Passive Encrypted IoT Device Fingerprinting with Persistent Homology

Joseph R. Collins
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250
jc17@umbc.edu

Michaela Iorga
National Institute of Standards and Technology
Gathersburg, MD 20899
michaela.iorga@nist.gov

Dmitry Cousin
National Institute of Standards and Technology
Gathersburg, MD 20899
dmitry.cousin@nist.gov

David Chapman
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250
dchapm2@umbc.edu

Abstract

Internet of things (IoT) devices are becoming increasingly prevalent. These devices can improve quality of life, but often present significant security risks to end users. In this work we present a novel persistent homology based method for the fingerprinting of IoT traffic. Traditional passive device fingerprinting methods directly inspect the packet attributes or contents within the captured traffic. But techniques to fingerprint devices based on inter-packet arrival time (IAT) are an important area of research, as this feature is available even in encrypted traffic. We demonstrate that Topological Data Analysis (TDA) using persistent homology over IAT packet windows is a viable approach to obtain discriminative features for device fingerprinting. The clique complex construction and weighting function we present are efficient to compute and robust to shifts of the packet window. The 1-dimensional homology is calculated over the resulting filtered clique complex. We obtain competitive accuracy of 95.34% on the UNSW IoT dataset [1] by using a convolutional neural network to classify over the corresponding persistence images [2].

Topological Data Analysis and Beyond Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

1 Introduction

There are an estimated 20 billion IoT devices in active use, up from 8 billion just in 2017 [3, 4], yet many vulnerabilities have been discovered in IoT devices which put users security and privacy at risk. Device fingerprinting is the problem of uniquely identifying devices on a network [5]. Methods to fingerprint devices are valuable both for discovering new attacks as well as improving countermeasures. *Passive* device fingerprinting is when a listener attempts to identify the devices without initiating or responding to communications. A particularly important variant of *passive* fingerprinting involves identifying encrypted traffic, in which the packet contents cannot be analyzed and only meta-information is available. Inter-packet Arrival Time (IAT) is known to be a valuable information source for traffic classification [5–7], but feature extraction from IAT remains challenging.

We present a generalized persistent homology based approach to network traffic classification for IoT device fingerprinting. Our approach operates over fixed size packet windows of network traffic. The clique complex constructed over the IAT of the packets in this window has a fixed structure, allowing for fast computation of the filtration. We assume a fixed window size of k, but extension to arbitrary and variable window sizes is straight-forward. Our results using the UNSW IoT data set [1] are competitive with state-of-the-art methods that use only a single feature classifier [8, 9].

Related Work The problem of network traffic classification has wide applicability and is the subject of active study. In this work, we are concerned with classifying traffic solely based on packet timing information. This information has been shown to be a useful feature for several traffic classification tasks and fingerprinting particularly involving encrypted traffic [5–7]. This approach as been applied to fingerprint devices across a wide variety of protocols including 802.11 frames [10], and recently extended to fingerprint IoT devices over the ZigBee and Z-wave protocols [11].

There are several recent works on encrypted passive IoT traffic classification. Ortiz et al. apply recurrent neural networks to the encrypted payload data of TCP flows [3]. A TCP flow is a sequence of TCP packets between two devices, analogous to a well defined conversation between the devices. Sivanathan et al. applies a bag-of-words approach over high level attributes of packets within flows [9]. Pinheiro et al. utilizes a time binned packet length statistics with strong results [8]. IAT based approaches may be considered a complimentary feature space, because IAT is fundamentally not a single packet attribute but rather a difference in timing between a pair of packets that can be naturally encoded as an edge weight of a graph.

Historically, TDA has seldom been applied to network traffic, although recently this subject is starting to gain greater attention. A notable work is that of Bruillard et al. which focuses on anomaly detection [12]. Also, Gabdrakhmanova used Betti numbers and Euler's characteristics to predict network traffic volume [13]. Perhaps the most similar work to ours is Postol et al. which uses a sliding window embedding over IoT traffic time-series data. The persistent homology of the resulting Vietoris-Rips complex is used for traffic classification over relatively long time spans. Our approach differs in two significant ways. First, our construction works well over small packet windows. Second, we obviate the need for the expensive calculation of the Vietoris-Rips complex by utilizing an efficient fixed clique complex construction.

2 Methods

In this section we define the clique complex construction and weighting function which enable the application of persistent homology to network traffic data. Let the packets of a given flow be $(p_0, p_1, ..., p_n)$. The arrival time of a packet is given by $T(p_q)$. The IAT is defined as $\Delta_T(p_q) = T(p_q) - T(p_{q-1})$. We classify over windows of the flow, given by $\omega_k(i) = (p_i, p_{i+1}, ..., p_{i+k-1})$.

Clique Complex Construction For such a window, we construct a clique complex as follows, using graph notation for simplicity. The vertex set is given by $V(\omega_k(i)) = \{j \forall j \in \{i, i+1, ..., i+k-1\}\} \cup \{V_k(i)\}$, where $V_k(i) = k + i$ is the flow vertex. The vertices representing packets will be connected to the flow vertex in ascending order of IAT. The edge set is $E(\omega_k(i)) = \{(j, j+1) \forall j \in \{i, i+1, ..., i+k-2\}\} \cup \{(j, V_k(i)) \forall j \in \{i, i+1, ..., i+k-1\}\}$. The first term contains the edges between vertices of sequential packets, and the second term contains the edges connected to the flow vertex, $V_k(i)$.

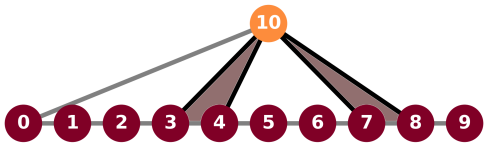

Figure 1: An example filtered clique complex of the window $\omega_{10}(0)$ after the first 4 non-zero edges have been added. Packet vertices are in red and the flow vertex is in orange. Dark black edges are non-zero edges that have been added. Gray edges are weight 0 and added at the initial step. The shaded regions represent the 2-simplices.

Weighting Function The weighting function over the edges is constructed in terms of packet IAT. All vertices are added at step 0. The edges connecting the sequential packet vertices form the basic structure of the filtration and are also added at step 0. Formally, the weighting function over the edges is given by

$$W_{k,i}(a,b) = \begin{cases} 0 & (a,b) \in \{(j,j+1) \forall j \in \{i, i+1, ..., i+k-2\}\} \cup (i, V_k(i)) \\ \Delta_T(p_a) & (a,b) \in (j, V_k(i)) \forall j \in \{i+1, i+k-1\} \end{cases}$$

This is to say that the packet vertices are connected to the flow vertex with an edge weighted by IAT. As there is no packet before p_i, we define $\Delta_T(p_i) = 0$, meaning the weight of the edge from the first packet vertex, i, to $V_k(i)$ is always 0. Figure 1 shows an example step in such a filtration.

Filtration Construction We are concerned with the 1-dimensional persistent homology of the constructed complex, so it is only required to add up-to and including 2-simplices/3-cliques to the filtration. In order to form the filtration of the clique complex, we first form the complex of all vertices and 0 weight edges at step 0. Then, it is only necessary to add the weighted edges in ascending order. By checking at each step, after each edge is added, if the adjacent packet vertices have also have an edge going to V_k, it is possible to tell when 2-simplices join the filtration. If such an edge exists, then the new edge forms a new 2-simplex. This means that given a a list of sorted edges, it is only an $O(n)$ operation to construct the filtration.

Persistence Calculations For these calculations we utilize the tools in Scikit-TDA (CechMate and Persim) [15]. The 1-dimensional persistence of the filtered clique complex is found using CechMate. The resulting diagrams are transformed to persistence images using Persim. We generate persistence images of size 128 by 128, which was experimentally chosen to minimize computer memory usage while retaining important features. The underlying persistence surface is generated from the source diagram using a Gaussian with standard deviation of 1.0 and a linear weighting function. The persistence images are re-scaled $0 - 255$ in order to reduce memory requirements (unsigned byte vs. orignal float) and as a normalization measure before passing the images to a downstream model.

Ortiz et al. stores a 0 padded array of 1500 bytes containing the payload of each packet in the time window [3]. For the window of size 25 that the authors use, the resulting descriptor is $37,500$ bytes. Over the same length window, our approach produces a single $128^2 = 16,384$ byte persistence image, less than half the size. Furthermore, the size of the window does not affect the size of the persistence image, and the two values can vary independently. If memory is a critical factor, the size of the persistence images can be reduced.

3 Experiments and Results

As we have presented, previous work has demonstrated that timing information is a powerful discriminative feature for network traffic. We aim to prove, by the accuracy of the classification results, that our construction captures this timing information in a robust way. Figure 2 shows that resulting persistence images do not change significantly given shifts in the underlying traffic.

Data The UNSW IoT data set [1] is used in this work. The publicly available data consists of 20 days of labeled data for 23 IoT devices. As with related work on this data set, we are concerned with accurately classifying the TCP flows of IoT devices.

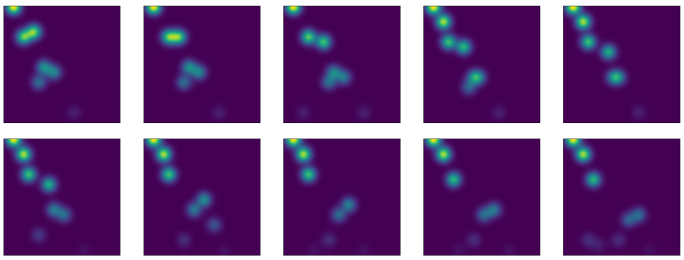

Figure 2: The persistence images of the windows corresponding to $\omega_{25}(i), 0 \leq i < 10$. Note that as the start of the window moves forward one packet at a time, the persistence image only changes slightly (left to right, top to bottom).

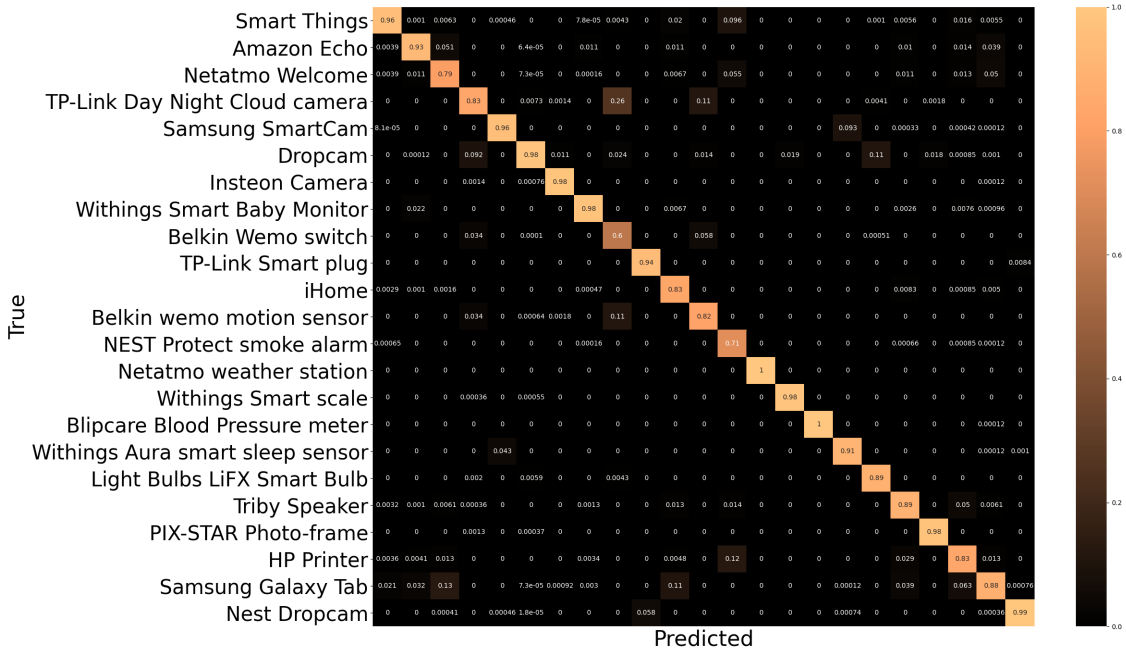

Figure 3: The confusion matrix of the VGG19 results normalized along the columns to adjust for the class imbalance in the data set.

Experiments Following the work of Ortiz et al., we choose $k = 25$ as the window size [3]. For each TCP flow, windows are taken sequentially with no overlap. We extract $630,842$ such TCP flow windows from the dataset.

Table 1: Results

Model	Accuracy	Recall	Precision
PH-VGG19	95.34%	95.27 %	95.46%

Using a randomized class balanced 70/30 train/test split over the persistence images for all 23 devices, a VGG19 [16] model is trained to classify the persistence images. We use the standard VGG19 architecture with two modifications. First, the input layer is reshaped to accept images of size 128 by 128. Second, the output layer is changed to output a softmax 23-dimensional vector, as we utilize a one-hot-encoding for device labels. We do not attempt transfer learning from weights trained on existing image datasets, given the unique nature of persistence images. A batch size of 64 is used for training. A larger batch size should be utilized if hardware permits to increase the speed of training, especially given the large size of the dataset.

Results and Comparison Our results over the test set are shown in Table 1. These results are competitive with other results which only utilize a single traffic feature. Sivanathan et al. obtains 92.13% accuracy classifying hourly flows using a bag-of-words approach over port numbers [9]. Pinheiro et al. obtains an accuracy of 96% classifying 1 second windows using packet length statistics [8]. As such, our approach is comparable with state of the art results on this dataset. These results validate that persistent homology is a viable approach for this task.

4 Conclusion

We have demonstrated a persistent homology based IAT feature extraction technique that is useful for passive fingerprinting of encrypted IoT device traffic. This approach achieves highly competitive accuracy on the UNSW dataset with state of the art single feature classifiers. The presented clique complex construction and weighting function have been shown to be useful for tasks relating to the classification of network traffic. We plan to expand this work to several additional public datasets. We hope that TDA will become a more widely employed technique for a variety of related network traffic analysis problems in the near future.

References

[1] Ayyoob Hamza, Hassan Habibi Gharakheili, Theophilus A Benson, and Vijay Sivaraman. Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity. In *Proceedings of the 2019 ACM Symposium on SDN Research*, pages 36–48, 2019.

[2] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1):218–252, 2017.

[3] Jorge Ortiz, Catherine Crawford, and Franck Le. Devicemien: network device behavior modeling for identifying unknown iot devices. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, pages 106–117, 2019.

[4] Mark Hung. Leading the iot: Gartner insights on how to lead in a connected world, 2020. URL https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf.

[5] A Selcuk Uluagac, Sakthi V Radhakrishnan, Cherita Corbett, Antony Baca, and Raheem Beyah. A passive technique for fingerprinting wireless devices with wired-side observations. In *2013 IEEE conference on communications and network security (CNS)*, pages 305–313. IEEE, 2013.

[6] Mohamad Jaber, Roberto G Cascella, and Chadi Barakat. Can we trust the inter-packet time for traffic classification? In *2011 IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE, 2011.

[7] Ke Gao, Cherita Corbett, and Raheem Beyah. A passive approach to wireless device fingerprinting. In *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*, pages 383–392. IEEE, 2010.

[8] Antônio J Pinheiro, Jeandro de M Bezerra, Caio AP Burgardt, and Divanilson R Campelo. Identifying iot devices and events based on packet length from encrypted traffic. *Computer Communications*, 144:8–17, 2019.

[9] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Classifying iot devices in smart environments using network traffic characteristics. *IEEE Transactions on Mobile Computing*, 18(8):1745–1759, 2018.

[10] Chao Shen, Ruiyuan Lu, Saeid Samizade, and Liang He. Passive fingerprinting for wireless devices: A multi-level decision approach. In *2017 IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pages 1–6. IEEE, 2017.

[11] Leonardo Babun, Hidayet Aksu, Lucas Ryan, Kemal Akkaya, Elizabeth S Bentley, and A Selcuk Uluagac. Z-iot: Passive device-class fingerprinting of zigbee and z-wave iot devices. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2020.

[12] Paul Bruillard, Kathleen Nowak, and Emilie Purvine. Anomaly detection using persistent homology. In *2016 Cybersecurity Symposium (CYBERSEC)*, pages 7–12. IEEE, 2016.

[13] N Gabdrakhmanova. Construction a neural-net model of network traffic using the topologic analysis of its time series complexity. *Procedia Computer Science*, 150:616–621, 2019.

[14] Michael Postol, Candace Diaz, Robert Simon, and Drew Wicke. Time-series data analysis for classification of noisy and incomplete internet-of-things datasets. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1543–1550. IEEE, 2019.

[15] Chris Tralie Nathaniel Saul. Scikit-tda: Topological data analysis for python, 2019. URL https://doi.org/10.5281/zenodo.2533369.

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.