HTTPS Traffic Classification

Wazen M. Shbair, Thibault Cholez, Jérôme François, Isabelle Chrisment

Jérôme François
Inria Nancy Grand Est, France
jerome.francois@inria.fr

NMLRG - IETF95
April 7th, 2016

Outline

The HTTPS Dilemma

Security vs. Privacy

- HTTPS or HTTP-over-TLS is a protocol for secure communication over a computer network.
- Content providers (Google, Facebook, ...) need securing contents over the web by moving to HTTPS.
- Despite SSL/TLS good intentions, it may be used for illegitimate purposes.

The main research question

Can we rely on the monitoring
techniques that don't decrypt HTTPS traffic?

Overview of SNI

What is SNI ?

SNI is an extension inside Client Hello Message, proposed to support virtual hosting for websites use HTTPS.

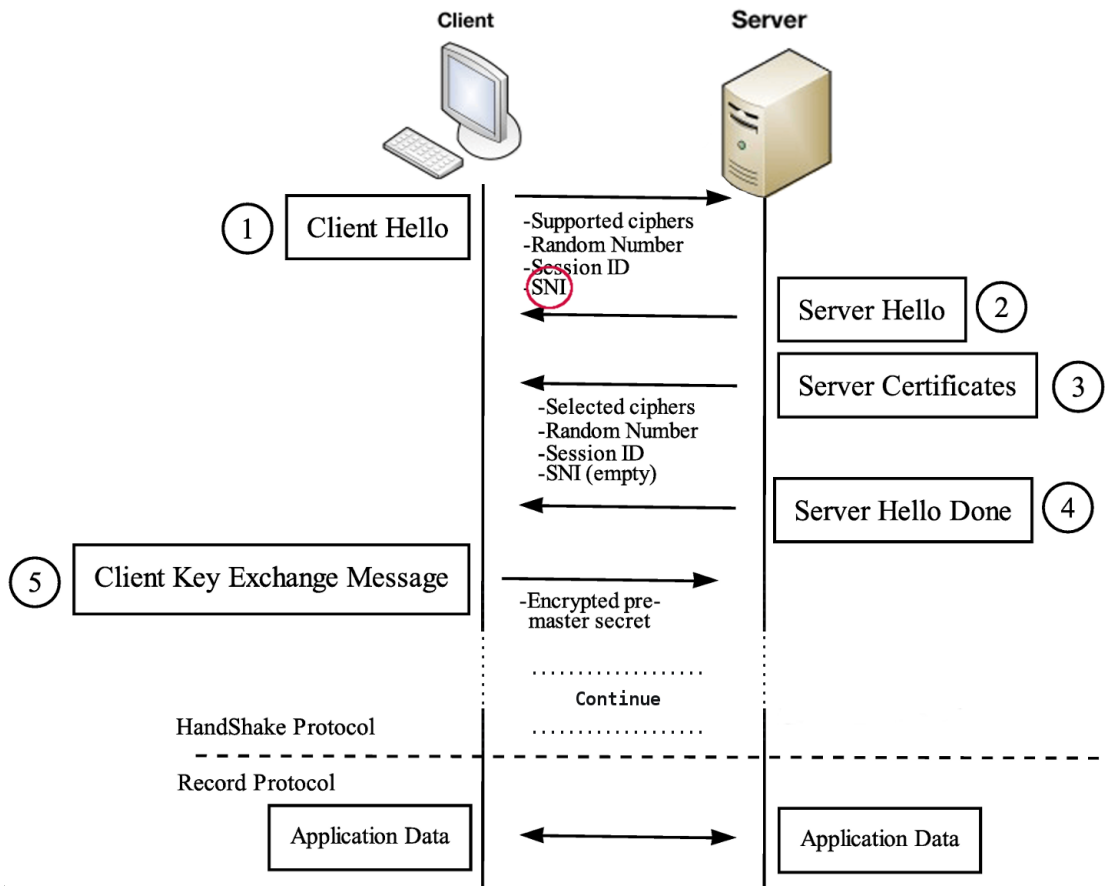


Figure : TLS handshake

SNI-based Filtering Evaluation

SNI-based filtering

- SNI-Filtering has two weaknesses, regarding the backward compatibility and multiple services using a single certificate.
- The "Escape" plug-in is our proof of concept exploiting SNI weaknesses.
- Successfully tested against 3 firewalls and top 20 visited websites such as Google Search, Facebook, Youtube, Twitter.

Publication

W.Shbair, T.Cholez, A.Goichot, I.Chrisment: "Efficiently Bypassing SNI-based HTTPS Filtering", IFIP/IEEE IM2015.

Identifying HTTPS Services

Flow-Based Statistical improvements

- One way is to combined it with algorithms from different fields like Machine Learning (ML) [1].
- It has been used widely in the identification of encrypted traffic problem.
- Mainly used to identifying the type of applications, such as (HTTPS, Mail, P2P, VoIP, SSH, Skype, etc.).

Identifying HTTPS Services

Flow-Based Statistical improvements

- One way is to combined it with algorithms from different fields like Machine Learning (ML) [1].
- It has been used widely in the identification of encrypted traffic problem.
- Mainly used to identifying the type of applications, such as (HTTPS, Mail, P2P, VoIP, SSH, Skype, etc.).

New Challenges

Considering all HTTPS as a single class is not enough for security monitoring because it regroups very different services.

Identifying HTTPS Services

Website Fingerprinting (WF)

- Defined as the process of identifying the URL of web pages that are accessed.
- Identifying accessed HTTPS encrypted web pages base on static object size parsed from unencrypted traffic [2].

Identifying HTTPS Services

Website Fingerprinting (WF)

- Defined as the process of identifying the URL of web pages that are accessed.
- Identifying accessed HTTPS encrypted web pages base on static object size parsed from unencrypted traffic [2].

WF Issue

It fails with dynamic web pages that use HTTPS Content Delivery Network (CDN) such as Akamai. (Too fine-grained)

A Multi-Level Framework to Identify HTTPS Services

The motivation

- An intermediate identification method monitors at service-level.
- Identify the HTTPS services without relying on header fields.
- Do not decrypt the HTTPS traffic.

The core techniques

1. Machine Learning techniques.
2. Novel multi-level classification approach.
3. Well tuned set of features

Machine Learning Techniques

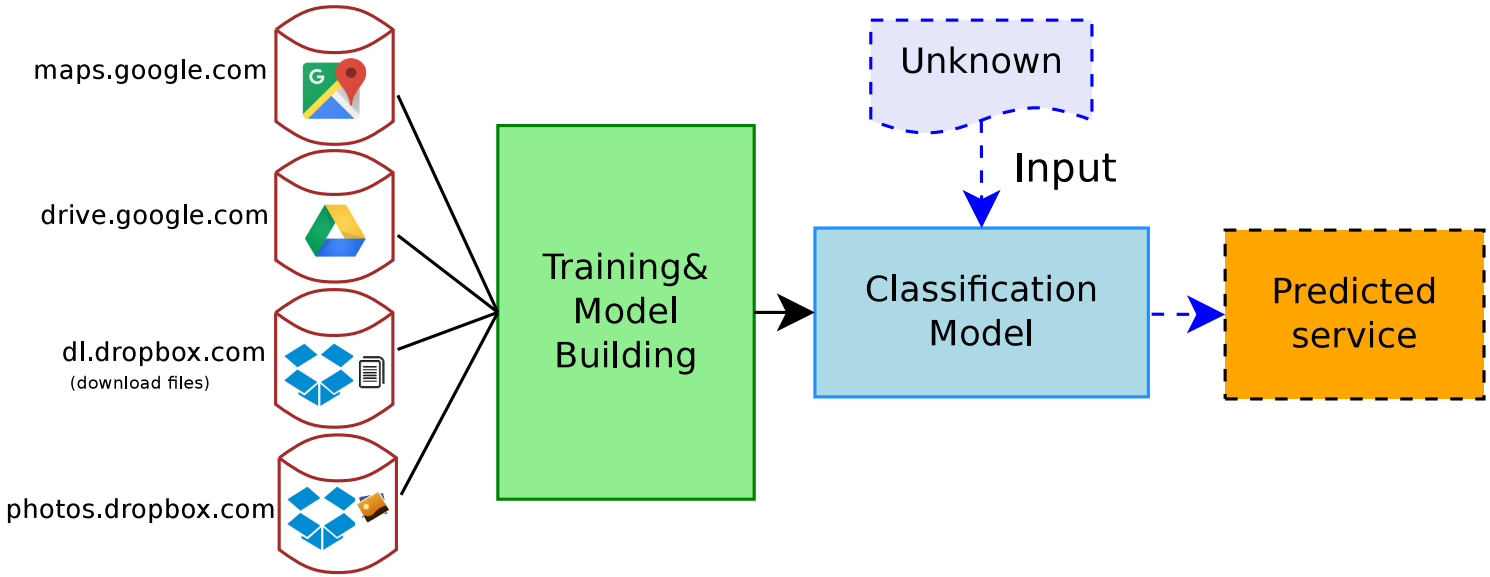


Figure : Flat classification view

The Legacy method

- The existing methods follow the "FLAT" view.
- Identifying the websites and applications directly.
- Drawbacks: low scalability, low accuracy and high error rate.

A Novel Multi-Level Classification Approach

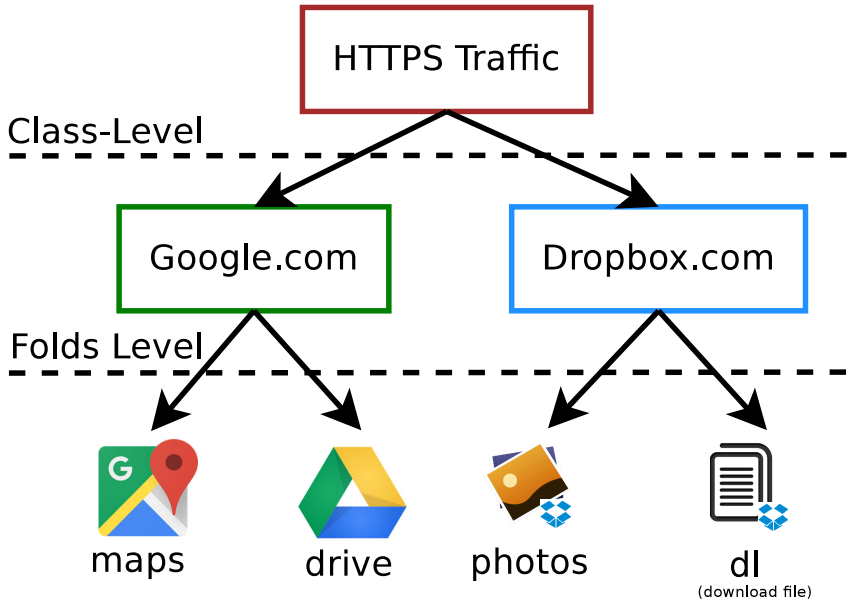


Figure : Multi-level presentation

Multi-level method

- Reform the training dataset into a tree-like fashion.
- The top level is refereed as Class-level (Root domain)
- The lower Level contains individual Folds-level (Sub-domain)

A Multi-Level Framework to Identify HTTPS Services

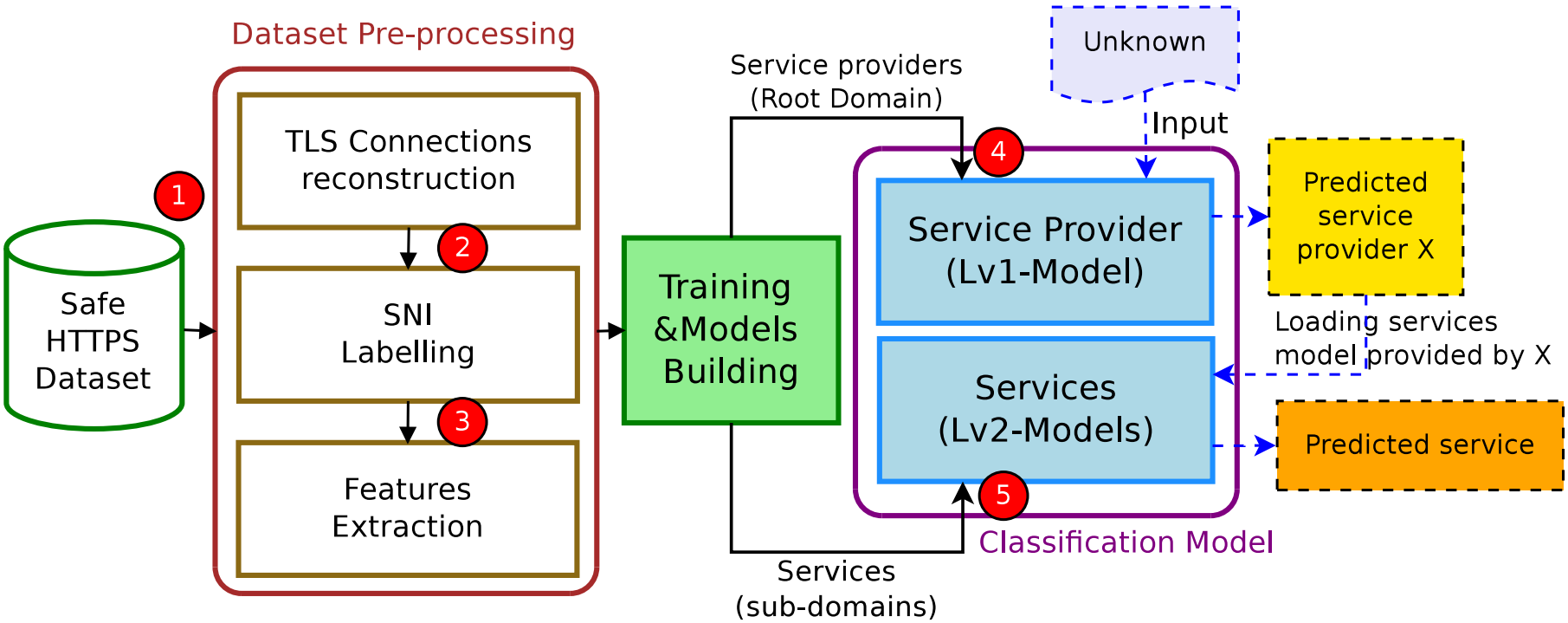


Figure : The work-flow of the HTTPS traffic identification framework

Multi-Level Classification Approach

The novel evaluation method

A novel method more suitable for multi-level approach:

- If service provider and the service name are predicted
  $\rightarrow$ **Perfect identification**.
- If service provider is predicted but not the service name
  $\rightarrow$ **Partial identification**.
- If neither service provider nor the service name are predicted
  $\rightarrow$ **Invalid identification**

Methodology

Overview

The evaluation of the proposed solution contains 3 parts:

- Evaluation of the collected dataset.
- Evaluation of the proposed features set.
- Evaluation of the multi-level classification approach.

Evaluation of the collected dataset

- Contains more than 288,901 HTTPS connections.
- Pre-processed to be suitable for multi-level approach.
- Processed to determine a reasonable threshold for the minimum number of labelled connections per service.

Features selection

Evaluation of the proposed features set

- Classical **30 features** from previous work [3, 4]
- New **12 features** are proposed over the encrypted payload
- The 42 features are optimized by Features Selection technique
- The key benefits is reducing over-fitting by removing irrelevant and redundant features [5]

Feature Selection result

- 18 features are highly relevant: 10 out of 12 from our proposed set and 8 out of 30 from the classical ones.
- This validates the rationale of the proposed features for identifying HTTPS services.

The 18 selected features

Client $\leftrightarrow$ Server
Inter Arrival Time (75th percentile)
Client $\rightarrow$ Server
Packet size (75th percentile, Maximum), Inter Arrival Time (75th percentile), **Encrypted Payload(Mean, 25th, 50th percentile, Variance, maximum)**
Server $\rightarrow$ Client
Packet size (50th percentile, Maximum), Inter Arrival Time (25th, 75th percentile), **Encrypted payload(25th, 50th, 75th percentile, variance, maximum)**

Experiments and Evaluation Results

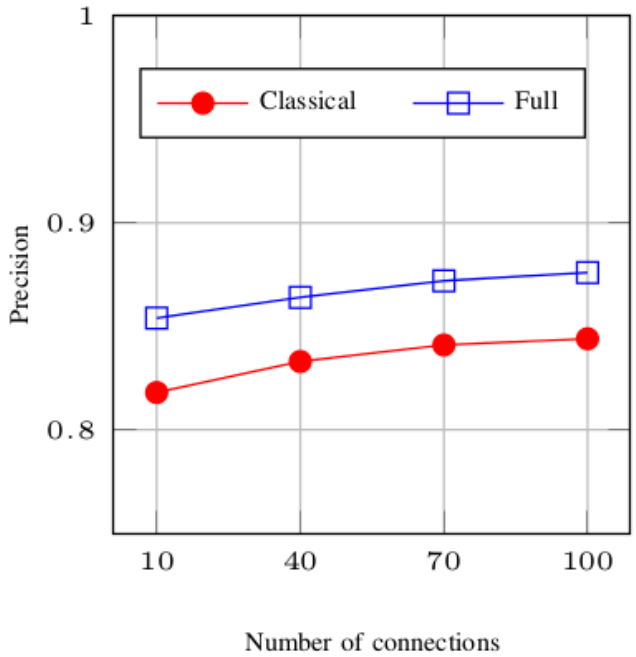
Evaluation of the proposed features set

By using WEKA [1] tool the features set are evaluated by C4.5 and RandomForest algorithm:

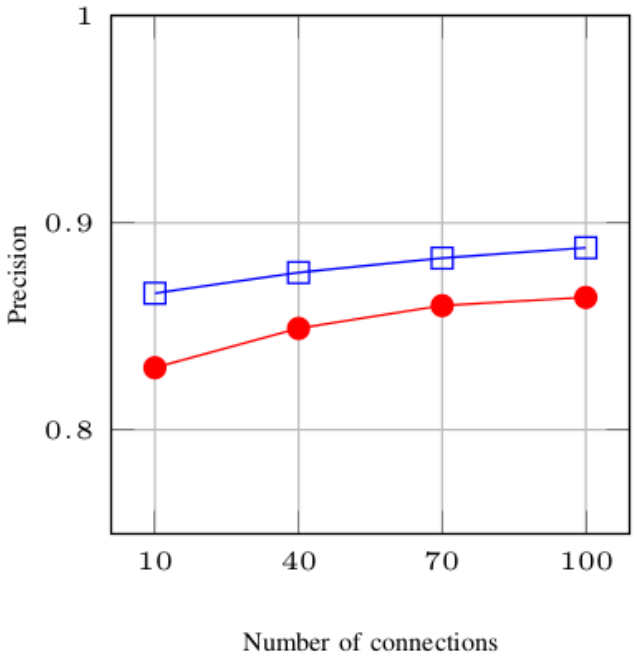
- **Classical 30-features**:
  C4.5 achieves 83.4%±1.0 Precision,
  RandomForest achieves 85.7%±0.4 Precision.
- **Selected 18-features**:
  C4.5 achieves 85.87%±0.64 Precision,
  RandomForest achieves 87.60%±0.10 Precision.
- **Full 42-features**:
  C4.5 achieves 86.65%±0.7 Precision,
  RandomForest achieves 87.82%±0.68 Precision.

[1] www.cs.waikato.ac.nz

Minimal number of connections



(a) Precision comparison between Classical and Full features with C4.5



(b) Precision comparison between Classical and Full features with RandomForest

Muti-level classification

HTTPS Identification Framework Evaluation

- The framework has been evaluated in two steps:
  - Evaluate each level separately, to measure the performance of each classification model.
  - Evaluate the whole framework as one black box.
- Evaluation conditions:
  - Full features set (42 features).
  - RandomForest as ML algorithm.
  - At least 100 connections number per service.
  - K-Fold cross validation with k=10.

Evaluation Results

Top Level Evaluation

Experiments show that we can identifying the service provider of HTTPS traffic with 93.6% overall accuracy.

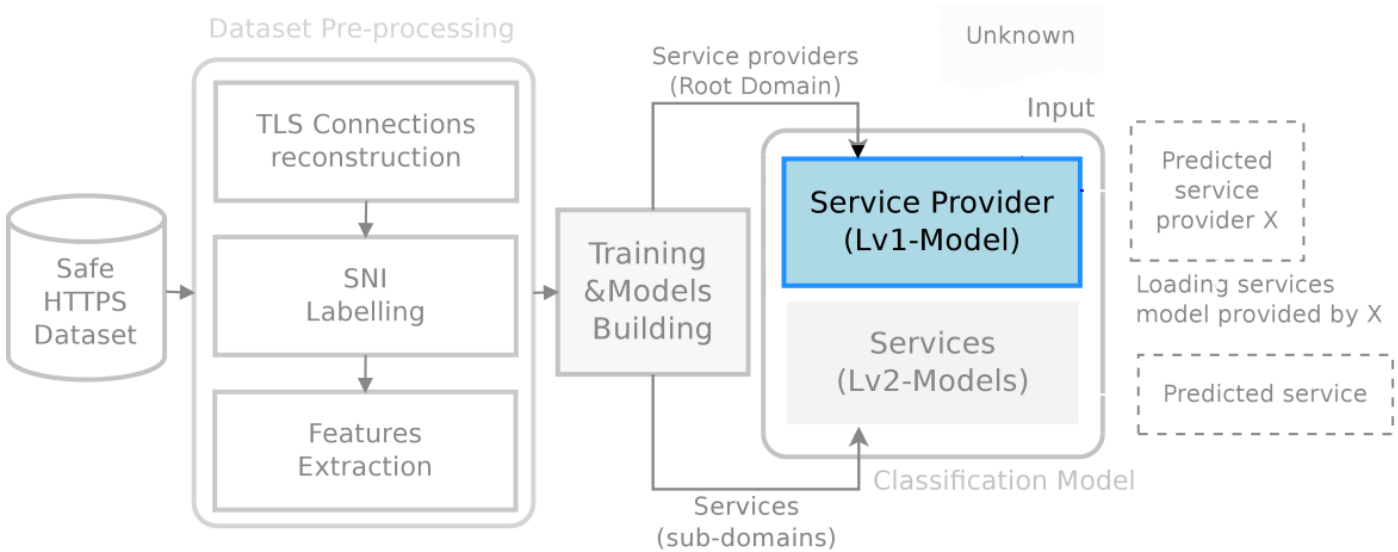


Figure : Top Level of the framework

Evaluation Results

Second Level Evaluation

A separate classification models are built and evaluated for each service provider with the same approach used in the Top-level.

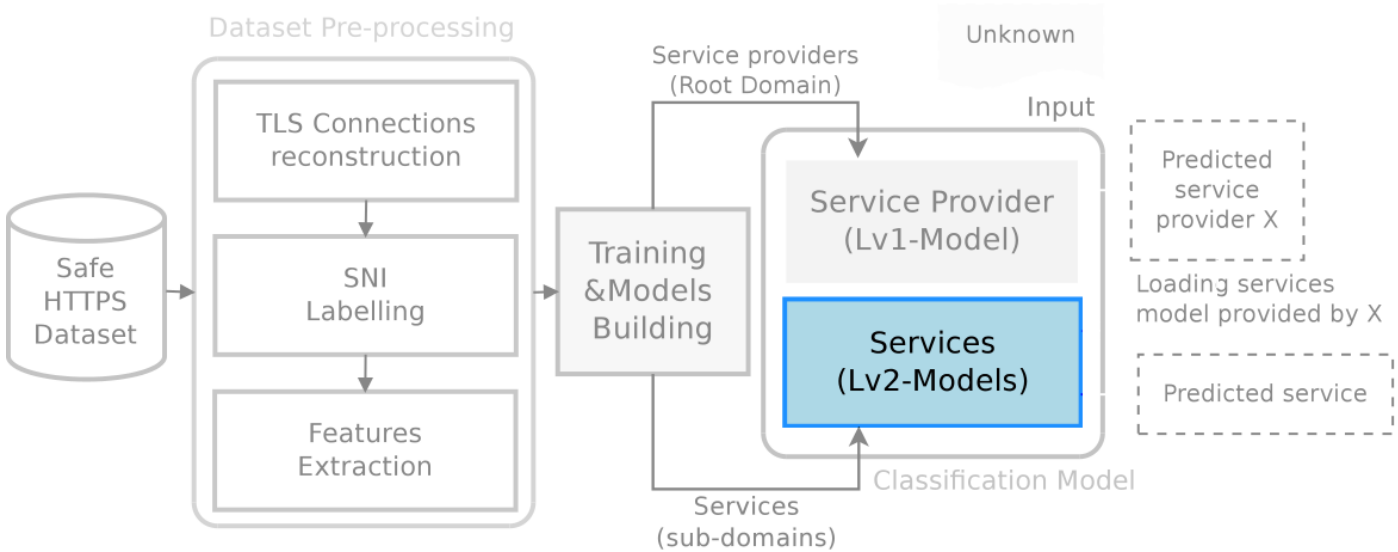


Figure : Second Level of the framework

Evaluation Results

Second Level Evaluation

- From 68 distinct service providers, 51 service providers have more than 95% of good classification of their own different services.
- For example, we can differentiate between 19 services run under Google.com, with 93% of Perfect identification.

Table : The second level models accuracy

Accuracy Range	Nb of service providers		
-	Classical Features	Full Features	Selected Features
100-95%	50	51	51
95-90%	5	5	5
90-80%	6	6	6
Less than 80%	7	6	6

Global results

Evaluate the framework as black-box (Level1&2)

Results show that we achieve 93.10% of Perfect identification and 2.9% of Partial identification.

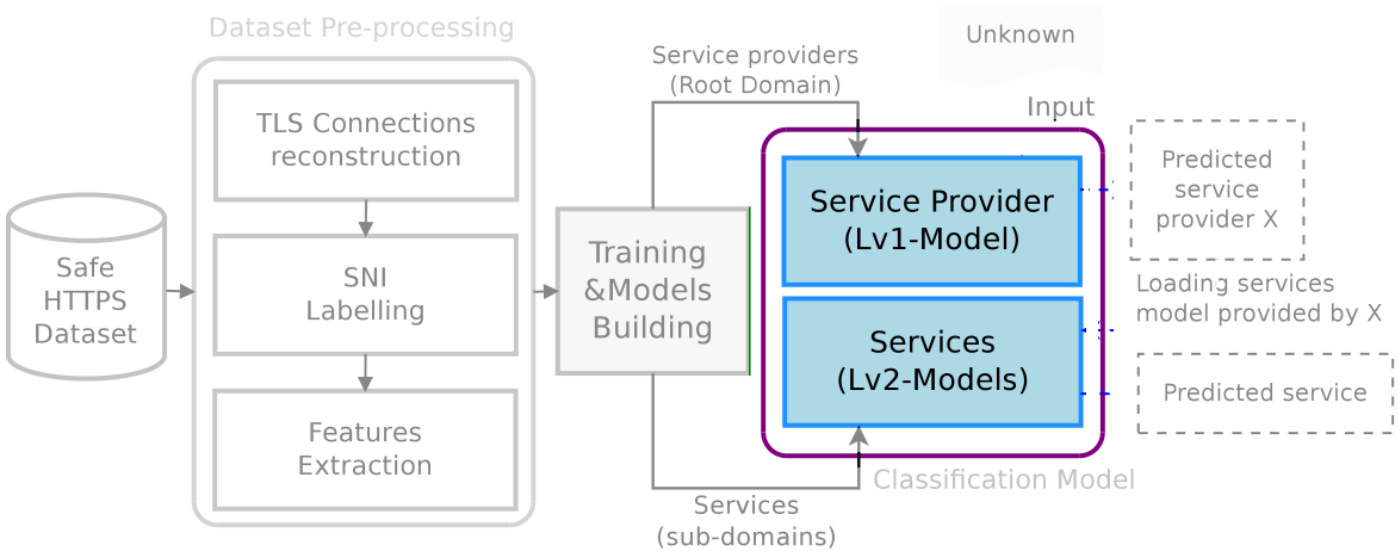


Figure : The complete classification model

Stability over time

The classification errors over time

We can notice that even after 23 weeks without new learning phase, we still identify 80% (error <20%) of HTTPS services.

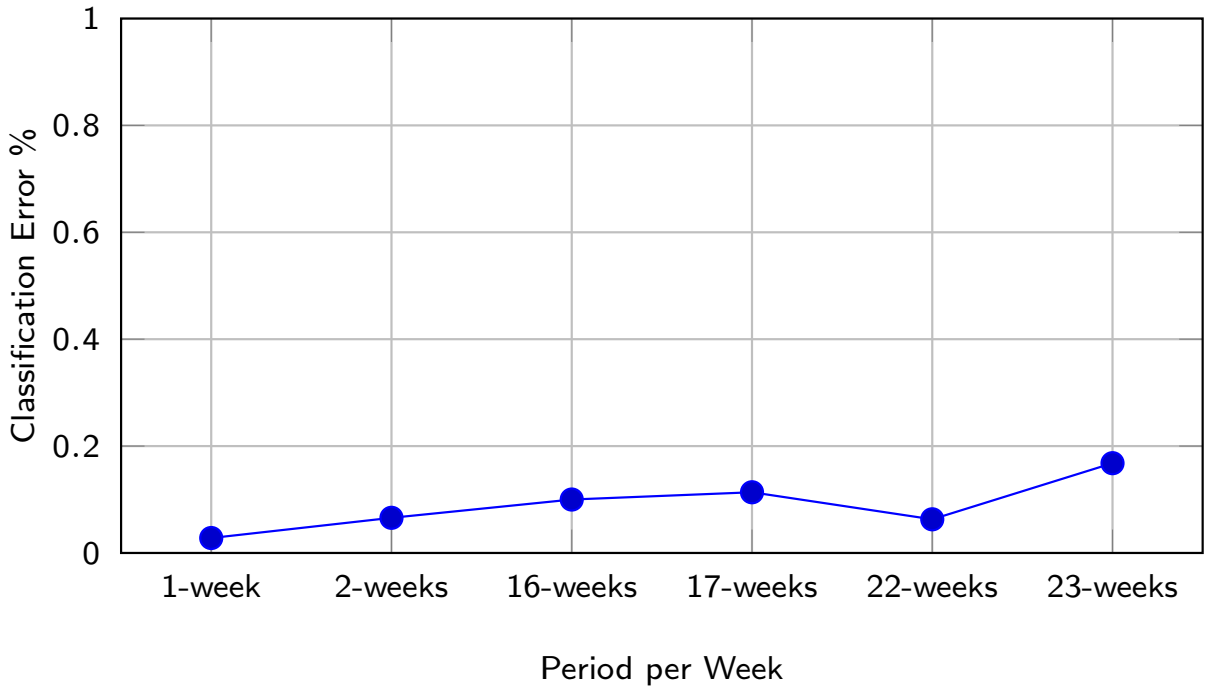


Figure : Effect upon classification error over time

Conclusion & Future work

Conclusion

- A complete framework to identify the HTTPS services with several innovations (Multi-level classification, SNI-labelling, new set of features).
- Based on real traffic, the results show that despite the challenging task, a high level of accuracy of 93.10% achieved.

Future Work

- To adapt and extend our current framework for real-time analysis identification of HTTPS services.
- Improve the global security of networks especially by developing a HTTPS firewall.

Publications

Publications

1. W.Shbair, T.Cholez, A.Goichot, I.Chrisment: "Efficiently Bypassing SNI-based HTTPS Filtering", IFIP/IEEE IM2015.
2. W.Shbair, T.Cholez, J.Franois, and I.Chrisment, A multi-level framework to identify HTTPS services", (To appear in IFIP/IEEE NOMS2016), April/2016.

References

[1] W. de Donato, A. Pescape, and A. Dainotti, "Traffic identification engine: an open platform for traffic classification," *Network, IEEE*, vol. 28, no. 2, pp. 56–64, 2014.

[2] B. Miller, L. Huang, A. D. Joseph, and J. D. Tygar, "I know why you went to the clinic: Risks and realization of https traffic analysis," in *Privacy Enhancing Technologies*, pp. 143–163, Springer, 2014.

[3] Y. Okada, S. Ata, N. Nakamura, Y. Nakahira, and I. Oka, "Comparisons of machine learning algorithms for application identification of encrypted traffic," in *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, vol. 2, pp. 358–361, IEEE, 2011.

[4] Y. Kumano, S. Ata, N. Nakamura, Y. Nakahira, and I. Oka, "Towards real-time processing for application identification of encrypted traffic," in *Computing, Networking and Communications (ICNC), 2014 International Conference on*, pp. 136–140, IEEE, 2014.

[5] H. Kim, K. C. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, and K. Lee, "Internet traffic classification demystified: myths, caveats, and the best practices," in *Proceedings of the 2008 ACM CoNEXT conference*, p. 11, ACM, 2008.