MSc. in Computing

Practicum Approval Form

Section 1: Student Details

Project Title	Machine Learning-Aided Side Channel Attacks on Network Traces
Student 1	
ID	22261972
Name	Daborah Djon
Email	deborah.djon3@mail.dcu.ie
Major	Artificial Intelligence
Student2	
ID	13469978
Name	Darragh Connaughton
Email	darragh.connaughton5@mail.dcu.ie
Major	Secure Programming
Supervisor	Geoff Hamilton
Date of Submission	11.11.2022

Section 2: About your Practicum

What is the topic of your proposed practicum? (100 words)

Coarse network data, such as the bits per second across a video streaming session leave an identifiable pattern that is stable enough across distinct traces to identify the video given sufficient training data [1]–[3].

We aim to improve on previous studies in the field of video fingerprinting using encrypted network traces in two aspects. First, we want to improve the classifier for video identification. And second, we want to improve on obfuscation methods that can reduce the accuracy of the classifier, specifically seeking to reduce the bandwidth cost incurred by obfuscation.

We attempt to validate the results of the SmartSwitch authors who were able to reduce the accuracy of the classifier to that of a random guess, while incurring 40% less bandwidth overhead [3].

Please provide details of the papers you have read on this topic (details of 5 papers expected). How does your proposal relate to existing work on this topic described in these papers? (200 words)

Virtual Private Networks (VPNs) are advertised as the singular tool required to protect one's privacy online. Such claims fail to appreciate the capability of a skilled adversary wielding modern traffic analysis techniques. Network traffic patterns can be used by an internet service provider, surveillance states, or company to identify critical material on a network, like political or pornographic videos, without needing to see the plaintext data. We will attempt to replicate the result achieved by the video identification methods [1]–[3] and, assuming successful replication, in place, attempt to sufficiently obfuscate the identifiable patterns through the introduction of targeted noise in a manner which retains users Quality of Experience [2]. Introduction of these techniques by service providers would diminish the aforementioned actors ability to profile users based on encrypted communications.

Modern web streaming services utilise Dynamic Adaptive Streaming over HTTP streaming standard, which supports adaptive bitrates, allowing for individual videos to be of varying quality depending on the networking conditions. Videos are encoded into various quality levels and bitrates and segmented across time. The requested party selects the quality of the next segment depending on the network conditions and the current state of the playback buffer. The sequential nature of the streaming standard introduces a distinct traffic pattern that can be used to identify the video [1]–[3].

Papers reviewed captured video traces using some variants of WireShark and extracted features from them using both machine learning and manual efforts. Previous authors have used differential fingerprints, m-order Markov chains, and convolutional neural networks (CNN) or a combination of these. However, there are other methods, such as an LSTM neural network, that could be suited. We aim to evaluate the best classifier architecture [1], [4], [5], [6].

What are the research questions that you will attempt to answer? (200 words)
Main research questions:

1) Attacker (Deborah): Can we improve upon existing identification methods using state of the art Machine Learning techniques?

2) Defender (Darragh): Can we improve upon existing obfuscation techniques to counter Deborah's attack? Can we achieve the same level of obfuscation as previous authors but with decreased bandwidth overhead?

Avenues to we could potentially explore beyond our main research questions:

Does the accuracy of the classification and obfuscation method improve with a larger data set?

- Previous papers use rather small data sets of 50-100 videos with 50-100 network traces each. Could the upper classification limit be increased using more data?
- We could perform an analysis to see which model is optimal given a certain data size.

How well do the experiments fare in a closed-world setting?:

- Each of the experiments operates in a closed world setting where the entire set of Youtube videos a target will view is known in advance, with the assumption that the next viewed

video will be a member of the known set of videos. How does this method operate in an open-world setting?
- With a larger dataset we could use a smaller number of videos for fingerprinting with several network traces (eg. 100 videos, 50 traces each) and a large number of single traces of videos that are used for testing the open world scenario (eg. 500 video traces).

What are the best noise-adding methods?:
- The author's solely explored the d*-privacy noise generation algorithm. We could look into other noise generation algorithms to determine most effective for the given context.

What are the best feature selection methods?:
- Both PFI and statistical methods were used to determine the most significant packets across time. Perhaps we can compare and contrast other Feature Selection methods.

Can we reduce the classification accuracy by making use of the density of significant packets?:
- The SmartSwitch author's found a higher density of significant packets within the first 40 seconds of the video. This aligns with the analysis of DASH for Youtube, which describes an initial playback buffer hydration stage occurring during the first 40 seconds.
    - The initial buffer hydration period is 34 seconds on mobile devices. We could replicate the results of the experiment in a mobile environment to determine if the hydration process leads to the higher density of significant packets. Specifically, we would want to see the window of higher density packets shrink from the first 40 seconds to the first 34 seconds. Such an observation provides evidence for causation.
    - What effect does applying noise solely to the first 40 seconds have on the effectiveness of the classifier and the bandwidth cost incurred (can we achieve good enough obfuscation for fewer network resources)?

What is the best Machine learning classification method?
- Fingerprinting technique must be effective, representative, stale, and reliable [7].
- Can we replicate the accuracy of previous experiments?`
- Could we use features that are not used in the testing in training? E.g., segment size of encoded video prior to encryption?
- Could we reduce the complexity of a CNN to improve the accuracy as in [8]?
- Could other deep learning methods, such as Stacked Denoising Auto-encoder (SDAE) or LSTM be viable options?

Is it possible to create user profiles based on the encrypted network traffic?
- Unicity is the measure to which an individual may be re-identified from anonymised data. Classifying encryption as anonymised data, could it be used to identify an individual on an encrypted network:
- OSINT could be used to build a viewing profile - content makers the target individual is likely to watch. Our method can learn the fingerprint from the most recent of these content makers' videos.

How will you explore these questions? (Please address the following points. Note that three or four sentences on each will suffice.)

- What software and programming environment will you use?
- What coding/development will you do?
- What data will be used for your investigations?
- Is this data currently available, it not, where will it come from?
- What experiments do you expect to run?
- What output do you expect to gather?
- How will the results be evaluated?

We will gather the initial training set using a combination of Selenium to automate the streaming process and PyShark, to capture the network packet captures (PCAP) files. We will capture the bits transferred per time point available. Not conforming to a particular number of bits per second during the initial data capture will allow us to later generate datasets using various window sizes w to see which is most effective. We will use a clustering algorithm, such as K-means clustering, to group bits per seconds into window sizes w. Multiple distinct traces for each video will be captured to account for the bitrate variability, introduced by Variable Bit Rate (VBR), that we'd expect to see in a real network.

We seek to use Deep Learning for the data analysis part of the project as this allows an automatic feature extraction. More concrete, we are considering Sequential Convolutional and Long Short-Term Memory Neural Networks as well as Stacked Demonising Auto-encoders [9], [10]. The literature we have reviewed so far indicates that these Deep Learning methods prove an advantage for website fingerprinting. However, further literature review is required to evaluate whether there are other, maybe newer, Deep Learning techniques that are promising for solving our problem. The main evaluation statistics are the accuracy, error metrics such as the Mean Squared Error (MSE) and the training and classification speed.

As for the programming environment, we aim to use local instances of Jupiter Notebook and Python scripts.Timeline

Goal	**Responsible**	**Deadline**
Code for gathering pcap files ready and collected sample files: 5 traces of 5 videos An additional 10 traces of random videos	Darragh	16.01.2022
Have the code for all the models we will test ready	Deborah	16.01.2022
Present Project Plan	Darragh and Deborah	28.-30.11.2022
Literature review done	Darragh and Deborah	20.02.2022
Finish data collection	Darragh	31.03.2022
Train Machine Learning model	Deborah	30.04.2022
Alterations / improvements to model and Obfuscation	Deborah and Deborah	31.05.2022
Document results	Debprah and Darragh	10.07.2023
Proof read	Deborah and Darragh	20.07.2023

References:

[1] W. Afandi, S. M. A. H. Bukhari, M. U. S. Khan, T. Maqsood, and S. U. Khan, 'Fingerprinting Technique for YouTube Videos Identification in Network Traffic', IEEE Access, vol. 10, pp. 76731–76741, 2022, doi: 10.1109/ACCESS.2022.3192458.

[3] H. Li, B. Niu, and B. Wang, 'SmartSwitch: Efficient Traffic Obfuscation Against Stream Fingerprinting', in Security and Privacy in Communication Networks, vol. 335, N. Park, K. Sun, S. Foresti, K. Butler, and N. Saxena, Eds. Cham: Springer International Publishing, 2020, pp. 255–275. doi: 10.1007/978-3-030-63086-7_15.

[4] L. Yang, S. Fu, Y. Luo, and J. Shi, 'Markov Probability Fingerprints: A Method for Identifying Encrypted Video Traffic', in 2020 16th International Conference on Mobility, Sensing and Networking (msn 2020), Los Alamitos, 2020, pp. 283–290. doi: 10.1109/MSN50589.2020.00055.

[5] M. U. S. Khan, S. M. A. H. Bukhari, T. Maqsood, M. A. B. Fayyaz, D. Dancey, and R. Nawaz, 'SCNN-Attack: A Side-Channel Attack to Identify YouTube Videos in a VPN and Non-VPN Network Traffic', Electronics, vol. 11, no. 3, Art. no. 3, Jan. 2022, doi: 10.3390/electronics11030350.

[6] A. Shusterman et al., 'Robust Website Fingerprinting Through the Cache Occupancy Channel', in 28th USENIX Security Symposium (USENIX Security 19), Santa Clara, CA, Aug. 2019, pp. 639–656. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/shusterman

[7] J. Gu, J. Wang, Z. Yu, and K. Shen, 'Traffic-Based Side-Channel Attack in Video Streaming', Ieee-Acm Trans. Netw., vol. 27, no. 3, pp. 972–985, Jun. 2019, doi: 10.1109/TNET.2019.2906568.

[8] Q. Zhu and X. Zu, 'Fully Convolutional Neural Network Structure and Its Loss Function for Image Classification', Ieee Access, vol. 10, pp. 35541–35549, 2022, doi: 10.1109/ACCESS.2022.3163849.

[9] 7. Convolutional Neural Networks. Accessed: Oct. 22, 2022. [Online]. Available: https://learning.oreilly.com/library/view/deep-learning-from/9781800206137/Chapter_7_SMP_ePub.xhtml

[10]V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, 'Automated Website Fingerprinting through Deep Learning', in 25th Annual Network and Distributed System Security Symposium (ndss 2018), Reston, 2018. doi: 10.14722/ndss.2018.23105.