protocols. Overall, we noticed that the LLM packet contents varied tremendously. ChatGPT's traffic pattern is characterized by an initial, substantial prepare token payload, followed by a sequence of redundant "success":"true" keep-alive packets. The model's inference is then delivered via a high-frequency stream of small payloads, consistent with real-time, token-by-token generation. In contrast, Gemini's network signature exhibited a distinct pattern of packet repetition, suggesting a different approach to reliability.

5 FRONT-END APPLICATION

We developed a React-based web dashboard that enables network administrators to monitor enterprise network traffic and interact with our LLM traffic classification system. The application provides an interface for the end-to-end workflow from packet capture to ML inference, integrated with a FastAPI backend and SQLite database. The dashboard displays a table of all network captures with metadata, flow counts, and classification results, with automatic updates every 5 seconds. Administrators can start new packet captures through a configuration dialog that supports standard captures and optional TLS decryption when SSL keys are provided.

The system automates the processing pipeline with one-click actions to parse raw captures into flowlets and run ML classification models, storing all results in a centralized database. When TLS decryption is enabled, the system can identify LLM services from decrypted traffic, providing ground truth labels that enable real-time validation of model predictions. The detailed capture view includes interactive time-series visualizations of traffic patterns and a comparison table showing model predictions alongside ground truth when available, allowing administrators to assess classification accuracy.

The purpose of this front-end application is to streamline the workflow from packet capture through flowlet extraction to ML inference, enabling near real-time monitoring of LLM traffic and model validation in enterprise network environments. Future work will extend this platform to include dynamic visualizations of simulated enterprise network topologies with multiple routers and devices, similar to Cisco Packet Tracer, enabling more comprehensive network traffic analysis and monitoring capabilities.

6 DISCUSSION

6.1 Classification

Our results show that flowlet-level metadata provides a strong signal for distinguishing LLM traffic from non-LLM traffic, even under full transport-layer encryption and without reliance on domain knowledge or payload inspection. The poor recall rate is representative of a deliberately conservative decision boundary that prioritizes minimizing false positives over exhaustively identifying every LLM-related flowlet, particularly in the presence of short-lived or weakly expressive flowlets that resemble background traffic. This limitation could likely be reduced by including additional measurements [3] and using a larger, more diverse dataset for model training. Furthermore, the precision shows that when the model does flag a flowlet as LLM-generated, it is highly likely to be correct, which is critical in institutional settings where erroneous accusations of misuse would be unacceptable. An ideal model, given our use case of LLM flow classification via an encrypted network router, would therefore emphasize high precision, robustness to heterogeneous non-LLM traffic, interpretability of feature contributions, and stability across networks and LLM providers.

While testing on traffic obscured by a VPN was not included in this work, it is theoretically possible that our model can already detect LLM traffic through some VPNs, especially on a client device with otherwise low network utilization. Most VPN clients attempt to minimize added latency, even at the packet level, and may not significantly alter inter-packet delay [4].

6.2 Weaknesses

As discussed previously, our model is vulnerable to traffic obfuscation methods such as VPNs or Tor. Prior work has shown that traffic characterization through these barriers is possible, but requires significantly more granular flow metrics [3]. Extending the proposed method to overcome this challenge would require a much stronger dataset, as well as significantly larger detection models which come with their own tradeoffs[5]. The end of this research presents a brief set of methodologies and results for identifying hidden services via website fingerprinting techniques, which could serve as an effective framework for future research on distinguishing LLM versus non-LLM flows over Tor. A key consideration here is that this process necessitates new data collected as a

Tor client and corresponding model retraining. Due to the constant sizing Tor packets use, future work should focus on non-size features when considering Tor traffic identification, like inter-packet time measurements and non-aggregation metrics.

Another potential drawback to our methods is data collection methods. All collected traffic was captured at the end-host, which is not reflective of the intended use-case as router-based detection. There is consensus among prior work that mirroring the environment that a targeted client is using is an effective tactic for appropriately training a classification model. Factors like computation and telemetry ability as well as packet appearance could differ, yet we are fairly confident that there is transference to a router-based methodology, particularly because the end-host traffic collected was being sent directly to the router in both the eduroam and home-network setting. We propose collaboration with network administrators in university settings for future work to more accurately reflect router-based detection.

6.3 Extensibility

There is a significant amount of room for fine-tuning features, weights, and model selection to more effectively classify these flows. Although we only used 3 models, this should mark the beginning of extensive research potential regarding classification of LLM traffic flows. We also intend for this general framework to be applicable to identifying other application-behavior flows via fingerprinting. Rather than just differentiating between LLM and non-LLM, future work could investigate diverse application categories like distinctions between different types of content streaming.

7 CONCLUSION

Our project demonstrates an LLM packet fingerprinting model to detect LLM usage in encrypted enterprise networks with important applications for cheating detection in schools or leaked information in classified work settings. We utilized flowlet-based statistical analysis to identify traffic patterns of popular LLM websites (ChatGPT, Gemini, and Claude) that will ideally generalize to many LLM traffic flows, potentially over VPNs and Tor. After capturing packets and parsing them into flowlets, we trained and tested Random Forest, SVM, and XGBoost models, achieving high accuracy but low recall for distinguishing between LLM traffic and non-LLM network packets. Finally, we developed a front-end application that a network administrator could use to monitor real enterprise network traffic for LLM usage. Overall, our work shows how enterprises can monitor their network traffic while preserving privacy and avoiding host-based surveillance through statistical traffic patterns at the router-level.

8 ETHICS

Privacy is an integral part of encrypted enterprise networks. Our research upholds user privacy assumptions by not decrypting any packets and operating solely on visible attributes of their traffic flow. All training data came from either home network packet captures or eduroam network packet captures limited to our personal devices (no sniffing of non-consenting party traffic). Furthermore, this work presents an alternative to host-based intrusion detection systems, which are a far more invasive approach to monitoring user activity. Extensions of this work should consider the privacy and safety implications of identifying and revealing student location through router-based detection.

9 ARTIFACTS

Our code and scripts for evaluation can be found at [2].

REFERENCES

[1] F. Adrangi, A. Lior, J. Korhonen, and J. Loughney. 2006. Chargeable User Identity. *Internet RFCs, No ISN* RFC 4372 (2006). https://www.rfc-editor.org/rfc/pdfrfc/rfc4372.txt.pdf

[2] Gavin Crigger, Tao Groves, Matthew Lucio, and Sebastian Wiktorowicz. 2025. Project Artifacts. (2025). https://github.com/MatthewELucio/Networks-Project

[3] Gerard Draper-Gil, Arash Habibi Lashkari, Mohammad Saiful Islam Mamun, and Ali A. Ghorbani. 2016. Characterization of Encrypted and VPN Traffic using Time-related Features. In *International Conference on Information Systems Security and Privacy*. https://api.semanticscholar.org/CorpusID:21535780

[4] S. Khanvilkar and A. Khokhar. 2004. Virtual private networks: an overview with performance evaluation. *IEEE Communications Magazine* 42, 10 (2004), 146–154. https://doi.org/10.1109/MCOM.2004.1341273

[5] Albert Kwon, Mashael AlSabah, David Lazar, Marc Dacier, and Srinivas Devadas. 2015. Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services. *Proceedings of the 24th USENIX Security Symposium* (2015), 287–302. https://www.usenix.org/conference/usenixsecurity15/technical-sessions/presentation/kwon

[6] Arash Habibi Lashkari, Gerard Draper-Gil, Mohammad Saiful Islam Mamun, and Ali A. Ghorbani. 2017. Characterization of Tor Traffic using Time based Features. In *International Conference on Information Systems Security and Privacy*. https://api.semanticscholar.org/CorpusID:38122314

[7] Chang Liu, Zigang Cao, Gang Xiong, Gaopeng Gou, Siu-Ming Yiu, and Longtao He. 2018. MaMPF: Encrypted Traffic Classification Based on Multi-Attribute Markov Probability Fingerprints. *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)* (2018), 1–10. https://ieeexplore.ieee.org/document/8624124

[8] Liming Liu, Ruoyu Li, Qing Li, Meija Hou, Yong Jiang, and Mingwei Xu. 2025. FlowletFormer: Network Behavioral Semantic Aware Pre-training Model for Traffic Classification. (2025). https://doi.org/10.48550/arXiv.2508.19924

[9] Mohammad Lotfollahi, Mahdi Jafari Siavoshani, Ramin Shirali Hossein Zade, and Mohammdsadegh Saberian. 2017. Deep packet: a novel approach for encrypted traffic classification using deep learning. *Soft Computing* 24 (2017), 1999 – 2012. https://api.semanticscholar.org/CorpusID:35187639

[10] Oliver Michel, Satadal Sengupta, Hyojoon Kim, Ravi Netravali, and Jennifer Rexford. 2022. Enabling passive measurement of zoom performance in production networks. *IMC '22: Proceedings of the 22nd ACM Internet Measurement Conference* (2022), 244–260. https://dl.acm.org/doi/10.1145/3517745.3561414

[11] Eva Papadogiannaki and Sotiris Ioannidis. 2021. A Survey on Encrypted Network Traffic Analysis Applications, Techniques, and Countermeasures. *ACM Computing Surveys (CSUR), Volume 54, Issue 6* (2021), 1–35. https://doi.org/10.1145/3457904