Explaining Deep Learning Models for Per-packet Encrypted Network Traffic Classification

Luis Garcia, Genevieve Bartlett, Srivatsan Ravi, Harun Ibrahim, Wes Hardaker, Erik Kline
University of Southern California Information Sciences Institute
Marina Del Rey, CA, USA
lgarcia@isi.edu, bartlett@isi.edu, sravi@isi.edu, ibrahim@isi.edu, hardaker@isi.edu, kline@isi.edu

2022 IEEE International Symposium on Measurements & Networking (M&N) | 978-1-6654-8362-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/MN55117.2022.9887744

***Abstract*—Machine learning is increasingly applied to network traffic analysis to aid in tasks such as quality of service management, trend monitoring, and security. Recent advances in deep learning have enabled not only the classification of encrypted transits, but classification on a per-packet level. End-to-end deep learning models are becoming increasingly ubiquitous given their ease of use, i.e., developers do not need to engineer features, and their apparent versatility. However, deep learning entails black-box models that hinder the capability to debug and explain classifications. Moreover, the computational complexity of deep learning can incur unnecessary latency, which is problematic for real-time classification needs. In this paper, we propose a methodology to interpret black-box, deep learning-based encrypted network traffic classification models, with an attempt to understand the dominant features a classifier is focusing on for a given task. We evaluate our approach on state-of-the-art deep learning classification techniques for encrypted per-packet classification and demonstrate how interpretability can be used to debug and improve the training pipeline while significantly reducing the size of the deep learning model. We propose future directions toward optimizing model performance while maintaining explainability.**

***Index Terms*—Encrypted Network Traffic Classification, Interpretable Machine Learning**

I. Introduction

Traffic classification is vital for a multitude of network management tasks, including quality of service, security, and trend monitoring. Over the last two decades, a large body of work focused on leveraging machine learning algorithms based on human-engineered features to classify encrypted transits [1]. However, as traffic patterns and applications become more complex, researchers have leveraged recent advances in deep learning to alleviate the developer burden for feature extraction and classification model architecture. In particular, researchers leveraged deep learning to classify transits on a per-packet basis [2]—rather than looking at features extracted from the entire flow. Although deep learning models provide developers with a powerful tool for robust classification, they come with the caveat of black-box decision-making.

Deep learning excels at tasks where the relevant features are poorly understood. On the one hand, deep learning models tend to be more robust to noisy data than models based on human-engineered features. On the other hand, the black-box nature of deep learning models limits the developer's capabilities to debug and intervene when a model is consistently misclassifying easy-to-disambiguate cases. For instance, the per-packet deep learning model we study performs well overall for a given flow for both traffic type classification and application type classification. However, our evaluation of the model shows that the model tends to misclassify certain instances of e-mail and chat applications as file transfer applications. Moreover, the model struggles to transfer to datasets with different characteristics than the training data [3]. Finally, the associated models for the per-packet classification of encrypted transits include the encrypted payload in their input—implying that the models can find patterns in the encryption. If developers could understand the model's decision-making, they could prove that the model cannot detect patterns in the encrypted payloads and then be able to streamline the model to focus on the packet preamble.

The explainability of deep learning models is an emerging area of research that has thus far focused on visual, text, audio, and sensor data domains [4]. For models that are not inherently interpretable, understanding deep learning models involves post-hoc explanations where the attributions of a given sample's features relative to its output classification are overlaid on a sample visualization [5]. Thus, the explanation hinges on the ability to both effectively generate an input's feature attributions as well as visualize the sample efficiently. For per-packet encrypted traffic classification, overlaying attributions could provide intuitions as to what features a model focuses on and give intuitions to the aforementioned debugging and intervention challenges.

In this paper, we propose a methodology to interpret deep learning classification to understand and debug classifications in the context of per-packet encrypted network traffic classification. In particular, we aim to interpret decision-making for end-to-end black-box models that are trained to classify unstructured network traffic information, i.e., a sequence of packet bytes. We evaluate input attribution techniques to understand classifications and misclassifications for pre-trained, per-packet encrypted traffic classifications. Finally, we show how input attribution can streamline deep learning model tasks to be more computationally efficient. Our results demonstrate a mitigation technique for a cautionary use case where initial assumptions about the input preprocessing were flawed.

The rest of the paper is organized as follows. We describe the preliminary information and related work for this paper in

Section II. We then describe our methodology and evaluation results in Section III, and conclude with discussing how we plan to build on this work in Section IV.

II. Preliminaries

This section presents the background and related works motivating the need for explainable encrypted network traffic classification. We first present a brief overview of methods for machine learning explainability and interpretability. We then present a brief overview of machine learning techniques for encrypted network traffic classification and highlight the limitations of black-box classifiers. Finally, we describe related works in explainable AI for network traffic classification.

A. Explainable Machine Learning

Explaining or interpreting machine learning model decision-making is critical for trustworthy autonomy in network management. We first describe models that are inherently interpretable along with their limitations. We then describe how black-box deep learning model decisions can be interpreted. We also briefly discuss consolidating the tradeoffs in explainability and performance of end-to-end deep learning models.

Inherently explainable models. In this paper, "inherently interpretable" models refer to any models with a symbolic interface feeding into a reasoning layer. The reasoning layer may be composed of first-order principles or a sub-symbolic reasoning layer, as in neural-symbolic computation [6]. A common approach is for an expert to craft a set of human-understandable symbols that represent features that can be extracted from training data. These features can then be fed into standard machine learning training algorithms. One drawback of these approaches is that they require an expert to craft the necessary and sufficient set of features for a domain-specific problem. Moreover, the understandability of a model hinges on the understandability of both the features and the reasoning framework. Recent works have explored concept-bottleneck models, where models are forced to extract and identify concepts through multi-task learning automatically [7], [4], [8]. However, bridging the gap between explainability and end-to-end learning frameworks and going from deep learning models to inherently interpretable models are open research challenges.

Post-hoc explanations for black-box models. Post-hoc explanations of pre-trained black-box deep learning model decisions often entail visualizing what aspect of a sample input or learned feature caused a particular prediction. A large body of research has emerged presenting different methods to either attribute input features for a given neural network prediction, visualize what latent features a neural network has learned, or provide an explanation by examples from the training data, and we refer the reader to machine learning interpretability surveys for a comprehensive overview [9]. And although post-hoc explanations usually target images, recent works have shown that post-hoc methods can be generalized to other modalities such as audio, text, and sensor data, as long as the samples can be visualized [4]. The process entails the additional step of visualizing the data before overlaying attributions.

In this work, because we are interested in debugging neural network misclassifications of an encrypted packet, we explore attribution methods for a given input. In particular, we explore utilizing saliency maps for post-hoc explanations, which were initially designed to visualize the gradients of a class score for an input image [10]. Because the deep learning models under scrutiny treat each packet as an "image" of bytes, where each byte is analogous to a pixel, we similarly adopt saliency maps for pixel attribution. Generally, the vanilla gradient saliency map model works in three steps: (1) perform a forward pass on the model with a given input; (2) compute the gradient of a class score with respect to the given input; and (3) visualize the gradient over the input image to highlight the attributions [9]. The gradient of the class score, E_{grad}, for a given input, X_0, is calculated as follows:

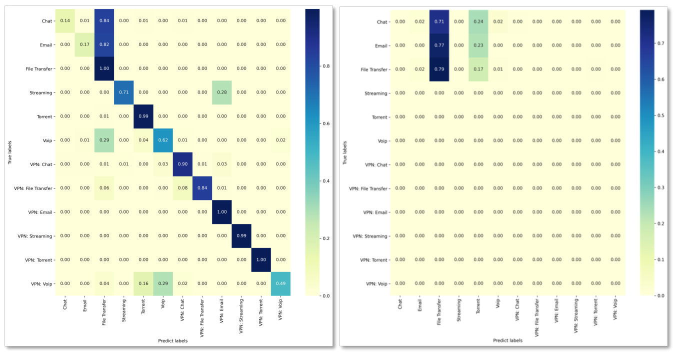
$$E_{grad}(X_0) = \frac{\delta S_c}{\delta X}|_{X=X_0} \tag{1}$$

where $S_C(X)$ is the output of the neural network output for an input X and a class C. In this paper, we will first show that we can visualize an input packet or traffic flow before overlaying a model's attributions for a given class and input packet instance.

B. Machine Learning for Encrypted Network Traffic

We broadly categorize prior approaches into models that reason about human-engineered features across entire encrypted transit flows versus deep learning-based methods that defer feature extraction to deep learning models. We do not seek to survey and taxonomize all machine learning approaches for encrypted network traffic classification as prior works have already done so [1], [11]. Instead, we aim to present popular approaches relative to how explainable their decision-making is to developers.

Models Based on Expert-crafted Features. Most feature-based approaches leverage expert-crafted tools to perform the feature extraction, such as CICFlowMeter [3]. For a given packet capture, CICFlowMeter extracts around 80 statistical network traffic features, such as flow duration and packet lengths. Once features are extracted, they are typically fed into a decision-tree learning framework to generate an inherently interpretable model. However, decision trees are only deemed interpretable if they are short, i.e., a complex decision tree provides little intuition about decision-making [9]. Upon replicating common methods on the same datasets from [3] using CICFlowMeter for feature extraction along with a popular open-source decision tree toolbox [12], we found that the generated decision-tree rules were often very complex trees correlated to the number of features (80 in this case) multiplied by the number of classes (16 for traffic classification tasks in the VPN/Non-VPN dataset [3]). Since these approaches were designed to leverage expert-crafted features, they were not focused on interpretability, but working towards interpretability enables debugging and accountability for model bias and misclassifications.



(a) VPN/Non-VPN dataset [3] (b) Custom minimega dataset [13]

Fig. 1. Evaluating an off-the-shelf deep learning per-packet traffic classification model [2] on baseline dataset as well as a test dataset from a different distribution consisting of only file transfer, e-mail, and chat applications. The model is not reliable in both cases, and is even more confused with the out-of-distribution dataset.

End-to-end Deep Learning-based Approaches. On the other hand, black-box deep learning models have been shown to outperform decision-tree based approaches for classification tasks [14], [2]. In all cases, the VPN/Non-VPN dataset [3] is commonly used as a benchmark for both traffic type classification and application type classification since it offers both the VPN and non-VPN versions of encrypted traffic for popular applications. A recent survey taxonomized the deep learning approaches, and highlighted the differences in pre-processing approaches [11]. Most approaches focused on header information, payload data, or both, while masking features that may cause overfitting, e.g., any I.P. address information. This paper focuses on one of the more prominent approaches that performed per-packet classification based on header and payload data. We focus on the per-packet deep learning classification model due to the broad implications of classifying entire traffic flows based on a single packet. Models that can classify an entire flow on a single packet are highly efficient, but can also be highly sensitive to misclassifications.

C. Related Work in XAI for Network Traffic Classification

Recent works have explored explainable AI (XAI) for network traffic classification in different contexts. Ahn et al. [15] explore interpretability techniques on models trained on human-engineered features for traffic flows to quantify the importance of each feature before using genetic algorithms to select important features. Callegari et al. [16] maintain explainability by exploring interpretable reasoning rules based on human-engineered features. Similarly, Chowdhury et al. [17] aim to identify human-engineered "super features" that incorporate statistical information about traffic flows, and use another method of feature attribution, i.e., Shapley values [18], to quantify feature importance. Nascita et al. [19] also use Shapley values to quantify feature importance at the global interpretation level rather than sample-based interpretation to evaluate the trustworthiness of a model. Morichetta et al. [20] explore XAI techniques for traffic classification in unsupervised settings, where clustering techniques are used to extract classes that experts vet. All these works aim to explain a model's reasoning about human-interpretable features, where we are exploring interpreting end-to-end black-box models that reason about unstructured data, i.e., a sequence of packet bytes. Given that black-box methods are becoming increasingly ubiquitous due to their versatility and lack of human effort needed to engineer features, we need to support methods to interpret, debug, and intervene in models.

III. Intrepreting Deep Learning Models for Per-Packet Classification

This section provides an overview of the experimental setup for the per-packet deep learning model while highlighting the issues we wish to interpret. We then show how interpretability can aid in debugging two use cases: traffic type classification and application classification.

Use case: Per-packet Classification. We leveraged an open-source implementation of the per-packet deep learning classification model [2] with the same convolutional neural network architecture[1]. A pre-processing stage transforms each packet to avoid overfitting by removing the ethernet header, masking the IP address, padding UDP headers, removing irrelevant packets, and either padding or truncating each packet to ensure a length of 1500 bytes. We focus on the two more prominent use cases in literature: traffic classification and application classification using the VPN/Non-VPN dataset.

Performance metrics. We will compare model performance based on precision, recall, and F1-score, i.e.,

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

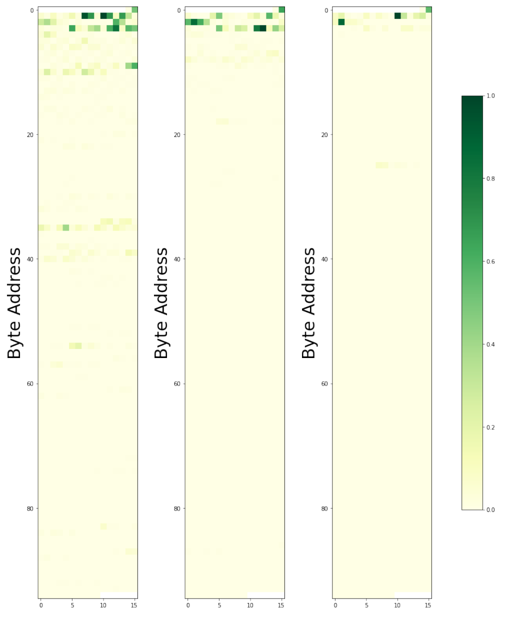
$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

where TP, FP, and FN are true-positive, false-positives, and false-negatives, respectively. We will also provide qualitative comparisons by visualizing the confusion matrix of the models across all classes. Finally, since we expect a significant size reduction in the models without payload information, we will compare the size and speed of the models. For speed, we compare the average speed relative to CPU time across all inferences for the same dataset.

A. Use case 1: Interpreting Traffic classification

Figure 1-A shows the replicated results when an off-the-shelf model is evaluated on the dataset it was trained on (VPN/Non-VPN dataset). The model generally performs well when validated on the VPN/Non-VPN dataset, but tends to confuse e-mail and chat traffic as file transfer traffic. We subsequently generated chat, e-mail, and file transfer traffic

[1]Although the authors did not open-source their code, an open-source, third-party re-implementation of the model with the same hyperparameters can be found at https://github.com/munhouiani/Deep-Packet



(a) Chat misclassified as file transfer (b) E-mail misclassified as file transfer (c) File transfer classified as file transfer

Fig. 2. Average attributions for per-packet traffic type classification across entire flow.

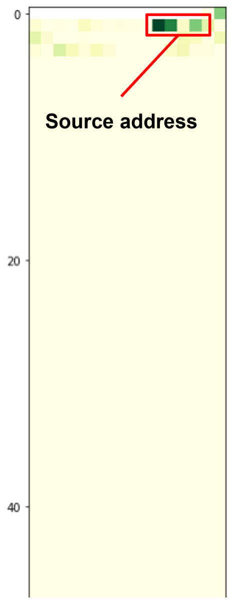
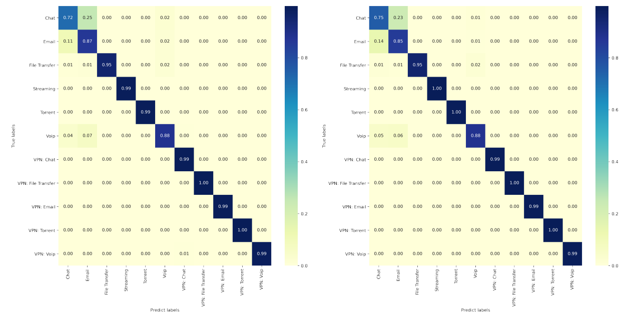


Fig. 3. Specific example for a single packet mis-classification with high confidence (Email as FTP).

using minimega [13] and found the model has similar results–depicted in Figure 1-B. Thus, we leverage interpretability to understand the underlying reasons behind the misclassifications.

Misclassification attribution. To understand the behavior of the per-packet traffic classification model, we leverage an open-source interpretability toolbox [21] to implement saliency attribution for post-hoc explanations. We first analyzed the average attributions across all packets for a given



(a) Trained with payload (Packet length = 1500) bytes (b) Trained without payload (Packet length = 60 bytes)

Fig. 4. Comparing retrained implementations of the per-packet deep learning model for traffic type classification [2]. The original proposed architecture considered an entire encrypted traffic (1500 bytes). Our model trained on only header data (60 bytes). The smaller model has competitive performance while being scaled down an order of magnitude in size and latency.

flow to qualitatively determine any patterns in the attributions. Then, since post-hoc explanations require a means to visualize the input before overlaying attributions, we visualize packets in the same format as packet-inspection tools such as WireShark and Scapy: a pane of hexadecimal bytes with rows sixteen bytes long.

Figure 2 shows the average attributions for three cases: (a) chat traffic misclassified as file transfer, (b) e-mail misclassified as file transfer, and (c) file transfer classified corrected[2]. In all three cases, the strongest attributions resided in the header portion of the packets. Moreover, the average attributions correspond to the entire flow, not attributions for single packet classification.

Figure 3 depicts the attributions for an e-mail misclassified as a file transfer with 99% confidence. Interestingly, the highest activations occur at the location of the IP address. However, the pre-processing step masks the IP address to zeros before training. Intuitively, we can speculate that the model may be learning to identify an e-mail address as a file transfer when a mask exists at this particular location, with a minor combination of other input features. We also notice that the first byte after the removed ethernet header, which represents the end-to-end congestion notification (ECN), shows significant attribution for this classification. Relying on the ECN byte for classification is problematic for a multitude of reasons, e.g., the classification incorporates the network conditions in training which may not be present in operational environments and unless congestion indicators are combined with other features, these features are not drawing on detectors inherent to the application. Future work can explore masking features not intrinsic to applications, though prior work downplays the level of detail required to do so. Moreover, given the hypothesis that the model only focuses on header information, we will evaluate whether we can achieve the same performance on the given task and dataset while removing the packet payload.

[2]Note that the white space at the top represents the removed ethernet header that is excluded during classification.

Label	With Payload			Without Payload		
	Pr	Rc	F1	Pr	Rc	F1
Chat	0.25	0.72	0.37	0.20	0.75	0.32
Email	0.12	0.87	0.21	0.14	0.85	0.23
File Transfer	1.00	0.95	0.98	1.00	0.95	0.98
Streaming	0.97	0.99	0.98	0.98	1.00	0.99
Torrent	0.93	0.99	0.96	0.93	1.00	0.96
Voip	0.97	0.88	0.92	0.97	0.88	0.93
VPN: Chat	0.57	0.99	0.72	0.79	0.99	0.88
VPN: FileTransfer	0.99	1.00	1.00	1.00	1.00	1.00
VPN: Email	0.89	0.99	0.94	0.98	0.99	0.98
VPN: Torrent	1.00	1.00	1.00	1.00	1.00	1.00
VPN: Voip	0.98	0.99	0.98	1.00	0.99	0.99

TABLE I

COMPARING TRAFFIC CLASSIFICATION MODEL PERFORMANCE WHEN TRAINED WITH AND WITHOUT PAYLOAD. IN ALL CASES, THE F1-SCORE PERFORMANCE IS EITHER MAINTAINED OR IMPROVES WHEN THE MODEL IS TRAINED WITHOUT PAYLOAD DATA, I.E., WITH A MODEL REDUCED BY AN ORDER OF MAGNITUDE IN SIZE.

Model	Speed Increase	Model Size Reduction	Average F1-Score
Traffic	**x22.23**	**x11.39**	**x1.02**
Application	**x33.90**	**x19.88**	**x1.04**

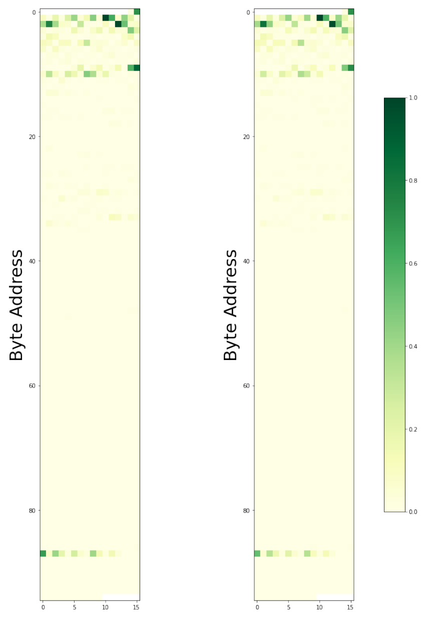
TABLE II

SUMMARIZING THE INCREASE IN SPEED (BASED ON CPU TIME), THE REDUCTION IN SIZE, AND THE IMPACT ON THE AVERAGE F1-SCORE PERFORMANCE WHEN TRAINING A MODEL WITHOUT PAYLOAD INFORMATION.

Retraining model on a subset of input features. Based on the post-hoc explanation results, we hypothesize that retraining the per-packet classification model on only pre-processed header information will perform similarly to a much smaller model. We retrained the model using the same hyperparameters with and without payload data. For the model without payload data, we truncated packets to the first 60 bytes—which corresponds to the max value of a TCP header. Upon retraining with only header information, the model was reduced by an order of magnitude both in size and latency. As depicted in Figure 4 and Table I, we found the smaller model maintained competitive performance with the larger model, even outperforming the model for certain classes. Table II summarizes the reduction in model size, the increase in inference speed, and the average F1-Score. We explore similar hypotheses for application classification.

B. Use case 2: Interpreting Application Classification

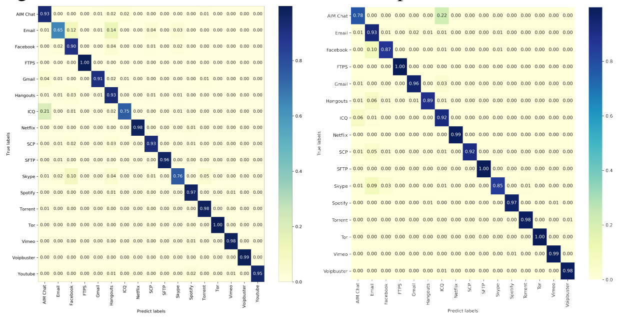
We similarly explored instances where certain flows had high instances of misclassifications for application classification. Figure 5 depicts the average attributions for misclassifications for an e-mail flow. In this case, a significant percentage of the packets were classified as either Hangouts or AIM Chat packets. Although some attributions reside in the payload section of the packet, the strongest activations still reside in the header information. Thus, we retrained the application classification model with only header information. Figure 6 and Table III depict the results comparing the off-the-shelf application classification model to the retrained model with only header information. As in the traffic classification, the



(a) E-mail misclassified as Hangouts

(b) E-mail misclassified as file AIM Chat

Fig. 5. Average attributions for per-packet application classification across entire flows. In this case, we analyzed the classes that were causing most confusion for the same e-mail flow. Similar to the traffic classification, the strongest attributions occur in the header of the packets.



(a) Off-the-shelf model trained with payload

(b) Trained without payload (Packet length = 60 bytes)

Fig. 6. Comparing off-the-shelf model to retrained implementation of the per-packet deep learning model for application classification [2] using only the header data.

model was reduced by an order of magnitude while maintaining competitive performance. The reduction in model size and increase in inference speed are summarized in Table II.

IV. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we leveraged post-hoc explanation methods to understand and debug black-box deep learning models in the context of encrypted network traffic classification. In particular, we focused on per-packet traffic classification. Our results found that, as expected, the models only attributed activations to header information rather than encrypted payloads. We demonstrated how this information could be used to streamline the model in terms of accuracy and performance. More importantly, our results highlighted that ignoring inherent structure

Label	With Payload			Without Payload		
	Pr	Rc	F1	Pr	Rc	F1
AIM Chat	0.02	0.41	0.04	0.04	0.78	0.08
Email	0.07	0.92	0.12	0.07	0.93	0.12
Facebook	0.94	0.83	0.88	0.94	0.87	0.90
FTPS	1.00	0.99	1.00	1.00	1.00	1.00
Gmail	0.17	0.94	0.29	0.19	0.96	0.32
Hangouts	1.00	0.88	0.94	1.00	0.89	0.94
ICQ	0.02	0.93	0.04	0.04	0.92	0.07
Netflix	0.97	0.98	0.98	0.98	0.99	0.98
SCP	0.95	0.92	0.93	0.94	0.92	0.93
SFTP	0.98	0.99	0.98	0.99	1.00	0.99
Skype	0.94	0.83	0.88	0.99	0.85	0.91
Spotify	0.37	0.97	0.53	0.58	0.97	0.73
Torrent	0.40	0.98	0.57	0.53	0.98	0.69
Tor	0.99	1.00	0.99	1.00	1.00	1.00
Vimeo	0.96	0.97	0.97	0.93	0.99	0.96
Voipbuster	1.00	0.98	0.99	0.99	0.98	0.99

TABLE III

COMPARING APPLICATION CLASSIFICATION MODEL PERFORMANCE WHEN TRAINED WITH AND WITHOUT PAYLOAD. IN MOST CASES, THE F1-SCORE PERFORMANCE IS EITHER MAINTAINED OR IMPROVES WHEN THE MODEL IS TRAINED WITHOUT PAYLOAD DATA, I.E., WITH A MODEL REDUCED BY AN ORDER OF MAGNITUDE IN SIZE.

while performing traffic classification can lead to relying on fields not intrinsic to the applications.

As efforts in machine learning are increasingly applied to network environments, a greater focus will be on speed and representation cost, and not just model accuracy (e.g. [22]). Our next steps are focused on exploring multi-task learning using Concept Bottleneck Models (CBMs) [7]. These models, traditionally, use labeled concept data to first predict concepts (e.g. "streaming traffic requires bulk transfer") to then predict a label (e.g. "this flow is video streaming"). Through our post-hoc approach, we can instead interpret unrestricted deep-learning models without such labeled concept data. This results in an inherently interpretable model while maintaining performance and faithfulness of an end-to-end deep learning model.

V. ACKNOWLEDGEMENT

This work is supported by DARPA through W911NF19C0058. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government

REFERENCES

[1] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355–374, 2015.

[2] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.

[3] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related," in *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, 2016, pp. 407–414.

[4] J. V. Jeyakumar, J. Noor, Y.-H. Cheng, L. Garcia, and M. Srivastava, "How can i explain this to you? an empirical study of deep neural network explanation methods," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4211–4222, 2020.

[5] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[6] A. d. Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran, "Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning," *arXiv preprint arXiv:1905.06088*, 2019.

[7] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5338–5348.

[8] J. Huang, A. Mishra, B.-C. Kwon, and C. Bryan, "Conceptexplainer: Understanding the mental model of deep learning algorithms via interactive concept-based explanations," *arXiv preprint arXiv:2204.01888*, 2022.

[9] C. Molnar, *Interpretable machine learning*. Lulu. com, 2020.

[10] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[11] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE communications magazine*, vol. 57, no. 5, pp. 76–81, 2019.

[12] S. I. Serengil and Y. K. Teknoloji, "Chefboost: Alightweight boosted decision tree framework," 2021.

[13] D. J. Fritz, "Minimega." Sandia National Lab.(SNL-CA), Livermore, CA (United States), Tech. Rep., 2016.

[14] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.

[15] S. Ahn, J. Kim, S. Y. Park, and S. Cho, "Explaining deep learning-based traffic classification using a genetic algorithm," *IEEE Access*, vol. 9, pp. 4738–4751, 2020.

[16] C. Callegari, P. Ducange, M. Fazzolari, and M. Vecchio, "Explainable internet traffic classification," *Applied Sciences*, vol. 11, no. 10, p. 4697, 2021.

[17] S. Chowdhury, B. Liang, and A. Tizghadam, "Explaining class-of-service oriented network traffic classification with superfeatures," in *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*, 2019, pp. 29–34.

[18] E. Kalai and D. Samet, "On weighted shapley values," *International journal of game theory*, vol. 16, no. 3, pp. 205–222, 1987.

[19] A. Nascita, A. Montieri, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "Xai meets mobile traffic classification: Understanding and improving multimodal deep learning architectures," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4225–4246, 2021.

[20] A. Morichetta, P. Casas, and M. Mellia, "Explain-it: Towards explainable ai for unsupervised network traffic analysis," in *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*, 2019, pp. 22–28.

[21] N. Kokhlikyan, V. Miglani, M. Martin, E. Wang, B. Alsallakh, J. Reynolds, A. Melnikov, N. Kliushkina, C. Araya, S. Yan *et al.*, "Captum: A unified and generic model interpretability library for pytorch," *arXiv preprint arXiv:2009.07896*, 2020.

[22] F. Bronzino, P. Schmitt, S. Ayoubi, H. Kim, R. Teixeira, and N. Feamster, "Traffic refinery: Cost-aware data representation for machine learning on network traffic," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 5, no. 3, pp. 1–24, 2021.

[23] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, vol. 2, no. 11, p. e7, 2017.

[24] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[25] C. Beliard, A. Finamore, and D. Rossi, "Opening the deep pandora box: Explainable traffic classification," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 1292–1293.