FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Klasifikace akcí přenášených skrz šifrované TLS spojení
Student:	Zdena Tropková
Vedoucí:	Ing. Karel Hynek
Studijní program:	Informatika
Obor / specializace:	Bezpečnost a informační technologie
Katedra:	Katedra počítačových systémů
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

Seznamte se s problematikou monitorování síťového provozu na úrovni paketů a tzv. síťových toků (IP flows).
Nasbírejte vzorky vybraných typů TLS provozu (např. audio a video streaming, procházení webu) v podobě zachycených paketů a rozšířených síťových toků a tím vytvořte anotovanou datovou sadu síťového provozu. Proveďte analýzu zachyceného provozu se zaměřením na charakteristické vlastnosti, které je možné využít pro jejich identifikaci.
Navrhněte algoritmus pro rozpoznávání typů provozu přenášeného skrz šifrovaný TLS kanál na základě charakteristik chování komunikace.
Dle návrhu vytvořte softwarový prototyp, který je schopen zpracovávat provoz z reálné sítě.

Navržené řešení otestuje a vyhodnoťte přesnost klasifikace a výkonové parametry prototypu (např. potřebné zdroje a rychlost zpracování dat).

Elektronicky schválil/a prof. Ing. Pavel Tvrdík, CSc. dne 7. ledna 2021 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Klasifikace akcí přenášených skrz šifrované TLS spojení

Zdena Tropková

Katedra počítačových systémů
Vedoucí práce: Ing. Karel Hynek

10. května 2021

Poděkování

Ráda bych poděkovala vedoucímu Ing. Karlu Hynkovi za veškerou pomoc při vypracování této bakalářské práce a jeho cenné rady. Dále bych chtěla poděkovat mému příteli Petrovi za jeho ochotu a trpělivost. Poděkování patří také mé rodině, která mě podporovala během celého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 10. května 2021 .....

Odkaz na tuto práci

Tropková, Zdena. *Klasifikace akcí přenášených skrz šifrované TLS spojení.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Tato bakalářská práce se zabývá analýzou a klasifikací šifrovaného TLS spojení na bázi síťových toků. V první části je představena anotovaná datová sada vytvořená převážně z provozu z reálné sítě, na základě které je dále provedena analýza chování TLS (Transport Layer Security) spojení. Dohromady je v ní rozpoznáno šest kategorií síťového provozu. V druhé části práce je navrhnut klasifikátor, který využívá pouze paketové informace a dávky paketů k rozpoznání akcí přenášených v šifrovaném spojení. Na základě jeho výsledku je provedena analýza špatných predikcích a zároveň jsou uvedeny možné důvody, proč k chybám při klasifikaci dochází. Výstupem práce je prototyp, který umožňuje klasifikovat provoz z reálné sítě.

Klíčová slova klasifikace síťového provozu, TLS, síťové toky, šifrovaný provoz, dávky paketů

Abstract

This bachelor's thesis deals with the analysis and flow-based classification of encrypted TLS (Transport Layer Security) traffic. The first part describes an annotated dataset created from real network traffic and analyses of TLS connections. There are six categories of network traffic distinguished in total. The second part of the thesis focuses on the implementation of a classifier utilizing packet and burst information for recognizing actions transmitted over an encrypted connection. The classification results are analyzed and possible reasons for misclassification are discussed. The outcome of the thesis is a prototype that enables the classification of real network traffic.

Keywords network traffic classification, TLS, network flows, encrypted traffic, bursts

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Množství šifrovaných spojení v síťovém provozu se v posledních letech téměř zdvojnásobilo. Zatímco v roce 2017 bylo šifrováno pouze 55 procent síťového provozu [1], v roce 2019 to bylo již přes 90 procent [2]. Šifrování dat při přenosu sítí je tedy dnes běžnou praxí a zároveň i nutností, protože u mnoha online služeb jako je například internetové bankovnictví jsou přenášená citlivá data, která musí být chráněna před zneužitím a odposlechnutím třetí stranou. Zároveň šifrování poskytuje uživatelům pohybujícím se na Internetu větší soukromí, na druhou stranu přináší mnoho výzev při monitorování síťového provozu.

Šifrovaný provoz značně omezuje možnosti monitorovacích nástrojů v detekci škodlivé aktivity probíhající v síti, což může být velké bezpečností riziko. U šifrované komunikace je totiž velmi obtížné detekovat abnormální chování, protože informace o spojení jsou značně omezeny kvůli šifrování. Mnoho monitorovacích nástrojů se proto snaží využít dat, která jsou odeslána ještě před zahájením šifrované komunikace.

U protokolu TLS se jedná o informace z handshaku jako je například rozšíření Server Name Indication poskytující jméno serveru, ke kterému se snaží klient připojit. Tento údaj je při monitorování velmi cenný, ovšem již nyní existuje rozšíření [3], které umožňuje jeho zašifrování, čím je znemožněno jeho využití pro účely monitorování.

Tato bakalářská práce se zaměřuje na klasifikaci šifrovaného TLS provozu na bázi síťových toků bez znalosti Server Name Indication. Ke klasifikaci budou využity pouze paketové informace a dávky paketů, díky čemu klasifikování nebude závislé na žádné nezašifrované části spojení.

Cílem je vytvoření anotované datové sady TLS provozu, ve které bude identifikováno několik skupin akcí. Na základě těchto dat bude provedena analýza charakteristik chování jednotlivých kategorií. Poté bude navrhnut klasifikátor, který umožní rozpoznávání zadefinovaných skupin. Tento klasifikátor bude

následně využitý v prototypu, který bude schopný přijmout na vstupu provoz z reálné sítě a klasifikovat jednotlivé síťové toky.

V první kapitole je popsán protokol TLS a síťové toky, dále jsou uvedeny možnosti monitorování sítí a nástroje k tomu používané, nakonec jsou představeny způsoby klasifikace síťového provozu. V druhé kapitole je uveden postup tvorby datové sady a způsob získání dat. V další kapitole je provedena analýza charakteristik spojení a popsán postup návrhu klasifikátoru. Pátá kapitola popisuje implementaci prototypu. V poslední kapitole jsou diskutovány špatné predikce klasifikátoru a je provedeno testování výsledného prototypu.

Kapitola 1

Rešerše

V této kapitole je popsán protokol TLS, poté jsou zadefinovány síťové toky a popsány možnosti jejich obohacení. Dále jsou představeny způsoby monitorování síťového provozu a nástroje k tomu používané včetně systému NEMEA [4], který byl využit při vypracování této bakalářské práce. V poslední části kapitoly jsou uvedeny možné přístupy ke klasifikace síťového provozu.

1.1 TLS

Transport layer security neboli TLS je protokol umožňující zašifrovaný přenos dat v počítačové síti mezi dvěma subjekty, zajišťuje bezpečnou komunikaci a brání před její nežádoucí modifikací. [5] Nejčastěji se využívá společně s protokolem HTTP při prohlížení webových stránek, ale bývá také používán například s protokoly IMAP nebo FTP. [6]

Spojení se skládá ze dvou částí. První je handshake, při kterém jsou vyjednány parametry spojení a autentizace subjektů. Druhá část je šifrovaná komunikace mezi subjekty.

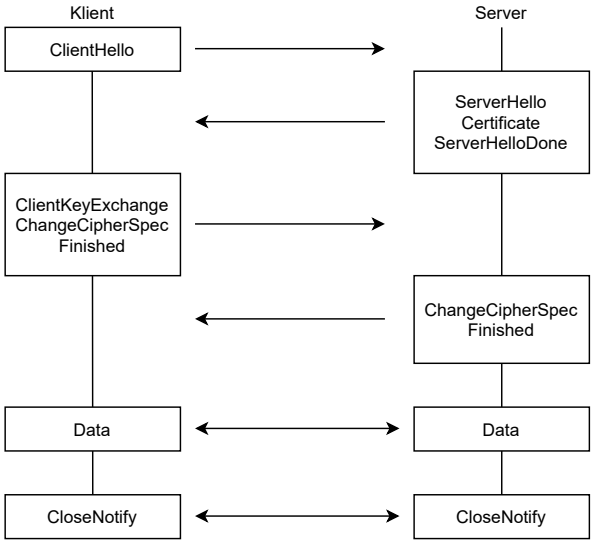
1.1.1 Handshake

Podle standardu [5] handshake začíná v momentě, kdy klient kontaktuje server. Odešle zprávu *ClientHello* obsahující náhodný řetězec a seznam podporovaných šifer a verzí TLS. Server vybere z nabízených parametrů vhodnou kombinaci a také vygeneruje náhodný řetězec, který pošle klientovi ve zprávě *ServerHello*. Pokud nelze najít shodu v parametrech, spojení je ukončeno. Následně server odešle svůj certifikát, podle kterého klient ověří jeho identitu, a zprávu *ServerHelloDone*, kterou oznamuje, že ukončuje tuto část handshaku. Klient pošle *ClientKeyExchange* zprávu, což je další náhodný řetězec tentokrát zašifrovaný veřejným klíčem serveru, který si ho pomocí svého soukromého klíče po přijetí dešifruje. Na základě těchto tří poslaných náhodných řetězců dopočítají oba subjekty šifrovací klíč pro stávající spojení. Subjekty

si vzájemně odešlou zprávu *ChangeCipherSpec*, kterou dávají najevo, že dále bude komunikace šifrovaná dle dohodnutých parametrů a klíčem, které obě strany dopočítaly. Zda klíč vyšel stejný u klienta i serveru se ověří vzájemným poslání zprávy *Finished*, která je už zašifrovaná vypočítaným klíčem.

1.1.2 Průběh spojení

Pokud vše v handshaku proběhlo v pořádku, je navázáno šifrované spojení a nastává druhá fáze, kdy si klient se serverem vyměňují zašifrované zprávy. Spojení je uzavřeno v momentě, kdy oba subjekty odešlou zprávu *CloseNotify*, čím dávají najevo, že už neodešlou další data. Tento krok je nezbytný pro zachování bezpečnosti, protože jinak by spojení mohlo být zneužito útočníkem. Během spojení může nastat fatální chyba, při její detekci musí subjekt okamžitě odeslat zprávu s varováním, načež musí oba subjekty okamžitě ukončit spojení a neodesílat žádná další data. Celý průběh spojení je vykreslen v obrázku 1.1.



Obrázek 1.1: Průběh spojení TLS [5]

1.1.3 Rozšíření

Protokol TLS je v dnešní době široce používán, aby byl kompatibilní s největším možným počtem platforem, používají se rozšíření, díky kterým je možné zařídit kompatibilitu i v případech, které nebyly zohledněny při návrhu protokolu. Jedno z rozšíření je SNI neboli Server Name Indication [7]. Je běžnou praxí, že pod jednou IP adresou běží více virtuálních serverů. Pomocí SNI specifikuje klient ve zprávě *ClientHello* virtuální server, se kterým chce navázat spojení.

1.1.4 Verze

Nejnovější verze TLS 1.3 byla vydána v srpnu 2018. Oproti minulým verzím poskytuje bezpečnější a rychlejší spojení. Od předchozí verze TLS 1.2 se liší odstraněním podpory šifer, které jsou považovány již za zastaralé, a zrychlením handshaku a jeho šifrováním od zprávy *ServerHello*. Dále byl přidán mód *zero round-trip time*, který umožňuje rychlé navázání spojení bez handshaku, pokud již klient se serverem komunikoval a obě strany si uložily lokálně šifrovací klíč.

Nejpoužívanější verzí je nyní TLS 1.2, naopak TLS 1.1 přestalo být na konci března 2021 podporováno a doporučuje se ho již nepoužívat [8], proto se budeme v této bakalářské práci věnovat jen TLS 1.2 a 1.3.

1.2 Síťové toky

Síťový tok [9] je v rámci této bakalářské práce zadefinovaný jako sekvence paketů, která je zachycená v určitém časovém intervalu a agregována podle zdrojové a cílové IP adresy a zdrojového a cílového portu. Příklad síťového toku ve formátu JSON je na obrázku 1.2.

Toky lze rozdělovat na dvě skupiny, jednosměrné a obousměrné. Jednosměrný tok obsahuje informace o sekvenci paketů, která byla poslána pouze zdrojovým nebo naopak cílovým zařízením. U obousměrných toků jsou informace o paketech poslaných ze směru zdrojového a zároveň i cílového zařízení.

Široce využívaným protokolem při práci se síťovými toky je NetFlow [10], vyvinutý firmou Cisco, na základě kterého vznikl následně protokol IPFIX [11], který je nyní standardem podle organizace IETF a není vázaný jen na zařízení od Cisca na rozdíl od NetFlow. Oba protokoly určují strukturu síťových toků a umožňují manipulaci s nimi a jejich analýzu.

```
{
  "sitovy_tok": {
    "zdrojova_ip_adresa:": "10.0.0.1",
    "cilova_ip_adresa": "151.101.14.214",
    "zdrojovy_port": 443,
    "cilovy_port": 62432,
    "protokol": "TLS",
    "pocet_bajtu": 304545,
    "pocet_paketu": 614,
    "zacatek_cas": "2020-12-02 13:16:11.623000",
    "konec_cas": "2020-12-02 13:16:52.234000"
  }
}
```

Obrázek 1.2: Příklad síťového toku ve formátu JSON

1.2.1 Obohacování toků

Doplňování toků o další informace nazýváme obohacování toků. Díky této metodě není nutné uchovávat a analyzovat data z celého spojení, což je zpravidla paměťově i výkonnostně náročné. Pro analýzu obvykle stačí pouze obohacený tok, který poskytuje dostatečné množství informací o spojení.

Jednou z možností pro obohacení toků jsou paketové informace. Lze získat o paketu data jako jeho velikost, čas zachycení a v případě obousměrných toků směr, ze kterého přišel. Na základě těchto informací je možné dopočítávat hodnoty jako průměrnou velikost paketů ve spojení, časové intervaly mezi příchody paketů, počet přenesených bajtů za vteřinu a mnoho dalších, které umožňují charakterizovat a popsat spojení.

Paketové informace obvykle nebývají uchovány pro všechny jednotlivé pakety v toku, protože by to zvyšovalo paměťovou a výpočetní složitost. Uchovává se většinou předem určený počet prvních paketů a dále dopočítané charakteristiky jako počet všech přenesených paketů.

U toků protokolu TLS můžeme využít pro obohacení paketové informace, i přesto že je spojení šifrované, neovlivní to možnost získat tato data o paketech. Dále lze využít nezašifrovanou část handshaku. Jak již bylo zmiňováno, klient na začátku spojení odesílá serveru seznam podporovaných šifer. Na základě těchto hodnot lze vypočítat otisk nazývaný JA3 [12], který se dá dále používat k detekování škodlivé aktivity. Další nezašifrovanou hodnotou je SNI, které můžeme také použít pro obohacení.

1.2.2 ipfixprobe

Ipfixprobe [13] je modul systému NEMEA. Slouží k převedení provozu na síťovém rozhraní nebo souboru PCAP na síťové toky. Skládá se z několika pluginů, které vytvářejí a obohacují toky. Pluginy využívané v této bakalářské práci budou dále popsány v podkapitole Zpracování dat.

1.2.3 Cisco Joy

Cisco Joy [14] je softwarový balíček pod BSD licencí. Umožňuje monitorování síťového provozu, jeho převod na síťové toky a jejich obohacení. Dále poskytuje analyzační nástroje sloužící hlavně pro detekování škodlivého provozu a odhalení zranitelností. Obohacuje síťové toky například o délky a příchozí časy paketů, pokud se jedná o TLS spojení, extrahuje nešifrované informace jako seznam podporovaných šifer ze zprávy ClientHello.

1.2.4 CICFlowMeter

CICFlowMeter [15] je open source nástroj, který vytváří obousměrné síťové toky ze souborů PCAP. U časových parametrů umí pracovat s toky jako s jednosměrnými. Umožňuje snadné filtrování parametrů a doplňování nových,

zároveň z hodnot toku vypočítává statistiky jako průměr, rozptyl nebo minimální hodnotu.

1.3 Monitorování síťového provozu

Monitorování sítě je proces, ve kterém je sledován provoz probíhající v počítačové síti mezi komponenty jako jsou například routery nebo servery. Tímto procesem jsou získávána data používaná pro správu sítě nebo analýzu komunikace.

Jedním z přístupů je monitorování na úrovni síťových toků, další metoda je na úrovní paketů, kdy je zpracováván každý paket zvlášť včetně obsahu payloadu [16]. Každý z přístupů má určité výhody a nevýhody, proto se v některých případech používá kombinace obou přístupů. Monitorování na úrovní síťových toků je méně výkonnostně náročné a umožňuje monitorovat i šifrovaný provoz, ovšem nemusí poskytovat dostatečné množství informací pro určité typy analýzy na rozdíl od monitorování na úrovni paketů.

Existuje velké množství nástrojů, které poskytují možnost monitorovat síť. V rámci síťových toků je to například systém NEMEA, pro monitorování na úrovni paketů je to například program Suricata [17].

1.3.1 Wireshark

Wireshark [18] je nástroj, který zachytává síťový provoz a následně umožňuje jeho offline analýzu. Zachycený provoz lze filtrovat podle mnoha parametrů a provádět hloubkovou analýzu velkého množství protokolů včetně TLS. K dispozici je grafické uživatelské rozhraní nebo verze pro příkazovou řádky nazývaná TShark.

1.3.2 NEMEA

NEMEA [4] je IDS systém určený pro monitorování síťového provozu v reálném čase na bázi síťových toků. Data přijímá ze souboru nebo pomocí kolektoru, který je získává z provozu sítě. Skládá se z modulů, které pracují nezávisle na sobě, ovšem všechny využívají framework NEMEA [19], pomocí kterého komunikují mezi sebou a sdílejí společné algoritmy a datové struktury [20].

Modul přijme na vstupním rozhraní data, zpracuje je a výstup odešle přes výstupní interface k dalšímu zpracování. Mezi činnosti modulů patří například vypočítávání statistik nebo detekce útoků a abnormálního chování.

1.4 Klasifikace v monitorování

Zastoupení šifrované komunikace v síťovém provozu v posledních letech prudce narůstá [2], v říjnu 2019 bylo z webové komunikace zašifrováno přes 90 procent, proto je problematika klasifikování šifrovaného provozu aktuální. Exis-

tuje mnoho přístupů, jedním z nich je klasifikace na úrovni aplikací jako například Skype, Gmail nebo Spotify. V této práci [21] se zabývali 15 aplikacemi, které jsou schopni klasifikovat pomocí hloubkového učení s vysokou přesností v reálném čase v rámci domácí sítě, nezaměřovali se ovšem pouze na protokol TLS. Klasifikaci aplikací využívajících pouze TLS publikovali v tomto článku [22], kde dosáhli v experimentu přesnosti 90 procent, přičemž využívali dat z handshaku a zároveň paketových informací.

Dalším přístupem je neklasifikovat přímo specifické aplikace, ale obecné skupiny aktivit jako přenos souboru, přehrávání videa nebo odesílání e-mailu. V této publikaci [23] se zaměřili na 7 skupin, které klasifikovali u šifrovaného VPN spojení. Použili strojové učení a zaměřili se hlavně na časové parametry jako například příchozí časy paketů. Podařilo se jim dosáhnout přesnosti více než 80 procent. Podobný přístup využívají i v tomto článku [24], kde klasifikují protokol TLS. Věnovali se jen aplikačním datům, hlavně délkám paketů a časovými intervaly mezi nimi. Klasifikovali pomocí strojového učení a dosáhli přesnosti více než 95 procent.

Článek [25] se zabývá nešifrovanou komunikací, ale i přesto se zaměřuje jen na parametry toku (délka, počet přenesených paketů apod.) a paketové informace. Klasifikuje 10 druhů spojení algoritmem SVM s přesností větší než 80 procent.

Kapitola 2

Tvorba datové sady

Tato kapitola popisuje postup tvorby datové sady. Nejprve jsou zadefinovány skupiny akcí společně s jejich zástupci, které bylo nutné si určit před začátkem zachytáváním dat pro tvorbu datové sady. Dále je popsán proces zachytávání síťového provozu, při kterém bylo dohromady zachyceno přes 8 terabajtů dat v síti CESNET2 a vygenerováno přes 25 gigabajtů síťového provozu. Následné je uveden postup zpracování zachyceného síťového provozu do podoby anotovaných obohacených síťových toků.

2.1 Popis skupin

Na základě analýzy bylo zvoleno dohromady 6 skupin, které budou v síťovém provozu klasifikovány. Jedná se o skupiny video přehrávané živě, video z přehrávače, přehrávání hudby, nahrávání souboru, stahování souboru a procházení webu. Pro každou z nich bylo určeno několik zástupců, protože vzhledem k rozšířenosti TLS není možné obsáhnout všechny služby, které ho používají. Někteří ze zástupců byly vybráni na základě seznamu [26], který obsahuje 500 nejoblíbenějších webových stránek. Skupiny s jejich zástupci jsou uvedeny v tabulce 2.1.

Tabulka 2.1: Skupiny klasifikace a jejich zástupci

Skupina	Zástupci
Video přehrávané živě	Twitch, živé vysílání ČT, YouTube Live
Video z přehrávače	DailyMotion, Stream, Vimeo, Youtube
Přehrávání hudby	AppleMusic, Spotify, SoundCloud
Nahrávání souborů	FileSender, OwnCloud, OneDrive, Drive
Stahování souborů	FileSender, OwnCloud, OneDrive, Drive
Procházení webu	Webové stránky ze seznamu Alexa Top Sites [27]

2.2 Získání dat

Data byla získána dvěma způsoby – generováním a zachytáváním. Nejprve byl vygenerován síťový provoz pro všechny zástupce skupin. Následně byla provedena jeho analýza a extrakce IP adres klasifikovaných služeb. Vytvořený seznam IP adres byl následně použit pro zachycení reálného síťového provozu, kterým byla dále datová sada rozšířena. Celý postup bude detailněji popsán v následujících podkapitolách.

2.2.1 Generování

Data byla získána manuálním i automatizovaným způsobem. V obou případech byl využit nástroj Wireshark, případně jeho verze pro příkazovou řádku TShark. Zachycený síťový provoz byl ukládán ve formátu PCAP, protože se jedná o standardní formát pro ukládání zachycených paketů, který je podporován velkým množstvím nástrojů.

Umělá data byla generována pomocí dvou různých operačních systémů a prohlížečů. Pro manuální záchyt byl použit MacBook s operačním systémem macOS Catalina a prohlížečem Mozilla Firefox. V případě automatizovaného způsobu byl využit nástroj VirtualBox, který umožňuje vytvoření a běh virtuálních strojů. Pro zachytávání byl vytvořen virtuální počítač, na kterým běžel operační systém Ubuntu 20 společně s prohlížečem Google Chrome.

Pro automatizaci procesu byl vytvořen shellový skript. Na vstupu je dvojice čísel (minimální a maximální délka nahrávání ve vteřinách) a seznam webových odkazů. Pro každý odkaz ze seznamu je následně spuštěn proces zachytávání, jehož výstupem je soubor PCAP.

Nejdříve je v rámci každého zachytávajícího procesu spuštěn program TShark, jakmile běží zachytávání síťového provozu, je otevřen prohlížeč na daném odkazu. Proces nahrávání je zastav v momentě, kdy vyprší čas. Délka je nastavována pomocí náhodně vygenerované hodnoty ležící v intervalu. Velikost intervalu je určena čísly, které byly zadány na vstupu.

Pseudokód 1 Skript pro automatizované zachytávání

Vstup: seznam odkazů, minimální a maximální délka trvání zachytávání
Výstup: soubor PCAP

```
1: for all odkazy do
2:     vygenerovat náhodné číslo X z intervalu ze zadaných čísel na vstupu
3:     spustit TShark
4:     spustit prohlížeč s odkazem
5:     čekat X vteřin
6:     ukončit běh prohlížeče
7:     ukončit běh TSharku
8: end for
```

2.2.2 Zachytávání

Jak již bylo zmiňováno, na základě vygenerovaných dat byl vytvořen seznam IP adres služeb, které jsou klasifikovány v rámci skupin. Zachycení spojení s těmito IP adresami z provozu reálné sítě bylo provedeno na perimetru sítě CESNET2.

2.3 Zpracování dat

Zachycená data ve formě souborů PCAP byla nejdříve zpracována exportérem síťových toků ipfixprobe, který spadá pod nástroj NEMEA. Byly využity kromě základního pluginu také pluginy *pstats*, *bstats* a *tls*. Výstupem každého rozšiřujícího pluginu je samostatný soubor ve formátu trapcap.

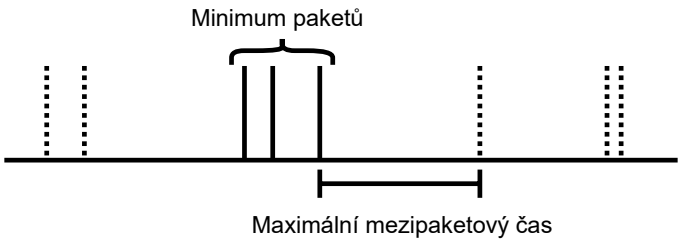
Základní plugin uvádí informace o tocích jako zdrojovou a cílovou MAC adresu, počet přijatých a odeslaných paketů a bajtů, časy začátku a konce spojení nebo protokol. Tyto hodnoty se následně exportují v souborech z dalších pluginů.

Plugin *pstats* obohacuje tok o informace o prvních 30 paketech. Jedná se o jejich délky, časy, směry a flagy.

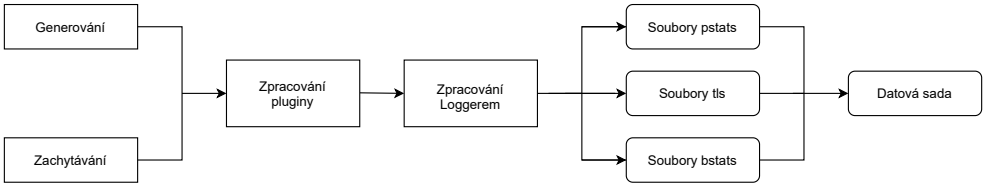
Další plugin *bstats* pracuje s dávkami paketů. Dávku paketů definuje jako sekvenci minimálně 3 paketů, které byly doručeny z jednoho směru spojení v maximálním intervalu sekundu po sobě. Znázornění dávky paketů je vykresleno na obrázku 2.1. *Bstats* obohacuje toky o počty paketů a bajtů v jednotlivých dávkách paketů a časy začátku a konce trvání dávky paketů. Tyto informace uvádí pro každý směr zvlášť.

Posledním použitým je plugin *tls*. Ten exportuje informace ze základního pluginu o celém TLS spojení a připojuje SNI a otisk JA3.

Pro převedení toků z binární podoby byl na závěr použit modul NEMEA logger, který převádí soubory trapcap na soubory typu CSV, které umožňují snadnější práci se síťovými toky.



Obrázek 2.1: Znázornění dávky paketů



Obrázek 2.2: Průběh tvorby datové sady

2.4 Anotace, agregování a filtrování dat

Informace o síťovém toku byly po výstupu ze zpracování dat ve 3 rozdílných souborech. Pro další práci bylo nutné všechny užitečné informace spojit dohromady, zároveň vyfiltrovat vadné toky a anotovat zbylé.

Jako první byla spojena data ze souborů vytvořených pluginy *bstats* a *pstats*. Informace byly agregovány podle IP adres, portů a času. Následně z nich byly selektována pouze TLS spojení, pomocí souboru z pluginu *tls*, který obohatil jednotlivé toky o SNI.

Dále bylo nutné toky vyfiltrovat. Vzhledem k tomu, že část dat pocházela z reálného provozu, bylo potřeba se ujistit, že byly správně zachycené. Záchyt může obsahovat šumy a chyby, proto byly odstraněny záznamy, které měly nulové počty přijatých nebo odeslaných paketů. Dále byly toky vyfiltrovány podle SNI. Zachycení dat bylo prováděno na základě seznamu IP adres, ovšem častým případem bylo, že na IP adrese bylo více virtuálních serverů, tudíž byly v datech i spojení patřící jiným službám, než které jsou klasifikovány. Těmto tokům by nemohly být s jistotou přiřazeny štítky, proto byly pomocí SNI vybrána jen data spadající pod zástupce z klasifikovaných skupin, které je možné bezpečně anotovat.

Anotaci byla prováděna pomocí SNI. Vzhledem k tomu, že o generovaných datech lze přesně říci, že patří dané službě, byl vytvořen seznam se SNI pro jednotlivé služby. Na základě tohoto seznamu byly anotovány ostatní toky štítky, které je přiřazovaly do jedné z 6 skupin.

Tabulka 2.2: Počet síťových toků podle skupiny v datové sadě

Skupina	Počet	Štítek
Video přehrávané živě	10 373	L
Video z přehrávače	12 553	P
Přehrávání hudby	10 701	M
Nahrávání souborů	10 862	U
Stahování souborů	20 393	D
Procházení webu	80 789	W

2.5 Struktura datové sady

Výše popsaných postupem vznikla datová sada, které obsahuje 145 671 záznamů. Je v ní rozlišeno 6 skupin síťového provozu, přesné zastoupení jednotlivých kategorií je v tabulce 2.2. Pro vytvoření datové sady bylo použito přes 8 terabajtů dat zachycených v síti CESNET2 a 25 gigabajtů vygenerovaného síťového provozu.

Kapitola 3

Analýza dat a návrh klasifikace

V této kapitole jsou popsány jednotlivé charakteristiky síťových toků. Je provedena analýza charakteristik celé datové sady a zároveň i vybraných jednotlivých síťových toků. Dále je popsán postup návrhu klasifikace, pro kterou byly vyzkoušeny 4 algoritmy strojového učení. Vyhodnocení jejich úspěšnosti je uvedeno na konci kapitoly.

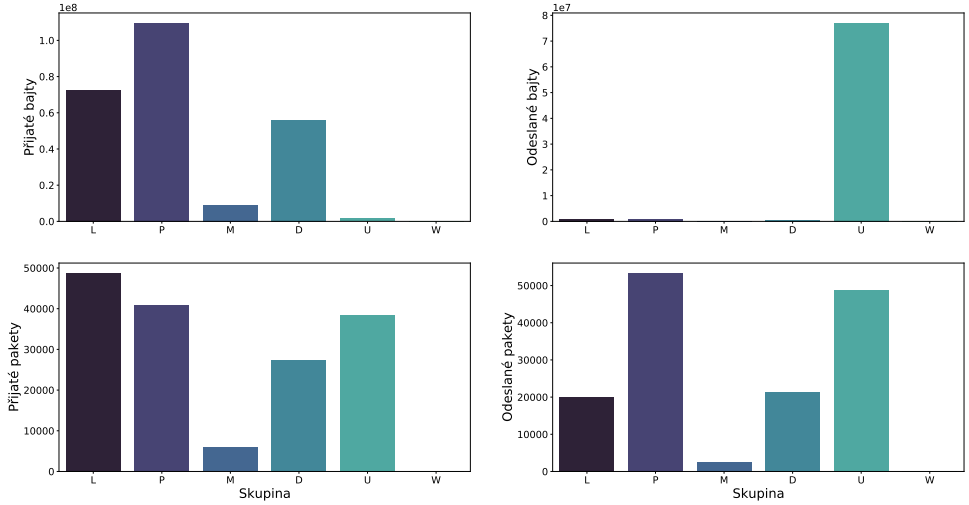
3.1 Charakteristiky skupin

Analýza dat byla provedena dvěma způsoby. Analyzována byla data z celé datové sady a zároveň bylo vybráno z každé klasifikované skupiny několik zástupců pro detailní analýzu průběhu celého TLS spojení.

Každá ze skupin má určité parametry, které charakterizují její chování. Patří mezi ně množství přenesených bajtů a paketů ve spojení. Průměrné hodnoty jednotlivých skupin jsou vidět v grafu 3.1. Dále byly zkoumány paketové informace jako velikosti paketů a časové intervaly mezi jejich příchody. Jako poslední byly analyzovány dávky paketů. Přesněji jejich velikost v bajtech, množství paketů v jednotlivých dávkách paketů, doba trvání a intervaly mezi příchody dávek paketů.

3.1.1 Přehrávání videa živě

Pro živé přehrávání je typické, že mezipaketové intervaly jsou krátké a dávky paketů jsou průměrně nejdelší v rámci všech skupin, což odpovídá intuitivní představě, že při přehrávání živého videa musí data neustále přicházet po malých částech, protože je nelze načíst dopředu na rozdíl od přehrávání videa z přehrávače, kdy přehrávaný obsah je dopředu znám. Zároveň tato skupina má největší průměrné množství příchozích paketů. V jednom spojení přijde v průměru téměř 80 megabajtů.



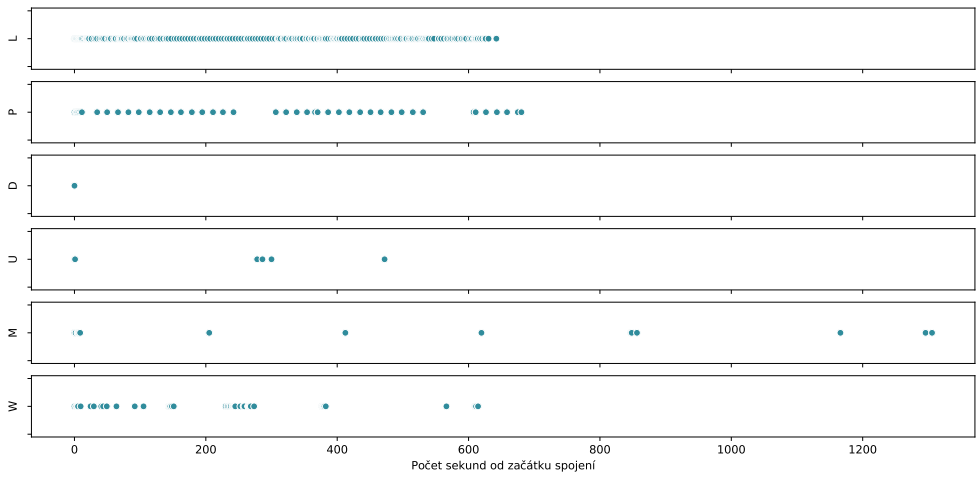
Obrázek 3.1: Průměrné množství paketů a bajtů

3.1.2 Přehrávání videa z přehrávače

Skupina se spojeními přenášející video z přehrávače se vyznačuje průměrně největšími pakety. Intervaly mezi jejich příchody jsou v průměru nejdelší po skupině hudba, což je způsobeno nejspíš bufferováním videa, kdy je video posíláno uživateli po větších částech dopředu, proto nastávají delší časové mezery bez příchodu paketů. Počet dávek paketů ve spojení je největší v rámci všech skupin, a zároveň jsou dávky paketů o něco menší a jsou mezi nimi delší časové rozestupy oproti skupině živému přehrávání videa. Také je u této skupiny průměrně přeneseno největší množství dat, v průměru je přijato ve spojení skoro 100 megabajtů.

3.1.3 Přehrávání hudby

Pro spojení kategorie hudba jsou charakteristické delší mezipaketové intervaly, stejně tak i mezery mezi dávkami paketů. Podle analýzy jednotlivých spojení bylo zjištěno, že přijde několik dávek paketů za sebou v krátké době a následuje mezera, která odpovídala délkám přehrávaných skladeb, což je přibližně 3 minuty. Toto chování je vidět v grafu 3.2, který ukazuje časy příchodů dávek paketů pro jedno z reprezentativní spojení z každé skupiny. Průměrně množství přenesených dat je druhé nejmenší po kategorii procházení webu, stejně tak i průměrná velikost paketů je menší proti ostatních skupinám, průměr nedosahuje ani 1000 bajtů.



Obrázek 3.2: Časy příchodu dávek paketů

3.1.4 Přenos souboru

Další skupinou je stahování souboru. Má jedny z nejmenších paketových intervalů a pakety jsou velké jako u skupiny přehrávání živého videa. Pro stahování je hlavně charakteristická velikost dávek paketů v bajtech, která je průměrně největší ze všech kategorií. Netrvají ovšem nejdelší dobu, jsou průměrně dvojnásobně kratší než u skupiny přehrávaní živého videa, která má dávky paketů v průměru nejdelší. U stahování je tedy přeneseno více dat za kratší dobu v porovnání s ostatními kategoriemi.

Skupina nahrávání souboru má jako jediná větší množství odeslaných dat než přijatých. Průměrná velikost paketů je druhá největší. Intervaly mezi jednotlivými pakety jsou průměrně delší než u stahování.

3.1.5 Procházení webu

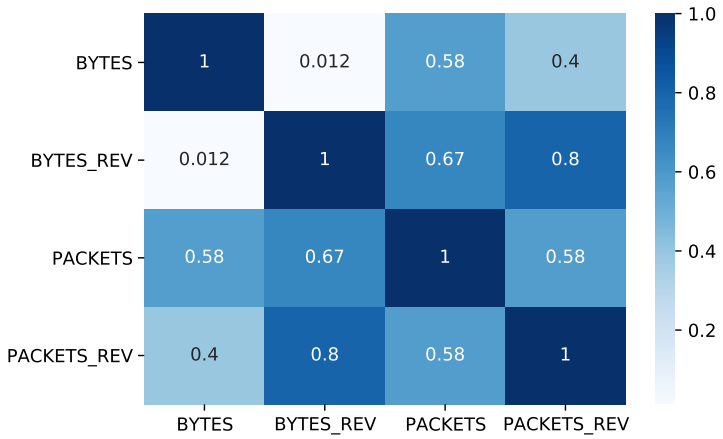
Poslední skupinou je procházení webu. Tato skupina má průměrně nejmenší množství přenesených paketů i bajtů. Průměrná velikost paketů je také výrazně menší oproti ostatním skupinám, zároveň jsou intervaly mezi jejich příchody v průměru nejdelší. V rámci spojení přijde také průměrně nejmenší množství dávek paketů, které trvají v průměru nejkratší dobu a mají nejmenší velikost oproti ostatním skupinám.

3.2 Výběr charakteristik klasifikace

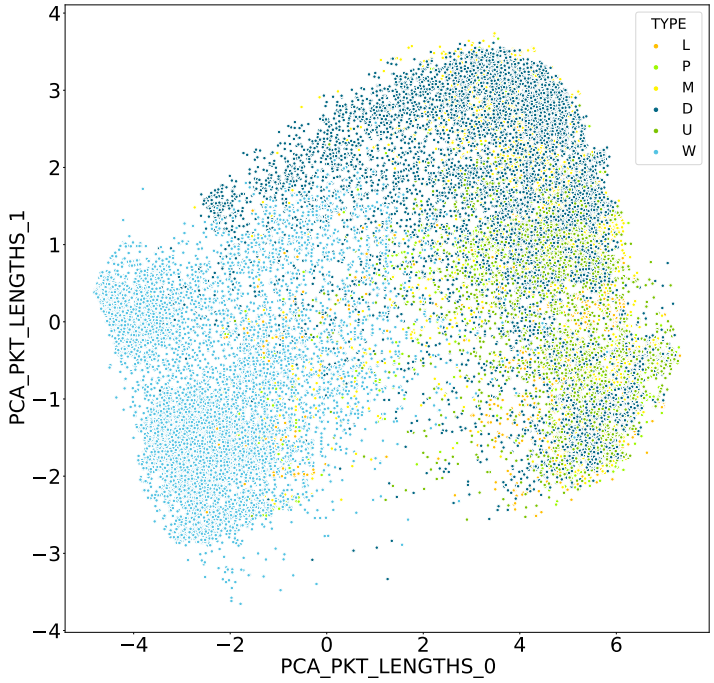
Dohromady bylo identifikováno 75 charakteristik, které jsou popsány v tabulce 3.1.

Tabulka 3.1: Charakteristiky klasifikace

Názvy charakteristik	Popis
BYTES, BYTES_REV, PACKETS, PACKETS_REV	Množství přijatých bajtů a paketů
REV_MORE	Zda bylo více bajtů přijato nebo odesláno
PKT_LENGTHS_MEAN, PKT_LENGTHS_MAX, PKT_LENGTHS_STD, PKT_LENGTHS_25, PKT_LENGTHS_50, PKT_LENGTHS_75	Informace o prvních 30 paketech
PKT_INTERVALS_MEAN, PKT_INTERVALS_MAX, PKT_INTERVALS_STD, PKT_INTERVALS_25, PKT_INTERVALS_50, PKT_INTERVALS_75	Informace o prvních 30 intervalech mezi pakety
BRST_BYTES_MEAN, BRST_BYTES_MAX, BRST_BYTES_STD, BRST_BYTES_25, BRST_BYTES_50, BRST_BYTES_75	Informace o velikostech dávek paketů v bajtech
BRST_PACKETS_MEAN, BRST_PACKETS_MAX, BRST_PACKETS_STD, BRST_PACKETS_25, BRST_PACKETS_50, BRST_PACKETS_75	Informace o velikostech dávek paketů v paketech
BRST_INTERVALS_MEAN, BRST_INTERVALS_MAX, BRST_INTERVALS_STD , BRST_INTERVALS_25, BRST_INTERVALS_50, BRST_INTERVALS_75	Informace o intervalech mezi dávkami paketů
BRST_DURATION_MEAN, BRST_DURATION_MAX, BRST_DURATION_STD, BRST_DURATION_25, BRST_DURATION_50 , BRST_DURATION_75	Informace o délkách trvání dávek paketů
PCA_PKT_LENGTHS_0, PCA_PKT_LENGTHS_1	PCA aplikováno na pole délek paketů
PCA_BRST_BYTES_0, PCA_BRST_BYTES_1, PCA_BRST_PACKETS_0, PCA_BRST_PACKETS_1, PCA_BRST_TIME_, START_0, PCA_BRST_, TIME_START_1, PCA_BRST_TIME_STOP_0, PCA_BRST_TIME_STOP_1, PCA_BRST_INTERVALS_0, PCA_BRST_INTERVALS_1, PCA_BRST_DURATION_0, PCA_BRST_DURATION_1	PCA aplikováno na pole s informacemi o dávkách paketů
BRST_BYTES_0–BRST_BYTES_9	Prvních 10 velikostí dávek paketů v bajtech
BRST_PACKETS_0–BRST_PACKETS_9	Prvních 10 velikostí dávek paketů v paketech



Obrázek 3.3: Korelační matice charakteristik bajtů a paketů



Obrázek 3.4: Graf PCA délky paketů

Z celé množiny byly eliminovány určité charakteristiky. Výběr menšího množství charakteristik pomáhá ke zlepšení výpočetního výkonu modelu a zároveň může v některých případech vylepšit jeho výsledky, protože některé charakteristiky mohou vést k chybnému učení klasifikátoru. Charakteristiky byly postupně vylučovány na základě korelace a také metody rekurzivní eliminace [28].

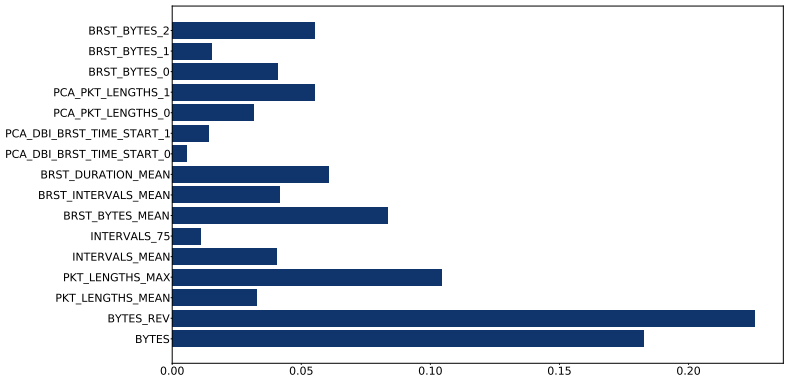
Velmi často mezi sebou korelují charakteristiky s pakety a bajty. Například množství přijatých bajtů koreluje s množstvím přijatých paketů, jak je vidět v korelační matici 3.3, proto byly odstraněné všechny charakteristiky obsahující pakety, pokud je šlo vyjádřit zároveň v bajtech. Stejně tomu bylo i u charakteristik, které reprezentovaly prvních 10 velikostí dávek paketů. Bylo zanecháno jen prvních 10 velikostí dávek paketů v bajtech, nikoliv i v paketech.

Dále bylo redukováno prvních 10 dávek paketů pouze na první 3, protože zbylých 7 nepřinášelo dostatečné množství informace, které by přispělo při učení modelu. Stejně tak byly odstraněny časové hodnoty, které reprezentovaly konec burstu, byly ponechány jen charakteristiky s časy začátku. Charakteristika REV_MORE byla také odebrána.

Korelace byla také mezi charakteristikami vyjadřující hodnoty jako rozptyl nebo maximální délka pro prvních 30 paketů. Stejně tomu bylo i u intervalů mezi pakety, velikostí dávek paketů, i u jejich doby trvání a intervaly mezi nimi. Z těchto hodnot bylo zanechaných jen několik.

Část informací o spojení byla uchovávána ve formě polí jako například prvních 30 paketů. Pro snížení dimenze do pouhých dvou bodů byla použita metoda PCA [29]. Ukázka PCA pro pole s délkou paketů je vidět v grafu 3.4. Výsledné hodnoty mezi sebou v některých případech korelovaly, proto jich část byla vyloučena.

Po eliminaci zůstalo 16 charakteristik, jejich důležitost pro klasifikátor Gradient boosting je vidět v grafu 3.5. Důležitost je vyjádřena Giniho důležitostí [30], čím větší hodnoty nabývá, tím je charakteristika pro klasifikátor důležitější.



Obrázek 3.5: Důležitost charakteristik

Využity byly následující charakteristiky. Množství přijatých a odeslaných bajtů, průměrná délka paketů a intervalů mezi nimi, velikost největšího paketu ve spojení, průměrná velikost dávek paketů v bajtech, dále průměr doby jejich trvání a intervalů mezi nimi. Také byly ponechány velikosti v bajtech prvních 3 příchozích dávek paketů. Z hodnot PCA byly použity jen časy začátků dávek paketů a velikosti paketů.

3.3 Undersampling

Protože počet spojení v jednotlivých skupinách datové sady byl nerovnoměrný, byly před klasifikací ze skupin procházení webu a stahování souboru náhodně odstraněné záznamy, aby jich v každé skupině byl přibližně stejný počet. Díky tomuto postupu vznikla rovnoměrná datová sada, kdy v každé skupině bylo okolo 10 000 záznamů. Tento postup se nazývá undersampling [31]. Pomáhá k přesnějšímu výsledku klasifikátoru, protože nepřevládá žádná skupina a výsledek není nerovnoměrným množstvím záznamů ovlivněn, klasifikátor by se mohl totiž častěji chybně přiklánět k převládající skupině. Opakem je oversampling, kdy jsou data kvůli nedostatku generována, aby byl počet záznamů ve všech skupinách dorovnán.

3.4 Použité klasifikační algoritmy

Pro klasifikaci síťového provozu bylo využito několik algoritmů strojového učení. Prvním z nich je algoritmus Random forest [32], který k učení klasifikátorů používá rozhodovací stromy. Rozhodovací strom je metoda strojového učení, kdy každý vnitřní uzel stromu reprezentuje podmínku pro charakteristiku, například zda je charakteristika větší než 0, a list uzlu je její rozhodnutí. Postupným procházením stromů od kořene a rozhodování jednotlivých podmínek je dosaženo výsledku klasifikace. Random forest využívá pro klasifikaci několik rozhodovacích stromů, kdy každý jednotlivý strom určí vlastní výsledek. Výsledkem je nejvíce zastoupený výsledek mezi jednotlivými stromy. Díky této metodě nedochází k přeučení klasifikátoru.

Dalším použitým algoritmem je Extra trees [33], který se liší od předchozího algoritmu ve způsobu volby charakteristik pro jednotlivé uzly. I dále použitý algoritmus Gradient boosting [34] využívá rozhodovací stromy. Během učení přidává postupně jednotlivé rozhodovací stromy, čím umožňuje lepší učení a větší přesnost klasifikátoru, může ovšem snadno nastat přeučení.

Posledním použitým algoritmem je k-NN neboli k-nejbližších sousedů [35]. Pro každý klasifikovaný prvek je v datové sadě nalezeno k nejpodobnějších prvků. Nejvíce zastoupená skupina u nalezených prvků je následně vybrána jako výsledek pro klasifikovaný prvek.

3.5 Učení klasifikátorů

Pro učení bylo využito výše vybraných 16 charakteristik. Datová sada byla před učením klasifikátoru rozdělena na učící a testovací skupinu v poměru 7:3, zároveň byl ve skupinách zanechán stejný poměr štítků, aby jednotlivé sady byly vyvážené. Dále byla data před učením klasifikátoru normalizována.

Zároveň byla použita metoda křížové validace [36], aby bylo zabráněno přeučení klasifikátoru. Jedná se o způsob, kdy jsou data rozdělena do n skupin. Následně proběhne n iterací, během kterých je vždy rozdílných *n-1* skupin trénovacích a poslední je použita na testování.

3.6 Hyperparametry

Pro dosažení nejlepšího výsledku byly laděny hyperpamarametry [37] jednotlivých algoritmů. Hyperpamatery jsou hodnoty, které jsou nastavovány uživatelem a používají se k nastavení způsobu učení klasifikátoru. Například u algoritmu Random forest lze nastavit množství využitých charakteristik nebo počet stromů a jejich maximální hloubku.

Existuje mnoho způsobů, jak nejvhodnější parametry najít. V rámci této bakalářské práce bylo použité tzv. náhodné hledání. Byla vytvořena matice obsahující různé hodnoty hyperparametrů, ze kterých byly náhodně vybírány kombinace. Následně byla pro konečnou klasifikaci použita kombinace, která dosáhla nejvyššího skóre při učení. Seznam nejlepších kombinací hyperparametrů je v tabulce 3.2.

Tabulka 3.2: Hodnoty hyperparametrů

Klasifikátor	Hyperparametr	Hodnota
Random forest	Počet stromů	30
	Max hloubka	20
	Max charakteristik	10
Extra trees	Počet stromů	400
	Max hloubka	30
	Max charakteristik	10
Gradient boosting	Počet stromů	300
	Max hloubka	10
	Max charakteristik	10
	Škálování	Standard
3-nn	Počet sousedů	3
	Škálování	Standard

3.7 Vyhodnocení výsledků

Pro hodnocení klasifikátorů byla zvolena metriky nazývané přesnost (accuracy), preciznost (precision), vytíženost (recall) a skóre F1. [38] Přesnost je počet správných predikcích klasifikátoru vydělených počtem všech provedených predikcí. Tato hodnota nemusí být vždy dostatečně vypovídající, protože záleží na rozložení datové sady. Skóre F1 zohledňuje další faktory jako pravdivě pozitivní (PP), falešně pozitivní (FP) nebo falešně negativní (FN) predikce. Jednotlivé metriky se vypočítají následujícím způsobem:

$$preciznost = \frac{PP}{PP + FP}$$

$$vytizenost = \frac{PP}{PP + FN}$$

$$F1 = 2 * \frac{preciznost * vytizenost}{preciznost + vytizenost}$$

Nejlépe si vedl klasifikátor Gradient boosting, jak je vidět v tabulce 3.3, dosáhl F1 skóre 0,987. Téměř stejný výsledek měl klasifikátor Extra trees se skórem F1 0,985, následoval Random forest se skórem F1 0,979. Posledním byl algoritmus 3-NN s F1 skórem 0,868.

Tabulka 3.3: Výsledky klasifikace

Klasifikátor	Přesnost	Skóre F1	Preciznost	Vytíženost
Random forest	0,987	0,979	0,976	0,982
Extra trees	0,990	0,985	0,982	0,988
Gradient boosting	0,992	0,987	0,985	0,990
3-NN	0,917	0,868	0,857	0,882

Kapitola 4

Popis a implementace řešení

Úkolem prototypu je přijmout provoz zachycený z reálné sítě, extrahovat z něj pouze TLS spojení v podobě síťových toků, dopočítat nutné charakteristiky a podle nich klasifikovat spojení do jedné z definovaných skupin, jak je vidět v diagramu 4.1. Pro vytvoření prototypu byl použit programovací jazyk Python, protože jeho moduly podporují práci s daty a použití nacvičeného klasifikátoru.

Prototyp je spouštěn z příkazové řádky. Na vstupu má 4 argumenty, které reprezentují cesty ke vstupním a výstupním souborům. Po spuštění je zavolána funkce `main`, ze které jsou potom volány další funkce.

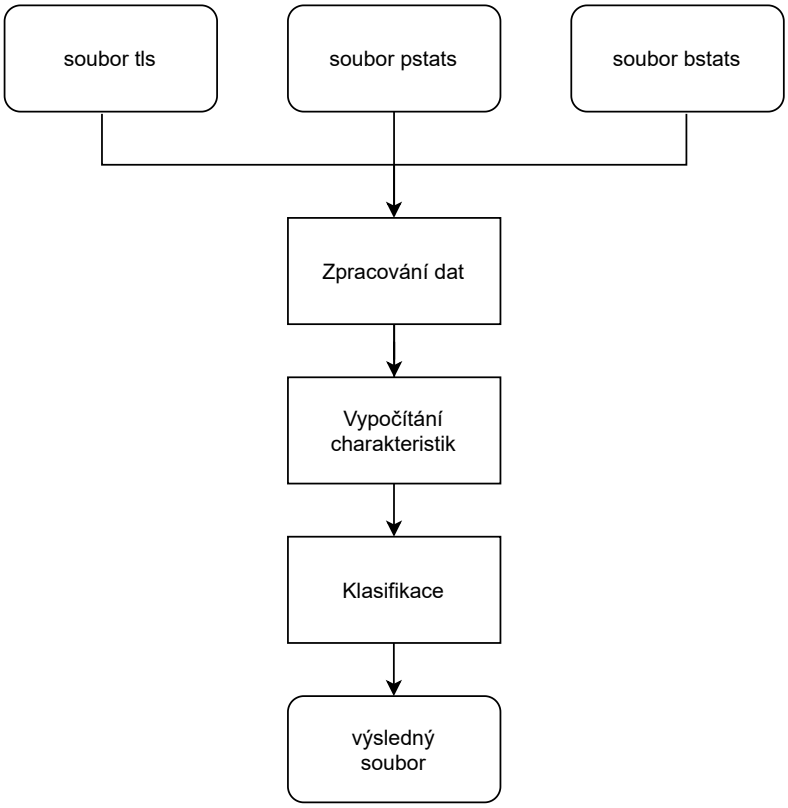
Ukázka příkazu pro spuštění prototypu z příkazové řádky:

```
$ python classifier.py tls.csv pstats.csv bstats.csv res.csv
```

4.1 Vstup a výstup

Vstupem prototypu jsou 3 soubory ve formátu CSV, které jsou výstupem pluginů modulu ipfixprobe ze systému NEMEA. Jedná se o pluginy *tls*, *pstats* a *bstats*. Díky tomu je provoz z reálné sítě již zpracován do síťových toků. Cesty k těmto souborů jsou přijaty jako první 3 argumenty, které jsou zadány při spouštění skriptu. Výstupem je soubor CSV, jehož cesta je zadána jako 4. argument. Výstupní soubor obsahuje klasifikované síťové toky, které jsou reprezentovány pomocí zdrojové a cílového IP adresy, zdrojového a cílového portu a času začátku a konce spojení, jak je vidět na obrázku 4.2.

Argumenty jsou zpracovány funkcí `process_arguments` pomocí modulu argparse. Následně jsou funkcí `load_file` nahrány vstupní soubory do struktury DataFrame modulu Pandas. Tato funkce zajišťuje, že jsou soubory nahrány korektním způsobem. Pokud je zadán špatný počet argumentů nebo je alespoň jeden ze souborů poškozený případně neobsahuje žádné spojení, je skript po vypsání chybové hlášky okamžitě ukončen.



Obrázek 4.1: Návrh prototypu

Zdrojová IP adresa	Cílová IP adresa	Zdrojový port	Cílový port	Začátek	Konec	Typ
10.0.2.15	65.9.91.5	52008	443	2021-02-19T07:47:26.429	2021-02-19T07:50:26.478	P
10.0.2.18	52.27.12.2	49196	443	2021-02-19T07:48:26.234	2021-02-19T07:52:39.431	W
10.0.2.16	151.101.14.214	59321	443	2021-02-19T07:48:59.142	2021-02-19T07:51:02.003	M

Obrázek 4.2: Ukázka výstupu prototypu

4.2 Zpracování dat

Nejdříve jsou upravená data z pluginů *pstats* a *bstats* pomocí funkcí `change_directions _pstats` a `change_directions_bstats`. Tento krok je nutný, protože může nastat případ, kdy síťový tok má zaměněné zdrojové a cílové informace.

Dále jsou informace o síťových tocích agregovány dohromady ze všech 3 vstupních souborů, přičemž jsou vyfiltrována pouze TLS spojení. Také jsou odstraněna poškozená spojení, protože je nutné počítat s chybou, která může nastat při zachytávání v síti s reálným provozem. Toto zajišťuje funkce `merge_connections`.

Protože jsou data nahrána ze souborů CSV do struktury DataFrame, jsou informace jako prvních 30 paketů uloženy ve formátu řetězce. Aby bylo možné z těchto hodnot vypočítat charakteristiky, jsou parsovány a převedeny na pole typu Numpy pomocí funkce prototypu `transform_arrays`. Stejně tak jsou upravovány časové údaje, které jsou převedeny z řetězců na struktury modulu datatime, který umožňuje práci s časovými údaji.

4.3 Vypočítání charakteristik

Když jsou vstupní data předzpracována, jsou dopočítány hodnoty nutné pro následné vypočítání charakteristik. Jedná se o dobu trvání dávek paketů a intervaly mezi jednotlivými pakety a dávkami paketů. Tyto údaje jsou vypočítány funkcí `calculate_time_features`.

Funkce `calculate_packets_features`, `calculate_bursts_features` a `calculate_PCA_features` slouží k vypočítání 16 charakteristik, které jsou nutné ke klasifikaci síťového provozu. Seznam s popisem charakteristik je uveden v podkapitole Výběr charakteristik klasifikace.

4.4 Klasifikace

Klasifikaci provádí funkce `predict`. Síťové toky jsou klasifikovány pomocí algoritmu Gradient boosting. Tento algoritmus byl zvolen, protože dosáhl nejlepšího výsledku F1 v porovnání s dalšími zkoušenými modely strojového učení. Klasifikátor byl naučen na vytvořené datové sadě, která je popsána v kapitole Tvorba datové sady, využita byla k tomu knihovna scikit-learn.

Klasifikátor je v prototypu načten ze souboru ve formátu Pickle před začátkem klasifikování jednotlivých síťových toků. Následně je každému spojení přiřazen klasifikátorem štítek skupiny. Výsledek je následně uložen do výstupního souboru.

Kapitola 5

Testování a vyhodnocení

Tato kapitola se zaměřuje na analýzu chybných predikcí klasifikátoru. Jsou popsány možné důvody, proč dochází k nesprávné klasifikaci síťových toků. Dále je provedena analýza špatně klasifikovaných spojení, u kterých jsou zkoumány anomálie v chování. V poslední části kapitoly je popsáno výkonnostní testování prototypu klasifikování provozu.

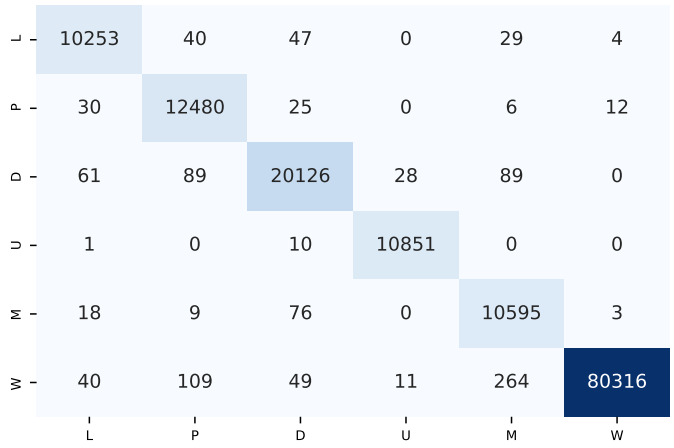
5.1 Špatné predikce modelu

Při analýze špatných predikcích je nutné zohlednit, že data pocházejí z větší části z reálné sítě. Při generování dat je možné kontrolovat, že spojení proběhlo v pořádku a nenastala chyba, toto ovšem nelze určit u spojeních z reálné sítě. Může nastat mnoho faktorů, které ovlivní průběh spojení a budou mít vliv na jednotlivé charakteristiky, kvůli čemu může dojít ke špatné predikci klasifikátoru. Zároveň je nutné se zaměřit na analýzu špatných predikcí, aby jim bylo možné předcházet a jejich množství zredukovat.

V této kapitole budou analyzovány pouze výsledky klasifikátoru, který byl použit při tvorbě prototypu. Jedná se o klasifikátor nacvičený pomocí algoritmu Gradient boosting.

5.1.1 Důvod špatných predikcí

Zachycená spojení mohou být ovlivněna chováním uživatele. Ten může například pozastavit video během sledování nebo přeskakovat skladby při poslechu hudby. Stejně tak průběh spojení závisí na kvalitě připojení uživatele. Může se stát, že uživatel má nestabilní připojení, načež může docházet k výpadkům a dalším chybám při přenosu dat. Mnoho přehrávačů videa podporuje změnu kvality videa, pokud je internetové spojení nestálé, což může dál ovlivnit množství přenesených dat a další charakteristiky. Stejně může případně kvalitu videa změnit i sám uživatel.



Obrázek 5.1: Matice záměn

Dalším možným důvodem pro špatné predikce je, že mnoho skupin se vzájemně prolíná. Například mnoho webových stránek spouští při navštívení automaticky video, i když se nejedná primárně o stránku určenou k přehrávání videa, kvůli čemu může docházet k problémům u klasifikace procházení webu a přehrávání videa. Dále některé platformy streamující video živě poskytují možnost vrátit se ve videu v čase, tudíž by mělo spojení být dále klasifikováno spíše jako přehrávání videa z přehrávače. Stejně tak bývá na některých webových stránkách možnost před stažením souboru si soubor prohlédnout, což může být například přehrání videa nebo písničky.

Také mnoho platforem jako například Spotify poskytuje možnost používat služby buď v podobě desktopové aplikace nebo přes webový prohlížeč. V některých případech se může chování spojení lišit podle způsobu použití, i když se jedná stále o stejnou službu.

5.1.2 Analýza špatných predikcí

Analýza špatných predikcí byla provedena v rámci jednotlivých klasifikovaných skupin. Nejdříve byla extrahována chybně klasifikovaná spojení, charakteristiky těchto spojení byly následně porovnávány s charakteristikami správně klasifikovaných spojení a hodnot charakteristik celé datové sady. Na základě těchto dat bylo zkoumáno, jaké typy spojení byly nejčastěji zaměňovány a jaké anomálie v chování případně tyto spojení měla.

5.1.2.1 Přehrávání videa živě

Tato skupina bývá nejčastěji zaměňována za stahování souboru a přehrávání videa z přehrávače.

Špatně predikovaná spojení této skupiny se vyznačují malými velikostmi dávek paketů, což může být důvod, proč dochází k chybným predikcím, protože jsou pro tuto skupinu typické velké dávky paketů, má je ze všech skupin téměř největší. Zároveň dávky paketů u chybně klasifikovaných spojení trvají několikanásobně kratší dobu a jsou mezi nimi i kratší intervaly, jak je vidět v grafu 5.3. První sloupec popisuje průměrné hodnoty všech spojení v datasetu, v dalším jsou pouze správně klasifikovaná a v posledním naopak špatně určená. Klesá i hodnota průměrné velikosti paketů ve spojení u špatných predikcí, délky intervalů mezi nimi zůstávají však stejné.

Byl proveden experiment, kdy spojení ze skupiny přehrávání videa živě a přehrávání videa z přehrávače byly sjednoceny do jedné skupiny a označeny stejným štítkem. Následně byl opět nacvičen klasifikátor, který dosáhl výsledku F1 0,991.

5.1.2.2 Přehrávání videa z přehrávače

Spojení z této skupiny bývají nejčastěji zaměněna za přehrávání videa živě, následně za stahování souboru a velmi zřídka za procházení webu.

Chybně klasifikovaná spojení z kategorie přehrávání videa z přehrávače se liší hlavně v chování dávek paketů, jejich délka je průměrně mnohonásobně větší. Množství přenesených dat je v jednotlivých dávkách paketů také několikanásobně větší. Tyto charakteristiky mohou způsobovat častou záměnu se skupinou přehrávání videa živě, protože velkými dávkami paketů se tato skupina vyznačuje. Dále je v rámci spojení přeneseno celkově menší množství dat, což je pravděpodobně způsobeno průměrně delšími intervaly mezi jednotlivými pakety.

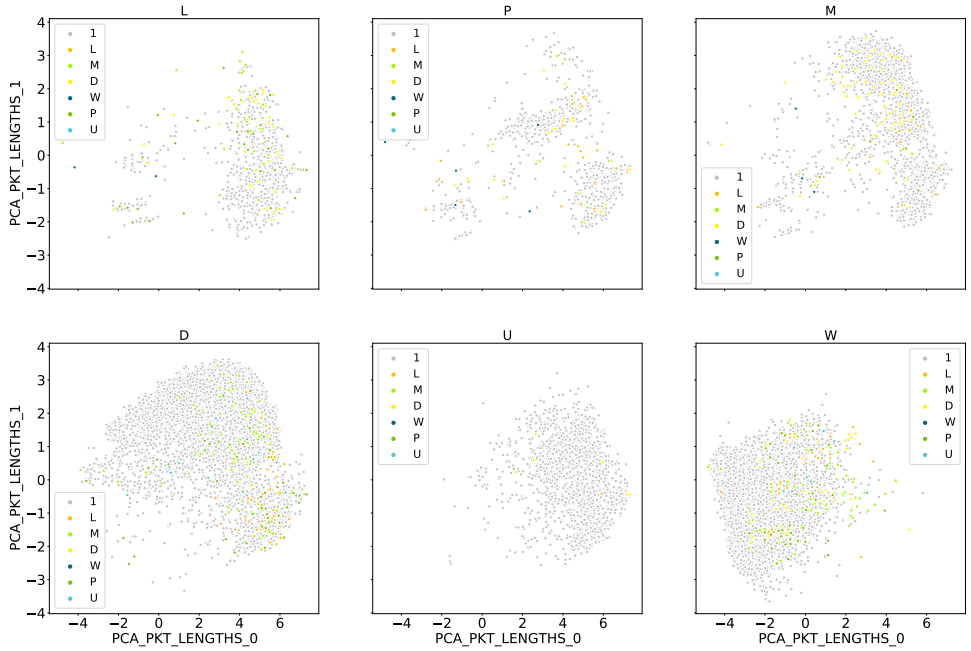
5.1.2.3 Přehrávání hudby

Skupiny přehrávání hudby a stahování souboru bývají vzájemně mezi sebou často zaměňovány. Dále velmi často bývá kategorie web klasifikována jako skupina poslouchání hudby. Tyto špatné predikce mohou být zapříčiněny tím, že se jedná o skupiny, které mají v průměru nejmenší množství přenesených dat.

Když je spojení patřící do skupiny poslouchání hudby špatně predikované, je v něm průměru přeneseno mnohonásobně více dat. Průměrná velikost paketů zůstává stejná, ale intervaly mezi nimi jsou kratší. To způsobuje, že i velikost dávek paketů je mnohonásobně větší a zároveň trvají delší dobu, naopak intervaly mezi nimi výrazně klesají.

5.1.2.4 Přenos souboru

Z matice záměn 5.1 lze vyčíst, že často dochází ke špatné predikci spojení, které by mělo patřit do skupiny stahování souboru. Nejvíce bývá zaměňováno



Obrázek 5.2: Ukázka špatných predikcích podle skupin na charakteristikách PCA délek paketů

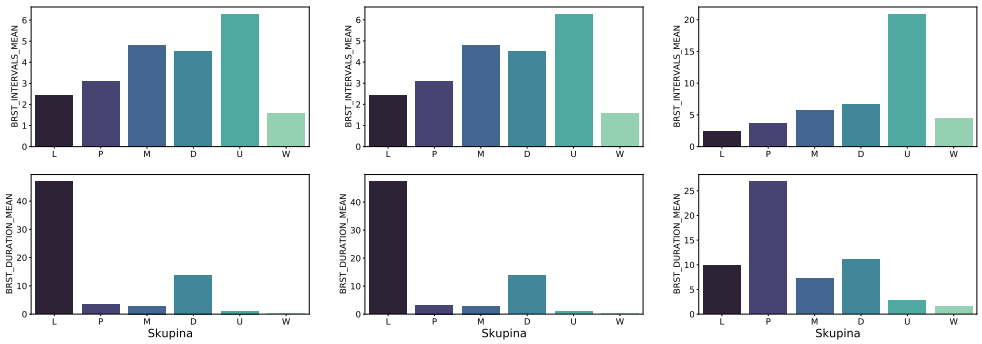
za kategorii poslouchání hudby, dále potom za přehrávání videa z přehrávače a přehrávání videa živě.

Špatně predikovaná spojení skupiny stahování mají v průměru delší dávky paketů, ovšem stejné množství přenesených dat jako ty správně predikovaná, což znamená, že průměrná velikost dávek paketů u chybě klasifikovaných je menší. Vzhledem k tomu, že pro stahování jsou charakteristické průměrně největší dávky paketů ze všech skupin a u chybně určených spojení klesá průměr téměř o čtvrtinu, může docházet ke špatným predikcím kvůli tomuto faktoru, protože průměry ostatních charakteristik zůstávají přibližně stejné.

Naopak zřídka chybně predikovanou skupinou je nahrávání souboru, což je nejspíše způsobeno tím, že jako jediná kategorie má obvykle u spojení více odeslaných bajtů než přijatých.

5.1.2.5 Procházení webu

Spojení patřící do této skupiny jsou nejčastěji špatně predikována. Nejvíce jsou zaměňována za spojení patřící do kategorie poslech hudby, následně za přehrávání videa z přehrávače. Naopak dochází velmi málokdy na chybnou predikci ostatních skupin na tuto, což je nejspíše způsobeno tím, že tato sku-



Obrázek 5.3: Časové údaje dávek paketů podle úspěšnosti predikce

pina se výrazně liší průměrným množstvím přenesených dat, které je obvykle mnohonásobně menší oproti ostatním skupinám.

Špatně predikovaná spojení patřící k procházení webu mají v průměru přeneseno mnohonásobně větší množství dat oproti správně klasifikovaným. Stejně tak je průměrná velikost paketů téměř dvakrát větší, ovšem intervaly mezi nimi mnohonásobně kratší. Odlišnost velikostí paketů je také vidět u charakteristik hodnot PCA, které jsou znázorněny v grafu 5.2. Správné predikce jsou v něm vykresleny světle šedou barvou, špatné mají barvu skupiny, za kterou byly chybně označeny. Podobné charakteristiky mají i dávky paketů, které jsou delší a větší, intervaly zůstávají ovšem stejné.

5.2 Testování

Testování prototypu bylo provedeno na serveru s procesorem Intel Xeon E5-2609 v3 a 256 GB operační paměti. Na vstupu byla data z reálného síťového provozu, která byla předzpracována modulem ipfixprobe a uložená ve formátu CSV. K testování byly využity tři sady souborů obsahující 13 000, 110 000 a 165 000 síťových toků.

Byl měřen běh celého programu a zároveň jednotlivých funkcí. Výsledky jsou uvedeny v tabulce 5.1. Nejpomalejší funkcí prototypu je funkce `transform_arrays`, což je způsobeno nutností parsování hodnot z řetězců. Potřeba parsování dat je ovšem pouze v prototypu, protože v případě nasazení v provozu budou data přijímána přímo jako hodnoty potřebného typu, tudíž je nebude nutné převádět. Tím bude i mnohonásobně zvýšeno množství zpracovaných toků za vteřinu. Prototyp je nyní schopný zpracovat průměrně 500 síťových toků za vteřinu.

Tabulka 5.1: Časové nároky funkcí prototypu v sekundách podle množství zpracovaných spojení

Funkce	13 000	110 000	165 000
process_arguments	0,001	0,002	0,001
load_file	1,229	5,346	5,727
change_directions_pstats	0,075	0,173	0,182
change_directions_bstats	0,039	0,190	0,193
merge_connections	0,354	1,741	2,571
transform_arrays	9,618	88,054	131,584
calculate_time_features	2,218	16,163	24,421
calculate_packets_features	6,014	49,551	79,242
calculate_bursts_features	4,281	35,933	59,000
calculate_pca_features	1,016	15,769	37,077
predict	1,486	9,359	16,433

Závěr

Cílem bakalářské práce bylo vytvořit anotovanou datovou sadu TLS provozu a implementovat softwarový prototyp, který bude schopen zpracovávat provoz z reálné sítě a klasifikovat akce přenášené skrz TLS spojení na bázi síťových toků. Součástí práce je také analýza síťového provozu TLS se zaměřením na charakteristiky chování spojení jednotlivých skupin.

Nejprve bylo zadefinováno 6 skupin, které jsou rozlišovány v síťovém provozu. Pro každou skupiny následně byly vybráni zástupci. Síťový provoz těchto zvolených služeb byl získáván 2 způsoby. Část dat byla získána generováním, jak manuálním, tak automatizovaným způsobem. Další část dat pochází z reálného síťového provozu, který byl zachycen na perimetru sítě CESNET2. Dohromady bylo získáno přes 8 terabajtů dat.

Získaný síťový provoz byl následně zpracován systémem NEMEA. Výsledné síťové toky byly agregovány z výstupních souborů systému, filtrovány a anotovány. Výsledná datová sada obsahuje 145 671 anotovaných síťových toků, přičemž každý z nich je definován 75 charakteristikami popisující chování spojení.

Pro klasifikování síťového provozu bylo vyzkoušeno několik algoritmů strojového učení, přičemž pro dosaženích co nejpřesnějšího výsledku byly laděny jejich hyperparametry a eliminovány přebytečné charakteristiky. Nejlepšího výsledku dosáhl model Gradient Boosting, který měl skóre F1 0,987. Dosáhl vyššího skóre než klasifikátory popsané v podkapitole Klasifikace v monitorování. Zároveň jako jediný pro klasifikaci využívá dávky paketů.

Špatné predikce klasifikátoru byly detailně analyzovány, aby bylo zjištěno, proč dochází k chybné klasifikaci a jaké skupiny bývají nejčastěji zaměňovány. Dále byly uvedeny možné příčiny, proč k těmto chybám dochází.

Výše zmiňovaný model byl použit v prototypu na klasifikování TLS síťového provozu. Ten na vstupu přijímá soubory se síťovými toky, pro které dopočítá nutné charakteristiky pro klasifikaci a následně jim přidělí štítek skupiny pomocí naučeného modelu. Nevyužívá přitom rozšíření SNI, tudíž zvládne kla-

sifikovat síťové toky i v případě, že jméno serveru bude zašifrované. Zároveň bylo provedeno výkonnostní testování tohoto prototypu.

V budoucnu by bylo možné rozšířit množství klasifikovaných skupin a také jejich zástupců, čím by byla zároveň zvětšena datová sada. Další možností by bylo omezit množství špatných predikcí na základě provedené analýzy.

Předpokládá se, že prototyp bude převeden na modul systému NEMEA a využit pro klasifikaci síťového provozu. Dále budou také výsledky této bakalářské práce publikovány jako konferenční článek.

Literatura

[1] Maddison, J.: Encrypted Traffic Reaches A New Threshold. Nov 2018, [Citováno 12-04-2021]. Dostupné z: https://www.networkcomputing.com/network-security/encrypted-traffic-reaches-new-threshold

[2] DDOS: HTTPS encryption traffic on the Internet has exceeded 90 percent. Nov 2019, [Citováno 15-03-2021]. Dostupné z: https://meterpreter.org/https-encryption-traffic/

[3] Rescorla, E.; Oku, K.; Sullivan, N.; aj.: TLS Encrypted Client Hello. Internet-Draft draft-ietf-tls-esni-10, Internet Engineering Task Force, Březen 2021, work in Progress. Dostupné z: https://datatracker.ietf.org/doc/html/draft-ietf-tls-esni-10

[4] Cejka, T.; Bartos, V.; Svepes, M.; aj.: NEMEA: A Framework for Network Traffic Analysis. In *12th International Conference on Network and Service Management (CNSM 2016)*, 2016, doi:10.1109/CNSM.2016.7818417. Dostupné z: http://dx.doi.org/10.1109/CNSM.2016.7818417

[5] Rescorla, E.; Dierks, T.: The transport layer security (TLS) protocol version 1.3. 2018.

[6] IBM: TLS protocol overview. [Citováno 10-03-2021]. Dostupné z: https://www.ibm.com/support/knowledgecenter/en/SSYKE2_7.1.0/com.ibm.java.security.component.71.doc/security-component/jsse2Docs/ssloverview.html

[7] Blake-Wilson, S.; Nystrom, M.; Hopwood, D.; aj.: Transport layer security (TLS) extensions. *Request for Comments*, ročník 3546, 2003.

[8] Moriarty, K.; Farrell, S.: Deprecating TLS 1.0 and TLS 1.1. 2021.

[9] Brownlee, N.; Mills, C.; Ruth, G.: Traffic flow measurement: Architecture. Technická zpráva, RFC 2722, 1999.

[10] Claise, B.; Sadasivan, G.; Valluri, V.; aj.: Cisco systems netflow services export version 9. 2004.

[11] Claise, B.; Trammell, B.; Aitken, P.: Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information. *RFC 7011 (Internet Standard), Internet Engineering Task Force*, 2013: s. 2070–1721.

[12] John B. Althouse, J. A., Jeff Atkinson: ja3. [Citováno 11-03-2021]. Dostupné z: `https://github.com/salesforce/ja3`

[13] Cesnet: CESNET/ipfixprobe. [Citováno 11-03-2021]. Dostupné z: `https://github.com/CESNET/ipfixprobe`

[14] Cisco: Joy. [Citováno 11-03-2021]. Dostupné z: `https://github.com/cisco/joy`

[15] Draper-Gil, G.; Lashkari, A. H.; Mamun, M. S. I.; aj.: Characterization of encrypted and vpn traffic using time-related. In *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, 2016, s. 407–414.

[16] Minarik, P.: Part 3: Network Traffic Monitoring or Packet Analysis? Get the Benefits of Both. [Citováno 29-03-2021]. Dostupné z: `https://www.flowmon.com/getattachment/1777528d-9f95-4661-9dc2-f53f04e7065b/network-traffic-monitoring-or-packet-analysis.aspx`

[17] Suricata: Suricata. [Citováno 13-03-2021]. Dostupné z: `https://suricata-ids.org/`

[18] Wireshark: Wireshark. [Citováno 11-03-2021]. Dostupné z: `https://www.wireshark.org/`

[19] Cejka, T.; Bartos, V.; Svepes, M.; aj.: NEMEA. [Citováno 10-03-2021]. Dostupné z: `https://nemea.liberouter.org/`

[20] Vaclav Bartos, T. C., Martin Zadnik: Nemea: Framework for stream-wise analysis of network traffic. 2013.

[21] Wang, P.; Ye, F.; Chen, X.; aj.: Datanet: Deep Learning Based Encrypted Network Traffic Classification in SDN Home Gateway. *IEEE Access*, ročník 6, 2018: s. 55380–55391, doi:10.1109/ACCESS.2018.2872430.

[22] Pan, W.; Cheng, G.; Tang, Y.: WENC: HTTPS Encrypted Traffic Classification Using Weighted Ensemble Learning and Markov Chain. In *2017 IEEE Trustcom/BigDataSE/ICESS*, 2017, s. 50–57, doi:10.1109/Trustcom/BigDataSE/ICESS.2017.219.

[23] Habibi Lashkari, A.; Draper Gil, G.; Mamun, M.; aj.: Characterization of Encrypted and VPN Traffic Using Time-Related Features. 02 2016, doi:10.5220/0005740704070414.

[24] Rusheng Ding; Wenmin Li: A hybrid method for service identification of SSL/TLS encrypted traffic. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, 2016, s. 250–253, doi:10.1109/CompComm.2016.7924703.

[25] Ding, L.; Yu, F.; Peng, S.; aj.: A Classification Algorithm for Network Traffic based on Improved Support Vector Machine. *Journal of Computers*, ročník 8, 04 2013, doi:10.4304/jcp.8.4.1090-1096.

[26] Moz, I.: Top 500 Most Popular Websites. [Citováno 13-03-2021]. Dostupné z: `https://moz.com/top500`

[27] Alexa: The top 500 sites on the web. [Citováno 13-03-2021]. Dostupné z: `https://www.alexa.com/topsites`

[28] scikit-yb developers, T.: Recursive Feature Elimination. [Citováno 22-03-2021]. Dostupné z: `https://www.scikit-yb.org/en/latest/api/model_selection/rfecv.html`

[29] Wold, S.; Esbensen, K.; Geladi, P.: Principal component analysis. *Chemometrics and intelligent laboratory systems*, ročník 2, č. 1-3, 1987: s. 37–52.

[30] scikit-learn developers: sklearn.ensemble.GradientBoostingClassifier. [Citováno 23-04-2021]. Dostupné z: `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html`

[31] Brownlee, J.: Undersampling Algorithms for Imbalanced Classification. Jan 2021, [Citováno 22-03-2021]. Dostupné z: `https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/`

[32] Breiman, L.: Random forests. *Machine learning*, ročník 45, č. 1, 2001: s. 5–32.

[33] scikit-learn developers: Ensemble methods. [Citováno 22-03-2021]. Dostupné z: `https://scikit-learn.org/stable/modules/ensemble.html#forest`

[34] Natekin, A.; Knoll, A.: Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, ročník 7, 2013: str. 21.

[35] scikit-learn developers: Nearest Neighbors. [Citováno 22-03-2021]. Dostupné z: `https://scikit-learn.org/stable/modules/neighbors.html`

[36] scikit-learn developers: Cross-validation. [Citováno 18-04-2021]. Dostupné z: `https://scikit-learn.org/stable/modules/cross_validation.html`

[37] Claesen, M.; De Moor, B.: Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*, 2015.

[38] Raschka, S.: An overview of general performance metrics of binary classifier systems. *arXiv preprint arXiv:1410.5330*, 2014.

Příloha A

Seznam použitých zkratek

CSV Comma-separated values

FTP File Transfer Protocol

HTTP Hypertext Transfer Protocol

IDS Intrusion Detection System

IETF Internet Engineering Task Force

IMAP Internet Message Access Protocol

PCA Principal Component Analysis

SVM Support Vector Machines

VPN Virtual Private Network

Příloha **B**

Obsah přiloženého USB

- `readme.txt` ..... stručný popis obsahu USB
- `dataset` ..... vytvořená anotovaná datová sada
- `src`
  - `impl` ..... zdrojový kód implementace prototypu
  - `notebooks` ..... notebooky Jupyter s experimenty a analýzou
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `doc` ..... dokumentace prototypu
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF