**Feature Extraction**

In this task, we aim to extract feature for each sequentially collected packet. Given the existing data, the attributes of a data package are dynamic. Being dynamic means that some features of some data packets may not exist for some other packets. For example, TCP packets do not have the features of security socket layer, or the TLS packets intended for handshake layer have certificate attributes while those for record layer have not. Nevertheless, we need to turn any packet into a numeric vector so that it can be trained in machine learning algorithm. This document illustrates how a given packet, p, is to be transferred to a numeric vector.

In our task, the data packets are classified as either TCP packets or SSL/TLS packets. Moreover, SSL/TLS packets have more cryptographic features.

**Shared Features (TCP features):**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Type** | **Description** |
| Come/Leave | Boolean (0/1) | From the view of webserver, whether the packet is leaving from or coming to the server. |
| Protocol | Enum (6 protocols):  TCP, SSL2, SSL3, TLS1, TLS2, TLS3 |  |
| Length | Integer |  |
| Interval | Float (unit: ms) | The time interval from previous packet |
| Flag | Enum (9 flags)  NS, CWR, ECE, URG, ACK, PSH, RST, SYN, FIN | See: https://en.wikipedia.org/wiki/Transmission\_Control\_Protocol |
| Windows Size | Integer |  |
| Total vector dimension: 19 | | |

**Dynamic Features (SSL/TLS features):**

The following features are optional for some packets.

*Dimension is zero if does not contain the feature.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | | | **Type** |
| Handshake Protocol | ClientHello | Length | Integer |
| Cipher suites | Enum (79 suites):  TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 (0xc02c), TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 (0xc030), TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 (0xc02b), TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (0xc02f), TLS\_ECDHE\_ECDSA\_WITH\_CHACHA20\_POLY1305\_SHA256 (0xcca9), TLS\_ECDHE\_RSA\_WITH\_CHACHA20\_POLY1305\_SHA256 (0xcca8), TLS\_DHE\_DSS\_WITH\_AES\_256\_GCM\_SHA384 (0x00a3), TLS\_DHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 (0x009f), TLS\_DHE\_DSS\_WITH\_AES\_128\_GCM\_SHA256 (0x00a2), TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (0x009e), TLS\_DHE\_RSA\_WITH\_CHACHA20\_POLY1305\_SHA256 (0xccaa), TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM\_8 (0xc0af), TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CCM (0xc0ad), TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384 (0xc024), TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384 (0xc028), TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA (0xc00a), TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA (0xc014), TLS\_DHE\_RSA\_WITH\_AES\_256\_CCM\_8 (0xc0a3), TLS\_DHE\_RSA\_WITH\_AES\_256\_CCM (0xc09f), TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256 (0x006b), TLS\_DHE\_DSS\_WITH\_AES\_256\_CBC\_SHA256 (0x006a), TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA (0x0039), TLS\_DHE\_DSS\_WITH\_AES\_256\_CBC\_SHA (0x0038), TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM\_8 (0xc0ae), TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CCM (0xc0ac), TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 (0xc023), TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 (0xc027), TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA (0xc009), TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA (0xc013), TLS\_DHE\_RSA\_WITH\_AES\_128\_CCM\_8 (0xc0a2), TLS\_DHE\_RSA\_WITH\_AES\_128\_CCM (0xc09e), TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 (0x0067), TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA256 (0x0040), TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA (0x0033), TLS\_DHE\_DSS\_WITH\_AES\_128\_CBC\_SHA (0x0032), TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 (0x009d), TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256 (0x009c), TLS\_RSA\_WITH\_AES\_256\_CCM\_8 (0xc0a1), TLS\_RSA\_WITH\_AES\_256\_CCM (0xc09d), TLS\_RSA\_WITH\_AES\_128\_CCM\_8 (0xc0a0), TLS\_RSA\_WITH\_AES\_128\_CCM (0xc09c), TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256 (0x003d), TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256 (0x003c), TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA (0x0035), TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA (0x002f), TLS\_EMPTY\_RENEGOTIATION\_INFO\_SCSV (0x00ff), TLS\_DH\_RSA\_WITH\_AES\_256\_CBC\_SHA (0x0037), TLS\_DH\_DSS\_WITH\_AES\_256\_CBC\_SHA (0x0036), TLS\_DHE\_RSA\_WITH\_CAMELLIA\_256\_CBC\_SHA (0x0088), TLS\_DHE\_DSS\_WITH\_CAMELLIA\_256\_CBC\_SHA (0x0087), TLS\_DH\_RSA\_WITH\_CAMELLIA\_256\_CBC\_SHA (0x0086), TLS\_DH\_DSS\_WITH\_CAMELLIA\_256\_CBC\_SHA (0x0085), TLS\_ECDH\_RSA\_WITH\_AES\_256\_CBC\_SHA (0xc00f), TLS\_ECDH\_ECDSA\_WITH\_AES\_256\_CBC\_SHA (0xc005), TLS\_RSA\_WITH\_CAMELLIA\_256\_CBC\_SHA (0x0084), TLS\_DH\_RSA\_WITH\_AES\_128\_CBC\_SHA (0x0031), TLS\_DH\_DSS\_WITH\_AES\_128\_CBC\_SHA (0x0030), TLS\_DHE\_RSA\_WITH\_SEED\_CBC\_SHA (0x009a), TLS\_DHE\_DSS\_WITH\_SEED\_CBC\_SHA (0x0099), TLS\_DH\_RSA\_WITH\_SEED\_CBC\_SHA (0x0098), TLS\_DH\_DSS\_WITH\_SEED\_CBC\_SHA (0x0097), TLS\_DHE\_RSA\_WITH\_CAMELLIA\_128\_CBC\_SHA (0x0045), TLS\_DHE\_DSS\_WITH\_CAMELLIA\_128\_CBC\_SHA (0x0044), TLS\_DH\_RSA\_WITH\_CAMELLIA\_128\_CBC\_SHA (0x0043), TLS\_DH\_DSS\_WITH\_CAMELLIA\_128\_CBC\_SHA (0x0042), TLS\_ECDH\_RSA\_WITH\_AES\_128\_CBC\_SHA (0xc00e), TLS\_ECDH\_ECDSA\_WITH\_AES\_128\_CBC\_SHA (0xc004), TLS\_RSA\_WITH\_SEED\_CBC\_SHA (0x0096), TLS\_RSA\_WITH\_CAMELLIA\_128\_CBC\_SHA (0x0041), TLS\_RSA\_WITH\_IDEA\_CBC\_SHA (0x0007), TLS\_ECDHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA (0xc012), TLS\_ECDHE\_ECDSA\_WITH\_3DES\_EDE\_CBC\_SHA (0xc008), TLS\_DHE\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA (0x0016), TLS\_DHE\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA (0x0013), TLS\_DH\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA (0x0010), TLS\_DH\_DSS\_WITH\_3DES\_EDE\_CBC\_SHA (0x000d), TLS\_ECDH\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA (0xc00d), TLS\_ECDH\_ECDSA\_WITH\_3DES\_EDE\_CBC\_SHA (0xc003), TLS\_RSA\_WITH\_3DES\_EDE\_CBC\_SHA (0x000a) |
| Cipher suites length | Integer |
| Compression method | Enum (1 method):  null (0) |
| Supported group length | Integer |
| Supported groups | Enum (4 groups):  x25519 (0x001d), secp256r1 (0x0017), secp521r1 (0x0019), secp384r1 (0x0018) |
| Encrypt then mac length | Integer |
| Extended master secret | Integer |
| Signature hash algorithm | Enum (15 algorithms):  rsa\_pkcs1\_sha512 (0x0601), SHA512 DSA (0x0602), ecdsa\_secp521r1\_sha512 (0x0603), rsa\_pkcs1\_sha384 (0x0501), SHA384 DSA (0x0502), ecdsa\_secp384r1\_sha384 (0x0503), rsa\_pkcs1\_sha256 (0x0401), SHA256 DSA (0x0402), ecdsa\_secp256r1\_sha256 (0x0403), SHA224 RSA (0x0301), SHA224 DSA (0x0302), SHA224 ECDSA (0x0303), rsa\_pkcs1\_sha1 (0x0201), SHA1 DSA (0x0202), ecdsa\_sha1 (0x0203) |
| ServerHello | Length | Integer |
| Extended master secret | Integer  (feature **not included** because not found in wireshark) |
| Renegotiation info length | Integer |
| Certificate | Average certificate length | Integer |
| Maximum certificate length | Integer |
| Minimum certificate length | Integer |
| Number of Certificates | Integer |
| Signature algorithm | Enum (4 algorithms):  1.2.840.113549.1.1.11 (sha256WithRSAEncryption), 1.2.840.113549.1.1.12 (sha384WithRSAEncryption), 1.2.840.10045.4.3.2 (ecdsa-with-SHA256), 1.2.840.113549.1.1.5 (sha1WithRSAEncryption) |
| Server Hello Done | Length | Integer |
| Client Key Exchange | Length | Integer |
| Pubkey length | Integer |
| Encrypted Handshake Message | Length | Integer |
| Change Cipher Spec | Change Cipher Spec | Length | Integer |
| Application Data Protocol | Application Data Protocol | Length | Integer |
| Total vector dimension: 120 | | | |

**For Discussion:**

1. ~~We need to complement some features (red ones) with more data.~~
2. What features should be used for classical machine learning algorithm? How do we aggregate the data?