VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA DNS DAT PRO ÚČELY IDENTIFIKACE MOBILNÍCH ZAŘÍZENÍ

DNS DATA ANALYSIS FOR MOBILE DEVICE IDENTIFICATION PURPOSES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **ALEX SPORNI**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. IVANA BURGETOVÁ, Ph.D.**
SUPERVISOR

BRNO 2020

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav informačních systémů (UIFS) Akademický rok 2019/2020

Zadání bakalářské práce

Student: **Sporni Alex**
Program: Informační technologie
Název: **Analýza DNS dat pro účely identifikace mobilních zařízení**
DNS Data Analysis for Mobile Device Identification Purposes
Kategorie: Data mining

Zadání:

1. Seznamte se s protokolem DNS.
2. Seznamte se s možnostmi identifikace mobilních zařízení prostřednictvím informací zjištěných ze síťové komunikace těchto zařízení.
3. Seznamte se s dostupnými datovými sadami, které obsahují reálnou komunikaci mobilních zařízení. Tyto datové sady vhodným způsobem předzpracujte pro další použití.
4. Po dohodě s vedoucí vyberte vhodnou metodu z oblasti dolování dat, kterou lze použít pro identifikaci mobilních zařízení na základě dat obsažených v DNS komunikaci.
5. Zvolenou metodu implementujte a otestujte na dostupných datových sadách.
6. Zhodnoťte dosažené výsledky.

Literatura:

- Matoušek, Petr: Síťové aplikace a jejich architektura, Brno: VUTIUM, 2014, 396 s., ISBN 978-80-214-3766-1.
- Kurtz, A.,Gascon, H., Becker, T., Rieck, K., Freiling, F.: Fingerprinting Mobile Devices Using Personalized Configurations, In Proceedings on Privacy Enhancing Technologies, PoPETS, 2016 (1), p. 4-19.
- Han, J., Kamber, M.: Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufmann Publishers, 2012, 703 p., ISBN 978-0-12-381479-1

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Burgetová Ivana, Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2019
Datum odevzdání: 28. května 2020
Datum schválení: 24. října 2019

Abstrakt

Táto bakalárska práca sa zaoberá identifikáciou mobilných zariadení na základe analýzy DNS dát. Poskytuje teoretický úvod k modelu počítačovej komunikácie. Popisuje význam protokolu DNS z pohľadu sieťovej komunikácie medzi zariadeniami. V práci sú predstavené poskytnuté dátové sady, ktoré obsahujú reálnu komunikáciu mobilných zariadení. Jednotlivé dátové sady je potrebné predspracovať a uložiť do SQL databázy za účelom jednoduchšej manipulácie v neskorších fázach implementácie. Práca popisuje jednotlivé techniky spracovania dát. V práci sú podrobne popísané metodiky hodnotenia relevancie TF-IDF a aplikácia kosínusovej podobnosti pri identifikácii mobilných zariadení. Výstupom práce je zhodnotenie dosiahnutých výsledkov.

Abstract

This bachelor's thesis deals with the problem of identification of mobile devices based on DNS data analysis. The thesis provides a theoretical introduction to the computer communication model. This thesis explains the importance of DNS in the terms of network communication between devices, It also presents the provided data sets, which contain real communication of mobile devices. These data sets must be with a suitable technique parsed and stored in a database to provide better data manipulation techniques in the later stages of implementation. This work further describes individual techniques of data processing. It also depicts in detail the methodologies for evaluating the relevance of TF-IDF and the application of cosine similarity to identify the mobile devices. The main output of this work is the evaluation of the achieved results.

Kľúčové slová

Dolovanie dát, analýza dát, metodológia, TF-IDF, kosínusová podobnosť, dátové sady, protokol DNS, mobilné zariadenia, identifikácia, Euklidovská vzdialenosť, Manhattanovská vzdialenosť, Minkowského vzdialenosť, Chebyshevova vzdialenosť, váhovanie, vektorový model, Python, MySQL, techniky spracovania dát, model OSI, model TCP/IP, DNS záznamy

Keywords

Data mining, data analysis, methodology, TF-IDF, cosine similarity, data sets, DNS protocol, mobile devices, identification, Euclidean distance, Manhattan distance, Minkowski distance, Chebyshev distance, weighting, vector space model, Python, MySQL, data processing techniques, OSI model, TCP/IP model, DNS resource records

Citácia

SPORNI, Alex. *Analýza DNS dat pro účely identifikace mobilních zařízení*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Ivana Burgetová, Ph.D.

Analýza DNS dat pro účely identifikace mobilních zařízení

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pani Ing. Ivany Burgetovej, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Alex Sporni
28. mája 2020

Poďakovanie

Chcel by som sa poďakovať svojej vedúcej bakalárskej práce Ing. Ivane Burgetovej, Ph.D. za odborné vedenie a cenné rady, ktoré mi pri riešení práce poskytla.

Obsah

Kapitola 1

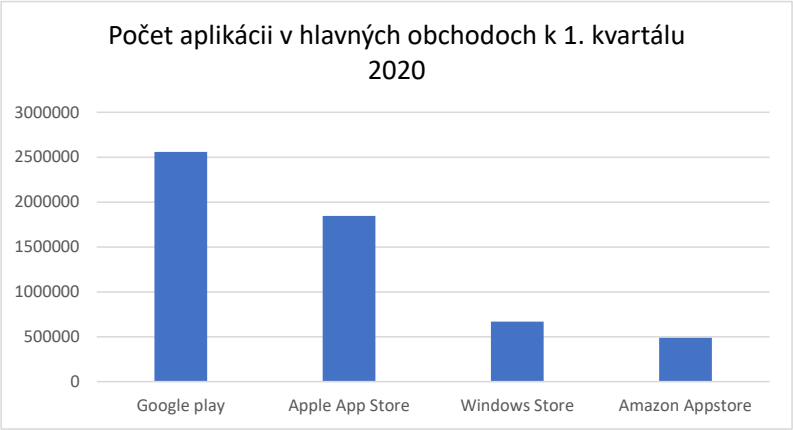
Úvod

Informatizácia našej spoločnosti podstatne rozšírila naše možnosti pre generovanie a zber dát z rozmanitých zdrojov. Obrovské množstvá dát zaplavili rôzne aspekty našich každodenných životov. Tento explozívny nárast v ukladaní a analyzovaní dát vyvolal urgentnú potrebu po novej technike a automatizovaných nástrojoch, ktoré by nám inteligentne mohli asistovať v transformácii tohto obrovského množstva dát na užitočné informácie a znalosti. Tento trend viedol k vývoju sľubného a prosperujúceho odvetvia v informačných technológiách nazývaného *data mining*. Populárne synonymom pre data mining nesie názov *knowledge discovery from data (KDD)* alebo tzv. získavanie znalostí z dát. Je to automatizované a praktické extrahovanie dát na základe vzorcov, ktoré reprezentujú znalosti z uložených alebo zachytených dát, ktoré sú väčšinou uložené v rozsiahlych databázach, dátových centrách, v rôznych dátových tokoch alebo na samotnom internete.

Cieľom tejto práce je identifikovať mobilné zariadenia na základe analýzy poskytnutých DNS dát z dátových sád. V kapitole 2 bude popísaný význam systému DNS z hľadiska sieťovej komunikácie medzi zariadeniami, budú popísané typy jednotlivých DNS dotazov. V kapitole 3 budú predstavené jednotlivé metódy a matematické postupy, ktoré slúžia k určovaniu podobnosti číselných dát, následne bude predstavená metóda, ktorá bude použitá k identifikácii mobilných zariadení. V kapitole 4 budú rovnako predstavené aj jednotlivé nástroje, ktoré boli počas vývoja skriptu využívané, ako jazyk *Python* a relačný databázový server *MySQL*, a vizualizujúci nástroj *MySQL Workbench*.. Kapitola 5 sa bude zaoberať popisom obdržaných dátových sád, spôsobom a technikami spracovania dát, medzi ktoré patrí čistenie dát, transformácia dát, integrácia dát a redukcia dát. V kapitole 6 bude popísaný návrh riešenia a samotná implementácia skriptu, ktorý bude slúžiť k rozboru, analyzovaniu a vyhodnoteniu dát. V kapitole 7 budú vyhodnotené experimenty prevedené nad dátovými sadami. V záverečnej kapitole 8 budú zhrnuté dosiahnuté výsledky výskumu a možnosti rozšírenia práce.

1.1 Rozšírený úvod do problematiky a identifikácie mobilných zariadení

V polovici roku 2014 bolo prístupných viac ako 1.2 milióna aplikácií na App Store (Apple) a viac ako 1.3 milióna aplikácií na Play Store (Google), z toho bolo viac ako 1.1 milióna bezplatných. V prvom kvartáli roka 2020 tento počet dramaticky vzrástol, ako je možné vidieť na obrázku 1.1. Bezplatné aplikácie často vzbudzujú otázky ohľadom biznis modelu. Ak stiahnutie aplikácie je v podstate zadarmo, v mnohých prípadoch sa užívatelia bez svojho vedomia stávajú subjektmi marketingu, ktorým sú vystavované reklamy s cieľom finančného profitu aplikácie [8].



Obr. 1.1: Počet aktuálne dostupných aplikácií v jednotlivých obchodoch [3]

V minulosti boli prevedené štúdie, ktoré dokázali, že okolo polovice všetkých aplikácií na App Store obsahovali sledovacie knižnice a zobrazovanie reklám. S vysokou pravdepodobnosťou sa dá predpokladať, že podobné aplikácie sa nachádzali aj na Play store. Tieto knižnice mali za úlohu sledovať správanie sa užívateľov a zbierať o nich citlivé informácie. V niektorých prípadoch aplikácie dokonca posielali tieto osobné údaje reklamným sieťam, čo im umožňovalo poskytovať cielené reklamy *šité na mieru* alebo dokonca predať tieto údaje s cieľom finančného zisku. Väčšina užívateľov nemá vedomosti o takejto aktivite a tým pádom sa nemôže vyhnúť takémuto zásahu do súkromia [8].

Za účelom priradenia týchto nazbieraných citlivých údajov k jednotlivým užívateľom, musí byť užívateľské zariadenie jednoznačne identifikované. Tento proces identifikácie zariadenia sa nazýva *device fingerprinting*. Existuje viacero spôsobov, ako si vytvoriť takýto indentifikátor (fingerprint), pri zariadeniach Android je známym faktom, že povolujú prístup k jednoznačnému identifikátoru zariadenia (IMEI pre GSM a MEID/ESN pre CDMA mobilné zariadenia), čo ich činí triviálne jednoduchými k identifikácii. Až do vydania iOS 7[1] bolo možné taktiež získať niektoré hardvérové identifikátory, okrem unikátneho identifikátora *Unique Device Identifier (UDID)*, ktorý okrem iných elementov je založený na sériovom čísle zariadenia. Rovnako aj MAC adresa WiFi antény bola v mnohých prípadoch používaná k identifikácii užívateľa. Ako odpoveď na tieto závažné bezpečnostné hrozby, spoločnosť Apple zakázala v roku 2013 akýkoľvek prístup k hardvérovým identifikátorom. V nasledujúcich kapitolách budú predstavené možnosti identifikácie mobilných zariadení aj napriek týmto snahám o anonymizáciu, predovšetkým na základe DNS komunikácie [8].

[1]Changelog k aktualizácii iOS 7 https://support.apple.com/en-us/HT207979#7

Kapitola 2

Model sieťovej komunikácie

Ľudské bytosti môžu byť identifikované viacerými spôsobmi. Napríklad môžu byť identifikované svojim menom a priezviskom, ktoré sa nachádzajú v rodnom liste. Môžu byť identifikované číslom občianskeho alebo vodičského preukazu, poprípade číslom pasu. Rovnako ako ľudské bytosti, tak aj užívatelia/zariadenia na internete, môžu byť identifikovaný rôznymi spôsobmi. Jedným z identifikátorov pre zariadenie môže byť doménové meno (domain name). Doménové mená sú mnemotechnické (ľahko zapamätateľné), napr. `vutbr.cz`, `yahoo.com`. Názvy hostiteľov však poskytujú len málo informácií o umiestnenení zariadenia na internete. Doménové meno ako `www.eurecom.fr` môžu napovedať, že zariadenie sa pravdepodobne nachádza vo Francúzku, ale to je asi tak všetko. Okrem toho, doménové mená môžu pozostávať z variabilnej dĺžky alfanumerických znakov a smerovače (routre) by ich mohli len ťažko spracovať. Zo zmienených dôvodov sú zariadenia na internete identifikované pomocou *IP adries* [7].

IP adresa je logický číselný identifikátor daného uzla (počítača, smerovača, servera atď. . .) v sieti. Adresa IP verzie 4 je 32-bitové číslo, takže teoreticky existuje 2^{32} (4 294 967 296) možných adries. Z praktických dôvodov sa toto číslo delí na štyri 8-bitové čísla (číslo v rozsahu 0-255), ktoré sa zapisujú dekadicky a sú oddelené bodkou, napr. 147.229.9.26 (`https://www.fit.vut.cz`) [13]. V sekcii 2.1 budú predstavené dva základné modely počítačových sietí TCP/IP a OSI. V sekcii 2.2 a 2.3 bude predstavený a podrobne popísaný protokol DNS. V sekcii 2.4 budú opísané jednotlivé DNS záznamy.

2.1 Model počítačovej siete

Správne pochopenie vrstvovej architektúry počítačových sietí je nevyhnutným základom, ako porozumieť činnosti sieťových služieb. Každá sieťová služba (ako elektronická pošta alebo DNS) využíva služby ďalších sieťových vrstiev (správne doručenie dát, kódovanie, oprava chýb) pre svoju činnosť. Architektúra počítačových sietí bola od svojho počiatku navrhnutá tak, aby odlišovala tri základné časti komunikačného systému [9]:

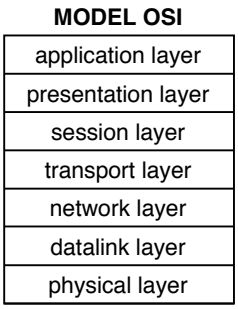
1. technológie pre prenos signálu (metalická kabeláž, rádiové vlny, optické vlákna),
2. vrstvu pre zaistenie spoľahlivosti prenosu (TCP/IP, IPX/SPX, AppleTalk, atď. . .),
3. aplikačnú vrstvu, ktorá poskytuje služby užívateľovi.

Behom historického vývoja počítačových sietí vzniklo niekoľko modelov architktúry počítačových sietí postavených na firemných protokoloch. V súčasnej dobe sa používajú v ob-

lasti internetu dva modely. Model ISO/OSI, ktorý slúži ako referenčný model a model TCP/IP, ktorý je založený na protokoloch TCP (Transmission Control protocol) a IP (Internet Protocol). Model TCP/IP predstavuje štandard dnešného internetu a dnes je implementovaný ako prenosová vrstva väčšiny počítačových sietí [9].

2.1.1 Model OSI

Pre popis sieťovej architektúry sa používa referenčný model OSI (Open Systems Interconnect Reference Model), ktorý je definovaný Medzinárodnou štandardizačnou organizáciou ISO[1] (International Standards Organization) od roku 1987 ako rámec štandardov pre komunikáciu po sieti medzi rôznymi zariadeniami a aplikáciami rôznych výrobcov. Model OSI sa v dnešnej dobe dá považovať za primárnu architektúru pre počítačovú komunikáciu. Mnoho dnešných protokolov má štruktúru vytvorenú práve podľa modelu OSI, ktorý je možné vidieť na obrázku 2.1. V praxi sa implementovala iba časť tohto modelu, plná implementácia nikdy nebola vo väčšej miere nasadená do prevádzky [9].

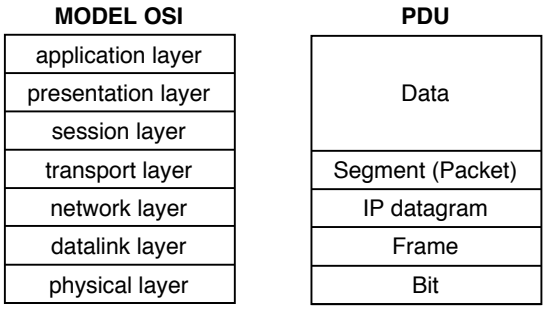


Obr. 2.1: Grafická ukážka sedemvrstvového referenčného modelu OSI

Referenčný model OSI tvorí sedem vrstiev, ktoré sú znázornené na obrázku 2.1. Tieto vrstvy definujú služby príslušnej vrstvy a zodpovedajúce protokoly. Každá vrstva modelu OSI popisuje funkcie pre prenos dát medzi procesmi rovnakej vrstvy (tzv. partnerská komunikácia). Vrstva nedefinuje iba jeden protokol, ale funkcie dátovej komunikácie, ktoré majú byť prevádzané protokolmi danej vrstvy. Pri komunikácii využíva daná vrstva služby nižších vrstiev bez toho, aby musela vedieť, ako sa tieto nižšie vrstvy chovajú. Prenos po sieti je z pohľadu danej vrstvy záležitosť najbližšej vrstvy [9].

Na obrázku 2.2 je možné vidieť základné dátové jednotky na jednotlivých vrstvách modelu OSI (PDU - Process Data Unit). Pokiaľ prebieha komunikácia medzi aplikáciami, dochádza k prenosu dát (správ) medzi komunikujúcimi programami. Tieto programy sa nestarajú o adresovanie, vyhľadávanie cesty, naväzovanie spojenia atď. . . čo je úlohou nižších vrstiev. Zoberme si užívateľa, ktorý posiela dáta pomocou aplikačného programu. Dáta sú transportnou vrstvou rozdelené na pakety TCP alebo UDP a sú predané sieťovej vrstve k doručeniu. Sieťová vrstva zapúzdri pakety do IP datagramov a predá ich sieťovému rozhraniu (napr. sieťová karta v počítači). Sieťové rozhranie prevedie dáta do rámcov (pridá fyzickú adresu)a zapíše ich na prenosové médium (metalická kabeláž, optické vlákno alebo WiFi). Prenosové médium prenáša dáta vo forme bitov, ktoré sú zakódované na určitú fyzikálnu veličinu podľa typu dátového média (napr. hladina elektrického napätia, frekvencia rádiového žiarenia, vlnová dĺžka svetelného toku) [9].

[1] https://www.iso.org/standards.html



Obr. 2.2: Grafická ukážka základných dátových jednotiek na jednotlivých vrstvách modelu OSI

Vrstvy modelu OSI

V tejto sekcii bude uvedený stručný popis jednotlivých vrstiev modelu OSI [9].

1. **Fyzická vrstva (physical leyer)**: Definuje fyzické vlastnosti linky, napr. napäťové charakteristiky, počet a umiestnenie pinov na konektore, kódovanie úrovní napätí pre logickú 0 a 1, fyzickú rýchlosť prenosového média a podobne. Medzi protokoly fyzickej vrstvy patria štandardy RS-232C (sériové rozhranie), V.35, IEEE 802.3 (CSMA/CD), IEEE 802.11 (WiFi), 802.5 Token Ring a mnoho ďalších...

2. **Linková vrstva (datalink layer)**: Popisuje prenos dát po konkrétnej dátovej linke, vytváranie rámcov, adresovanie na linkovej vrstve. Popisujú ju štandardy 802 (fyzické aj linkové), ISO 8802/2, 8802/3, 8802/4 a 8802/5 (zodpovedajúce štandardom IEEE 802.x), protokoly Ethernet, PPP, Token Ring, Frame Relay, SDLC, HDLC a ďalšie.

3. **Sieťová vrstva (network layer)**: Definuje protokol X.25 a štandardy ISO 8878, ISO 8473. Na tejto vrstve pracujú smerovacie protokoly ES-IS
(End System-to-Intermediate System) a IS-IS. Spojované služby sú popísané protokolmi CONP (Connection-Oriented Network Protocol) a CMNS (Connection-Mode Network Service), nespojované služby používajú protokol CLNP (Connectionless Network Protocol) a službu CLNS (Connectionless Network Service).

4. **Transportná vrstva (transport layer)**: Garantuje spoľahlivý prenos medzi koncovými uzlami. Štandard ISO 8073 definuje protokoly transportnej služby typu A (spoľahlivá sieťová služba), B (spoľahlivá sieťová služba s vyznačením chýb), C (nespoľahlivá sieťová služba). Vrstva implementuje spojované a nespojované transportné protokoly, ktoré sú podľa typu prenosu a spôsobu fragmentácie dát rozdelené do piatich protokolových tried - Transport Protocol Class (TP): TP0 až TP4. Tieto triedy boli implementované napríklad ako súčasť softvérového balíku ISODE.

5. **Relačná vrstva (session layer)**: Slúži k udržovaniu relácií medzi komunikujúcimi aplikáciami. Služby zahŕňajú vytvorenie relácie, zrušenie relácie, obsluhu dialógov (v prípade poloduplexných spojení), synchronizáciu a podobne. Štandard ISO 8326 definuje služby relačnej vrstvy a štandard ISO 8327 protokol relačnej vrstvy.

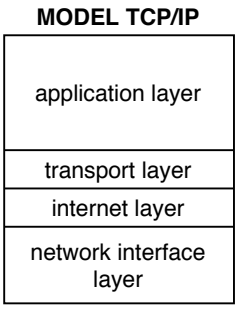
6. **Prezentačná vrstva (presentation layer)**: Zaisťuje zobrazenie (prezentáciu) dát medzi rôznymi aplikáciami a architektúrami. Zahŕňa odlišné formáty dát (ASCII, EBCDIC, binárne dáta), kompresiu dát a kódovanie. Štandard ISO 8822 definuje

služby prezentačnej vrstvy, štandard ISO 8823 popisuje protokoly. Štandard ISO 8824 definuje abstraktnú syntaktickú notáciu ASN.1 (Abstract Syntax Notation one), ktorá sa používa napr. pre prezentáciu dát v sieťových hardvérových zariadeniach.

7. **Aplikačná vrstva (application layer)**: Definuje užívateľské procesy a aplikácie komunikujúce po sieti. Patria sem napr. služby pre všeobecné aplikácie CASE (Common Application Service Elements), ktoré sú popísané štandardmi ISO 8649 (služby CASE) a ISO 8650 (protokoly CASE) Medzi tieto aplikácie patria:
   - elektronická pošta MHS (Message Handling System) - štandard ITU-T X.400
   - adresárové služby - štandard ISO 9594 a ITU-T X.500
   - virtuálny terminál (VT) - ISO 9040 (služby) a ISO 9041 (protokoly)
   - prenos súborov, prístup a spracovanie FTAM (File Transfer Access and Management)

2.1.2 Model TCP/IP

Architektúra TCP/IP je výrazne jednoduchšia ako pri referenčnom modeli OSI 2.1. Model TCP/IP spojuje služby prezentačnej a relačnej vrstvy do jednej vrstvy **aplikačnej**. Na fyzickej úrovni prenosu bitov spojuje fyzickú a linkovú vrstvu do jednej vrstvy **fyzického rozhrania**, ktorá je obvykle implementovaná na sieťovej karte. Na obrázku 2.3 je možné vidieť ukážku tohto modelu.



Obr. 2.3: Grafická ukážka štvorvrstvového modelu TCP/IP

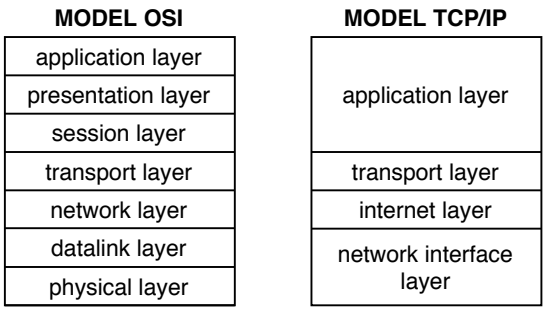
Model TCP/IP odstraňuje nedostatky, ktoré mala komplikovaná štruktúra vrstiev modelu OSI a ktoré neboli nikdy plne implementované. Zrovnanie týchto dvoch modelov je možné vidieť na obrázku 2.4

Implementácia architektúry TCP/IP je rozdelená na tri časti. Najnižšia časť, vrstva fyzického rozhrania, je implementovaná v sieťovej karte (NIC, Network Interface Card) a v ovládači karty (driver). Vyššie vrstvy - internetová a transportná - sú súčasťou sieťových modulov operačných systémov.

Vrstvy modelu TCP/IP

V tejto sekcii bude uvedený stručný popis jednotlivých vrstiev modelu TCP/IP.

1. **Vrstva fyzického rozhrania (network interface layer)**: Popisuje štandardy pre fyzické médium a elektrické signály, napr. Ethernet, Token Ring, FDDI, X.25 a Frame Relay. Okrem fyzických technológii (metalická kabeláž, bezdrôtový prenos, optika,



Obr. 2.4: Porovnanie modelov OSI a TCP/IP

rádiové vlny) zahŕňa tiež funkcie pre prístup k fyzickému médiu (ovládače sieťových kariet). Táto vrstva má taktiež na starosti zapúzdriť (zabaliť) IP datagramy do rámcov.

2. **Internetová vrstva (IP layer)**: Vytvára diagramy, adresuje ich a smeruje na miesto určenia. Zaisťuje doručenie s najväčším úsilím (best-effort delivery), čo znamená, že vrstva IP sa snaží doručiť dáta najvhodnejšou cestou. Pokiaľ dôjde k strate dát, vysielací uzol je o strate datagramu informovaný, avšak musí sám zaistiť opakované prenesenie dát. Internetová vrstva používa protokoly ARP (Address Resolution Protocol) a RARP (Reverse ARP), ktoré slúžia k mapovaniu IP adries na MAC adresy. Ďalej využíva protokoly ICMP (Internet Control Message Protocol) a IGMP (Internet Group Management Protocol) pre prihlasovanie sa do multicastových skupín.

3. **Transportná vrstva (transport layer)**: Prenáša dáta z aplikácie na zdrojovom počítači do aplikácie na cieľovom počítači. Vytvára tzv. **logické spojenia** medzi procesmi. Transportné protokoly rozdeľujú aplikačné dáta na menšie jednotky (pakety), ktoré posielajú po sieti. Na spoľahlivý prenos slúži protokol TCP (Transmission Control Protocol) a na nespoľahlivý prenos slúži protokol UDP (User Datagram Protocol)

4. **Aplikačná vrstva (application layer)**: Je tvorená procesmi a aplikáciami, ktoré komunikujú po sieti. Vrstva zaisťuje spracovanie dát na najvyššej úrovni vrátane reprezentácie dát, kódovania a riadenia dialógu.
   Aplikačné protokoly môžu byť rozdelené do dvoch hlavných kategórií:

   (a) **užívateľské protokoly**, sú protokoly, ktorých služby vykonávajú priamo užívatelia napr. Telnet, FTP, SMTP a podobne...

   (b) **systémové protokoly**, sú protokoly, ktoré zaisťujú sieťové funkcie napr. SNMP, BOOTP a **DNS**

Práve zmienený protokol **DNS** bude predmetom nasledujúcich sekcií 2.2 a 2.3.

2.2 Úvod do systému DNS

Doposiaľ boli predstavené dva spôsoby identifikácie zariadení: pomocou doménového mena a IP adresy. Ľudské bytosti preferujú mnemotechnické doménové mená, kým smerovače a iné sieťové zariadenia preferujú hierarchicky štruktúrované IP adresy. Na vyriešenie tohto problému bolo potrebné spraviť kompromis, a to v podobe prekladu doménových mien

na IP adresy. Služba, ktorá má tento preklad na starosti, sa nazýva *domain name system* (DNS) [7].

2.3 Protokol DNS

Adresovanie a preklad adries je nevyhnutná súčasť internetu, bez ktorej sa žiadna komunikácia medzi užívateľmi nezaobíde. To, že nefunguje preklad doménových mien, sa dá zistiť veľmi rýchlo, keď užívateľ nie je schopný načítať webovú stránku alebo poslať email. Základnou úlohou služby DNS je mapovanie (prevod) doménových adries na IP adresy. Doménová adresa sa často nazýva doménové meno (domain name). Služba DNS obsahuje databázu všetkých doménových adries a príslušných IP adries. Pretože sa jedná o veľmi rozsiahlu databázu, je distribuovaná na viacerých počítačoch, kde bežia špeciálne servery, ktoré sa nazývajú *nameservery*, doménové servery (menné servery) alebo taktiež servery DNS. Proces vyhľadávania v systéme DNS sa nazýva *rezolúcia*.

Systém DNS využíva aplikačné protokoly, napr. HTTP, SMTP, FTP pre preklad doménových adries na IP adresy. To je vždy prvá aktivita, ktorú aplikácia vykoná po tom, čo je požiadaná o pripojenie na vzdialenú službu a je zadaná adresa v podobe doménového mena. Služba najskôr požiada o preklad na IP adresu a následne môže dôjsť k naviazaniu spojenia [9].

Okrem prekladu doménových mien na IP adresu, poskytuje systém DNS aj nasledujúce služby [9]:

- preklad domnénových adries na IP adresy (s využitím záznamov typu A, AAAA),
- preklad IP adries na doménové adresy (s využitím záznamu typu PTR),
- preklad aliasov počítačov na tzv. kanonické mená (s využitím záznamu CNAME),
- určenie poštového servera pre danú doménu (s využitím záznamu MX),
- podpora rozloženia záťaže medzi viacero aplikačných serverov (rotácia záznamov),
- zdieľanie informácií v rámci globálneho priestoru mien alebo delegovanie správy domén na jednotlivé subjekty (pomocou záznamov typu NS).

2.3.1 Priestor doménových mien

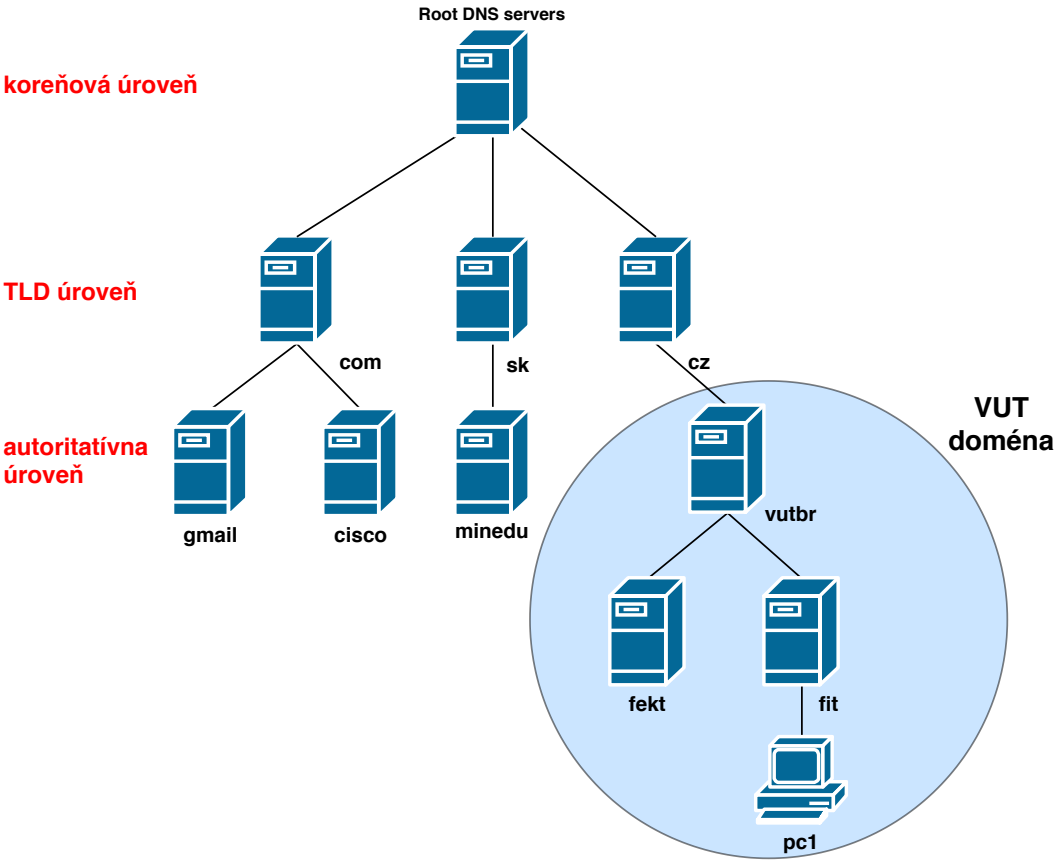
Z dôvodu uloženia a efektívneho vyhľadávania, sa logický priestor všetkých doménových mien ukladá v systéme DNS špeciálnym spôsobom. Systém DNS tvorí databáza usporiadaná hierarchicky ako koreňový strom doménových mien, ako značí obrázok 2.5.

Koreň stromu sa nazýva *root*, uzly stromu sú pomenované textovým reťazcom a niekedy sa nazývajú *domény*. Toto pomenovanie je nepresné, pretože názvy uzlov sú súčasťou domény a teda doména predstavuje podstrom v grafe doménových adries. Príkladom takejto domény môže byť doména VUT, ktorá je znázornená na obrázku 2.5 bledomodrou farbou. *Doménové meno* (domain name) je cesta medzi uzlom, ktorý tvorí vrchol domény, a koreňom stromu DNS. Príklad na takúto doménu môže predstavovať adresa `vutbr.cz` [9].
V zásade rozoznávame 3 druhy DNS serverov:

- koreňový (root nameserver) DNS server = je DNS server, ktorý vracia adresy autoritatívnych serverov zodpovedných pre TLD (top-level domain). Ku každej existujúcej

doméne TLD pozná adresu autoritatívneho serveru. Pri dotaze na akékoľvek doménové meno odpovie buď priamo alebo vráti odkaz na DNS server, ktorý hľadanú informáciu obsahuje. Existuje 13 koreňových serverov [1].

- top-level domain (TLD) DNS server = tento server je zodpovedný za všetky top-level domény[2] napr. `sk`, `cz`, `uk` [7].
- autoritatívny DNS server = je to server, ktorý pozná obsah zónového súboru. Informácie o doménach sú uložené v tzv. zónových súboroch. Jeden zónový súbor by mal obsahovať údaje o jednej doméne, adresách v rámci tejto domény a umožňuje preklad doménového mena servera na jeho IP adresu [12]. Autoritatívny server má oprávnanie odpovedať na dotazy zariadení bez toho, aby musel kontaktovať ďalšie DNS servery [6].



Obr. 2.5: Hierarchické usporiadanie doménových mien v DNS

K pochopeniu, ako tieto servery spolu pracujú, uvádzam krátky príklad. Predstavme si klienta, ktorý chce zistiť IP adresu pre doménové meno `cisco.com`. V prvej aproximácii nastane nasledujúca situácia. Klient najprv kontaktuje jeden z koreňových DNS serverov, ktorý vráti IP adresu TLD serveru pre top-level doménu `com`. Klient následne kontakuje jeden z týchto TLD serverov, ktorý vráti IP adresu autoritatívneho serveru pre `cisco.com`.

[2] Aktuálny zoznam všetkých TLD serverov na Internete: http://data.iana.org/TLD/tlds-alpha-by-domain.txt

Na záver tohto procesu klient kontaktuje autoritatívny server pre `cisco.com`, ktorý mu vráti pôvodne požadovanú IP adresu [7].

2.4 DNS záznamy

Pre ukladanie informácií v dátovom priestore DNS slúžia záznamy DNS tzv. *resource records*. Tieto záznamy sú uložené v textovej podobe v zónových súboroch na serveroch DNS [9]. Názorná ukážka DNS záznamov je demonštrovaná v tabuľke 2.1.

Typ záznamu	**Typ ID**	**Názov**	**Štandard**
SOA	6	Start of Authority	RFC 1034
NS	2	Name Server	RFC 1034
A	1	Address	RFC 1034
MX	15	Mail Exchanger	RFC 1034
CNAME	5	Canonical Name	RFC 1034
PTR	12	Domain Name Pointer	RFC 1034
NAPTR	35	Naming Authority Pointer	RFC 2915, 3403, 3761
TXT	16	Text	RFC 1034
SRV	33	Service Record	RFC 2782
LOC	29	Location	RFC 1876
AAAA	28	IPv6 Address	RFC 3596
DNSKEY	48	DNS Key Record	RFC 4034
RRSIG	46	DNSSEC Signature	RFC 4034
NSEC	47	Next-Secure Record	RFC 4034
NSEC3	50	NSEC record version 3	RFC 5155
DS	43	Delegation Signer	RFC 4034
*	255	All cached records	RFC 1035

Tabuľka 2.1: Záznamy DNS

2.4.1 Popis najbežnejších záznamov DNS

V tejto sekcii budú stručne popísané najbežnejšie záznamy DNS z tabuľky 2.1 [9].

- **Záznam SOA - Start Of Authority**: Záznam SOA obsahuje informácie týkajúce sa uloženia autoritatívnych dát pre danú zónu. Väčšinou je to prvý záznam v zónovom súbore. Každá zóna má práve jeden záznam SOA.
- **Záznam NS - Name Server**: Záznam typu NS určuje autoritatívny server/servery pre danú doménu. Za pomoci týchto záznamov dochádza k budovaniu hierarchickej štruktúry systému DNS. Záznam NS, ktorý je umiestnený v danom uzle stromu DNS, obsahuje ukazovatele na nižšie pripojených nasledovníkov uzlu, ako je možné vidieť na obrázku 2.5.
- **Záznam A - IP Address**: Záznam typu A je definovaný štandardom RFC 1034[10]. Patrí medzi najbežnejšie záznamy DNS, obsahuje priame mapovanie doménových adries na IP adresy. Príklad: vutbr.cz → 147.229.2.90.

- **Záznam MX - Mail Exchanger**: Záznam typu MX informuje o poštovom serveri, ktorý pre danú doménu prijíma poštu. Oproti ostatným záznamom, záznam MX obsahuje naviac prioritu, ktorá určuje preferenciu poštového serveru. Pokiaľ je uvedených viacero poštových serverov pre jednu doménu, použije sa ten s nižšou hodnotou. Nižšia hodnota označuje vyššiu prioritu.

- **Záznam CNAME - Canonical Name**: V DNS môže byť pre jeden počítač vytvorených viacero doménových mien. Pokiaľ napríklad na počítači `tereza.fit.vutbr.cz` pobeží služba WWW, môžeme mu priradiť ľahko zapamätateľný alias `www.fit.vutbr.cz`. Pokiaľ na ňom pobeží aj služba LDAP, je možné pridať ďalší názov `ldap.fit.vutbr.cz`.

- **Záznam PTR - Domain Name Pointer**: Záznam typu PTR je definovaný štandardom RFC 1034[10]. Prevádza spätné (reverzné) mapovanie. To znamená, že prevádza číselnú IP adresu na doménové meno. Príklad:
a.1.2.2.2.5.e.f.f.f.3.2.4.0.2.0.1.0.1.0.1.0.0.0.8.1.7.0.1.0.0.2.ip6.arpa. → www.cesnet.cz.

- **Záznam TXT - Text**: Záznam TXT uchováva v DNS textové dáta týkajúce sa dodatočných informácii o doméne serveru a správcoch. Dnes sa záznamy DNS používajú najčastejšie k overeniu vlastníctva domény.

- **Záznam SRV - Service Record**: Záznam SRV slúži pre lokalizáciu služieb a serverov. Môže sa taktiež použiť pre distribúciu záťaže alebo zálohovanie služieb.

- **Záznam NAPTR - Naming Authority pointer**: Záznam typu NAPTR bol navrhnutý pre mapovanie reťazcov na dáta. Záznam slúži ako podpora pre dynamicky konfigurovatelné systémy DDDS (Dynamic Delegation Discovery Systems). Pokiaľ nie je dosiahnutá podmienka ukončenia, pri vyhľadávaní dát sa nad reťazcami iteratívne používajú transformačné pravidlá.

- **Záznam LOC - Location**: Záznam typu LOC umožňuje administrátorovi zapísať do DNS údaje o umiestnení počítača, siete alebo ďalších zdrojov. Popis lokality zahŕňa informácie o zemepisnej šírke (latitude), dĺžke (longitude) a výške (altitude). Zemepisná šírka a dĺžka sa udáv v stupňoch (minútach a sekundách) a výška sa udáva v metroch.

- **Záznam AAAA - IPv6 Address**: Záznam typu AAAA je definovaný štandardom RFC 3539[15]. Mapuje doménové adresy na IP adresy verzie 6 (IPv6). Príklad: www.cesnet.cz → 2001:718:1:101:204:23FF:FE52:221A[9]

- **Záznam * - All cached records**: Záznam tohto typu patrí medzi pseudozáznamy, je definovaný štandardom RFC 1035[11]. Vracia všetky známe záznamy z doménového servera. Vrátený záznam nemusí byť kompletný.

Kapitola 3

Profilovanie zariadení a metodiky hodnotenia relevancie

Táto kapitola má za úlohu prezentovať techniky a metódy identifikácie mobilných zariadení. Každé zariadenie môže byť popísané nejakými vlastnosťami (atribútmi), ktoré môžu nadobúdať určité hodnoty. V tejto práci budú zariadenia popísané množinou hodnôt, konkrétne termov. Na základe termov bude vytvorený profil zariadenia, ktorý bude slúžiť k jeho identifikácii. Po vzniku týchto profilov zariadení bude potrebné tieto profily jednotlivo porovnať. Vždy sa budú porovnávať len dve zariadenia naraz: zdrojové a cieľové. Zmyslom tohto porovnania bude určenie miery podobnosti profilov medzi týmito zariadeniami. V sekcii 3.1 budú predstavené jednotlivé matematické postupy merania vzdialenosti, ktoré sa bežne používajú na výpočet podobnosti dvoch objektov. V sekcii 3.2 bude popísaný vybraný model, ktorý bude aplikovaný na určenie podobnosti dvoch profilov.

3.1 Určovanie podobnosti číselných dát

V tejto sekcii budú popísané jednotlivé matematické postupy na meranie vzdialenosti, ktoré sa bežne používajú na výpočet rozdielnosti dvoch objektov, ktoré sú popísané atribútmi. Tieto merania sú najčastejšie vykonávané pomocou *Euklidovej*, *Manhattanovej*, *Minkowského* a *Chebyshevovej* vzdialenosti. Metódy merania vzdialenosti dvoch objektov spĺňajú nasledujúce matematické vlastnosti:

- **Nezápornosť:** $d(i, j) \geq 0$: Vzdialenosť dvoch objektov predstavuje vždy kladnú (nezápornú) hodnotu
- **Identita indiscernibles**[1]: $d(i, i) = 0$: Vzdialenosť objektu od samého seba je vždy 0
- **Symetria**: $d(i, j) = d(j, i)$: Vzdialenosť je symetrická funkcia
- **Trojuholníková nerovnosť**: $d(i, j) \leq d(i, k) + d(k, j)$

3.1.1 Euklidovská vzdialenosť - Euclidean distance

Medzi najznámejšie matematické postupy pre výpočet vzdialenosti dvoch objektov patrí Euklidovská vzdialenosť, niekedy označovaná ako geometrická vzdialenosť. Vo všeobecnosti

[1]Identity indiscernibles je ontologický princíp, ktorí hovorí o tom, že nemôžu existovať samostatné objekty alebo subjekty, ktoré majú všetky svoje vlastnosti spoločné. zdroj: https://cs.qwe.wiki/wiki/Identity_of_indiscernibles

je výpočet založený na základe výpočtu dĺžky prepony pravouhlého trojuholníka a v prípade vektorov je výsledná blízkosť vypočítaná ako vzdialenosť koncových bodov vektorov [5]. Nech $i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ a $j = (x_{j1}, x_{j2}, \ldots, x_{jp})$ sú dva objekty popísané p numerickými atribútmi. Potom Euklidovská vzdialenosť medzi objektmi i a j je definovaná ako:

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2} \tag{3.1}$$

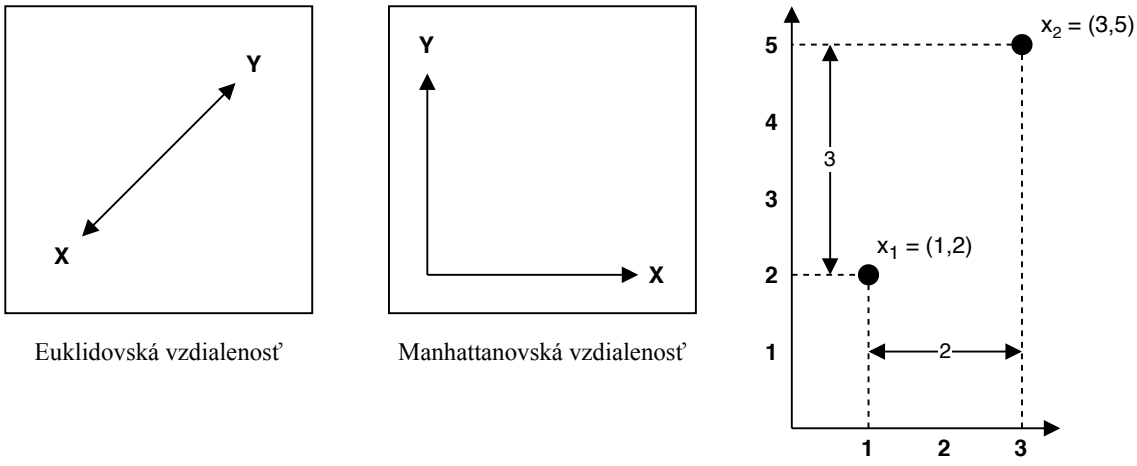
3.1.2 Manhattanovská vzdialenosť - Manhattan distance

Ďalším známym matematickým postupom merania vzdialenosti dvoch objektov (bodov, vektorov) je Manhattanovská vzdialenosť, ktorá meria vzdialenosť dvoch bodov ako súčet vertikálnej a horizontálnej vzdialenosti [5].
Je definovaná ako:

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}| \tag{3.2}$$

Príklad

Majme body $x_1 = (1,2)$ a $x_2 = (3,5)$ reprezentujúce dva objekty. Euklidovskú vzdialenosť týchto dvoch objektov dostaneme ako: $d(x_1, x_2) = \sqrt{(3-1)^2 + (5-2)^2} = \sqrt{2^2 + 3^2} \doteq 3.61$ a Manhattanovskú vzdialenosť dostaneme ako: $d(x_1, x_2) = |3-1| + |5-2| = 2 + 3 = 5$



Obr. 3.1: Porovnanie princípu Manhattanovskej a Euklidovskej vzdialenosti

3.1.3 Minkowského vzdialenosť - Minkowski distance

Minkowského vzdialenosť pozostáva z generalizácie Euklidovskej a Manhattanovskej vzdialenosti:

$$d(i,j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h} \tag{3.3}$$

kde h je reálne číslo, $h \geq 1$. Ak $h = 1$, tak reprezentuje Manhattanovskú vzdialenosť a ak $h = 2$, tak reprezentuje Euklidovskú vzdialenosť [5].

3.1.4 Chebyshevova vzdialenosť - Chebyshev distance

Chebyshevova vzdialenosť predstavuje generalizáciu Minkowského vzdialenosti pre $h \to \infty$. Na výpočet tejto vzdialenosti je potrebné nájsť atribút f, ktorý udáva maximálnu hodnotu rozdielu medzi dvoma objektmi. Tento rozdiel sa v niektorých literatúrach označuje pod pojmom *supremum distance* a je definovaný ako:

$$d(i,j) = \lim_{h\to\infty} \left(\sum_{f=1}^{p} |x_{if} - x_{jf}|^h \right)^{\frac{1}{h}} = \max_{f}^{p} |x_{if} - x_{jf}| \tag{3.4}$$

kde p predstavuje počet atribútov, f predstavuje atribút s maximálnou rozdielnou hodnotou medzi dvoma objektmi. Ak by sa rovnica 3.4 aplikovala na body x_1 = (1,2) a x_2 = (3,5) Chebyshevova vzdialenosť by sa vypočítala ako: $d(x_1, x_2) = |5 - 2| = 3$, pretože druhý atribút udáva najväčšiu vzdialenosť medzi objektmi [5].

3.1.5 Ďalší logický krok - the next logical step

Ďalším krokom v určovaní podobnosti je pridanie váh jednotlivým atribútom podľa dôležitosti, tzv. váhovanie (*weighting*). Príkladom na takúto váhovo ohodnotenú rovnicu môže byť váhovo ohodnotená Euklidovská vzdialenosť, ktorá je definovaná ako:

$$d(i,j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \cdots + w_m(x_{ip} - x_{jp})^2} \tag{3.5}$$

3.2 Vektorový model - Vector Space Model

Medzi najpoužívanejšie modely patrí vektorový model, ktorý predstavuje rozšírený model pre reprezentáciu dokumentov v oblasti získavania informácií. V tomto modeli sú dokumenty a dotazy reprezentované ako body v n-rozmernom priestore. Dimenzie v tomto priestore predstavujú jedinečné slová, tzv. *termy*, ktoré sa nachádzajú v korpuse. Tento dokument je potom možné vyjadriť ako vektor, ktorého zložky majú nenulovú hodnotu, pokiaľ daný dokument obsahuje dané slovo. Tento fakt môže byť chápaný ako najväčší nedostatok tohto modelu, pretože väčšina hodnôt vektora bude vždy nulová a informácia o poradí slov bude stratená. Na druhej strane, hlavnú výhodu vektorového modelu predstavuje jednoduchosť.[4]

3.2.1 Váhovanie termov pomocou TF-IDF

Vektorový model doteraz riešil len samotnú prítomnosť termu v dokumente. Ďalším logickým krokom vo vektorovom modeli je možnosť nastavenia určitej váhy konkrétnym termom. Toto ohodnotenie/váha (*weight*) môže byť nastavené napríklad na základe počtu výskytov slov v dokumente. Táto metóda hodnotenia sa nazýva *Term Frequency* (TF). [14]

Term Frequency - TF

Term Frequency vyjadruje, ako často sa daný *term* vyskytuje v dokumente. Term Frequency (frekvencia opakovania sa slova v dokumente) sa definuje ako: $tf_{t,d}$, kde t predstavuje slovo t v dokumente d. Toto hodnotenie má závažný nedostatok, a to ten, že môže dôjsť k nadhodnoteniu dlhých dokumentov, v ktorých sa hľadaný term môže vyskytovať častejšie

ako v kratších bez toho, aby bol dokument relevantnejší. Jednoduché riešenie predstavuje normalizácia, ktorá spočíva v podelení najčastejšie sa opakujúcim termom v dokumente:

$$ntf_{t,d} = a + (1 - a) * \frac{tf_{t,d}}{tf_{max}(d)} \tag{3.6}$$

v rovnici 3.6 a predstavuje vyhladenie, ktoré je reprezentované číselnou hodnotu medzi 0 a 1. Vo všeobecnosti sa táto hodnota nastavuje na 0.4, avšak niektoré literatúry ju uvádzajú ako 0.5 [14]. Pre účely tejto práce bude definovaná ľahká modifikácia vzorca 3.6 ako:

$$tf_{t,d} = \frac{tf_{t,d}}{tf_{len}(d)} \tag{3.7}$$

kde $tf_{len}(d)$ predstavuje veľkosť dokumentu, táto hodnota slúži ako normalizácia pre výpočet hodnôt TF.

TF - príklad

Uvediem krátky príklad, na ktorom predstavím metódu TF. V nasledujúcich sekciách ho budem rozširovať a demonštrovať jednotlivé metódy. Pre jednoduchosť predpokladajme existenciu štyroch dokumentov o veľkosti jednej vety.

#	**Dokument**
1.	A mother has got a bright smile
2.	A mother has got a car
3.	The mother is kind
4.	The sun shines bright

Tabuľka 3.1: Ukážkové dokumenty

V prvom rade je potrebné si vytvoriť zoznam unikátnych slov z dokumentov z tabuľky 3.1, ako je možné vidieť v tabuľke 3.2.

A	mother	has	got	bright	smile	car	The	is	kind	sun	shines

Tabuľka 3.2: Unikátne slová z dokumentov

Následne sa zostaví tabuľka obsahujúca počet výskytov jednotlivých slov v dokumentoch (Term Frequency), ako je možné vidieť na ukážke 3.3.

Doc	**Vektory dokumentov - TF bez normalizácie**											
	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
	A	mother	has	got	bright	smile	car	The	is	kind	sun	shines
1.	2	1	1	1	1	1	0	0	0	0	0	0
2.	2	1	1	1	0	0	1	0	0	0	0	0
3.	0	1	0	0	0	0	0	1	1	1	0	0
4.	0	0	0	0	1	0	0	1	0	0	1	1

Tabuľka 3.3: Vektory dokumentov bez normalizácie

V poslednom kroku sa aplikuje vzorec 3.6 a vektory sa normalizujú, výsledok tejto normalizácie je možné vidieť v tabuľke 3.4.

Doc	Vektory dokumentov - po aplikovaní normalizácie TF											
	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
	A	mother	has	got	bright	smile	car	The	is	kind	sun	shines
1.	1	0.5	0.5	0.5	0.5	0.5	0	0	0	0	0	0
2.	1	0.5	0.5	0.5	0	0	0.5	0	0	0	0	0
3.	0	1	0	0	0	0	0	1	1	1	0	0
4.	0	0	0	0	1	0	0	1	0	0	1	1

Tabuľka 3.4: Normalizované vektory dokumentov, po aplikovaní vzorca 3.6

Inverse Document Frequency - IDF

Term Frequency má závažný nedostatok pri určovaní relevancie. Pri všetkých termoch je predpoklad, že sú rovnako dôležité. Napríklad slovo *auto* v korpuse o autách môže byť zmienené takmer v každom článku, takže vyhľadávanie iba s použitím váh TF nad toutou kolekciou je v podstate nepoužiteľné, pretože dokumenty sa od seba nijako nerozlišujú. Vhodný spôsob, ako sa s týmto problémom vysporiadať, je predstavenie nového pojmu *Document Frequency*. Document Frequency $DF(t)$ je možné definovať ako počet dokumentov, v ktorých sa term t vyskytuje. Tieto termy je následne možné ohodnotiť váhami, ktoré budú predstavovať ukazateľ toho, ako bežné je dané slovo v korpuse dokumentov. Táto metóda sa nazýva *Inverse Document Frequency* (IDF) a je definovaná ako logaritmus podielu všetkých dokumentov N k počtu dokumentov, ktoré obsahujú dané slovo, teda $DF(t)$ [14].
Na výpočet hodnoty IDF je definovaný nasledovný vzťah:

$$idf_t = \log \frac{N}{df_t} \tag{3.8}$$

Výsledkom aplikácie metódy IDF je stav, kedy slová vyskytujúce sa vo všetkých dokumentoch majú váhu menšiu, zatiaľ čo slová, ktoré boli unikátne a vyskytovali sa v menšom počte dokumentov, majú váhu vyššiu. Tento jav je dôležitý, pretože zriadkavo vyskytujúce sa slová majú vzhľadom k dokumentu najväčšiu výpovednú informáciu. Daný vzťah 3.8 sa zvykne v niektorých prípadoch normalizovať, aby sa predišlo deleniu nulou v prípade, že sa dané slovo nevyskytlo v žiadnom z dokumentov v korpuse [14].
Táto normalizácia sa definuje ako:

$$nidf_t = \log \frac{N}{1 + df_t} \tag{3.9}$$

IDF - príklad

Tento príklad nadväzuje na príklad 3.2.1. K výpočtu hodnôt IDF bol použitý vzorec 3.8, s výpočtom bez normalizácie z dôvodu malého počtu dokumentov a jednotlivých slov, aby nedošlo k skresleniu reálnych hodnôt. Z tabuľky 3.5, ktorá obsahuje výsledné hodnoty IDF, je možné vidieť, že zriedkavo vyskytujúce sa termy ako **sun** a **shines**, majú IDF hodnoty vyššie ako často sa vyskytujúce slová **mother** a **has**, pretože majú vzhľadom k dokumentu väčšiu výpovednú hodnotu.

Doc	Vektory dokumentov - IDF											
	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
	A	mother	has	got	bright	smile	car	The	is	kind	sun	shines
1.	0.301	0.124	0.301	0.301	0.301	0.602	0	0	0	0	0	0
2.	0.301	0.124	0.301	0.301	0	0	0.602	0	0	0	0	0
3.	0	0.124	0	0	0	0	0	0.301	0.602	0.602	0	0
4.	0	0	0	0	0.301	0	0	0.301	0	0	0.602	0.602

Tabuľka 3.5: Vektory dokumentov po aplikovaní IDF 3.8

TF-IDF

Ako už názov metodiky napovedá, jedná sa o kombináciu metód TF a IDF. Váhovanie *TF-IDF* priraďuje váhu termu t v dokumente d na základe súčinu zložiek TF a IDF [14].

$$tf - idf_{t,d} = tf_{t,d} \times idf_t \tag{3.10}$$

Inými slovami, $tf - idf_{t,d}$ priradí váhu termu t v dokumente d tak, že [14]:

1. váha termu je najvyššia, keď sa term t vyskytuje veľakrát v malom počte dokumentov (term je s týmito dokumentmi úzko spätý, a preto ich silno charakterizuje),
2. váha termu je nižšia, keď sa term t vyskytuje málokrát alebo sa vyskytuje vo veľkom počte dokumentov,
3. váha termu bude najnižšia, keď sa term t vyskytuje v podstate každom dokumente.

TF-IDF - príklad

Tento príklad nadväzuje na príklad 3.2.1. K výpočtu hodnôt *TF-IDF* bol použitý vzorec 3.10 a výsledky po aplikovaní tohto vzorca je možné vidieť v tabuľke 3.6.

Doc	Vektory dokumentov - TF-IDF											
	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
	A	mother	has	got	bright	smile	car	The	is	kind	sun	shines
1.	0.301	0.062	0.1505	0.1505	0.1505	0.301	0	0	0	0	0	0
2.	0.301	0.062	0.1505	0.1505	0	0	0.301	0	0	0	0	0
3.	0	0.124	0	0	0	0	0	0.301	0.602	0.602	0	0
4.	0	0	0	0	0.301	0	0	0.301	0	0	0.602	0.602

Tabuľka 3.6: Vektory dokumentov po aplikovaní TF-IDF 3.10

3.2.2 Kosínusová podobnosť - Cosine Similarity

Dokument môže byť reprezentovaný tisíckami atribútov, každý obsahujúci napríklad počet výskytov určitého slova alebo frázy v dokumente. Teda dokument môže byť braný ako objekt, ktorý je reprezentovaný určitým vektorom. Tento vektor môže byť *Term Frequency vektor* alebo *TF-IDF* vektor. Tieto vektory môžu byť veľmi riedke[2] (*sparse*). Tradičné metódy, ktoré boli doposiaľ opísané v tejto kapitole, nie sú vhodné na porovnávanie takýchto riedkych číselných dát. Napríklad, dva term-frequency vektory môžu mať veľa spoločných nulových hodnôt, čo znamená, že korešpondujúce dva dokumenty nezdieľajú veľa spoločných

[2]Vektor obsahuje veľké množstvo nulových hodnôt.

slov, čo ich nečiní podobnými. Práve z tohto dôvodu je potrebné sa zamyslieť nad metódou, ktorá sa zameriava na slová, ktoré sú spoločné pre oba porovnávané dokumenty. Inými slovami, je potrebné aplikovať metódu, ktorá bude ignorovať zhodné nulové hodnoty. Táto metóda sa nazýva *Cosine similarity* (Kosínusová podobnosť). Kosínusová podobnosť predstavuje mieru podobnosti dvoch vektorov, ktorá sa získa výpočtom kosínusu uhla týchto vektorov [5]:

$$similarity = cos(\theta) = \frac{x \cdot y}{\|x\| * \|y\|} = \frac{\sum_{i=1}^{n} x_i \times y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \times \sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{3.11}$$

kde čitateľ (numerator) predstavuje skalárny súčin (dot product) vektorov a menovateľ (denominator) predstavuje Euklidovskú dĺžku tohto vektora. Výsledkom tohto vzťahu je kosínus uhla medzi dvoma vektormi v intervale $\langle 0, 1\rangle$. Hodnota *kosínus 0* znamená, že vektory zvierajú uhol 90°(sú ortogonálne) a nepredstavujú zhodu. Naopak, čím je hodnota kosínusu bližšia k 1, tým väčšiu podobnosť predstavuje [5].

Kosínusová podobnosť - príklad

Tento príklad nadväzuje na príklad 3.2.1. Predpokladajme, že **x** a **y** sú prvé dva TF-IDF vektory z tabuľky 3.6 a chceme zistiť, akú mieru podobnosti tieto dva dokumenty vykazujú.

$\mathbf{x} = (0.301, 0.062, 0.1505, 0.1505, 0.1505, 0.301, 0, 0, 0, 0, 0, 0)$
$\mathbf{y} = (0.301, 0.062, 0.1505, 0.1505, 0, 0, 0.301, 0, 0, 0, 0, 0)$

$\mathbf{x} \cdot \mathbf{y} = 0.301 \times 0.301 + 0.062 \times 0.062 + 0.1505 \times 0.1505 + 0.1505 \times 0.1505 + 0.1505 \times 0 + + 0.301 \times 0 + 0 \times 0.301 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 + 0 \times 0 = \underline{0.1397455}$

$\|x\| = \sqrt{0.301^2 + 0.062^2 + 0.1505^2 + 0.1505^2 + 0.1505^2 + 0.301^2 + 0^2 + \cdots + 0^2} = = \underline{0.3429406418}$

$\|y\| = \sqrt{0.301^2 + 0.062^2 + 0.1505^2 + 0.1505^2 + 0^2 + 0^2 + 0.301^2 + 0^2 + \cdots + 0^2} = = \underline{0.4799442676}$

$similarity(x, y) = \frac{x \cdot y}{\|x\| * \|y\|} = \frac{0.1397455}{0.3429406418 \times 0.4799442676} = \underline{0.8490398348} \approx 0.85$

Na základe výpočtu kosínusovej podobnosti medzi TF-IDF vektormi prvých dvoch dokumentov z tabuľky 3.1 môžme prehlásiť, že dokumenty vykazujú vysokú mieru podobnosti (0.85), čo logicky vyplýva z obsahu jednotlivých viet.

Kapitola 4

Použité technológie

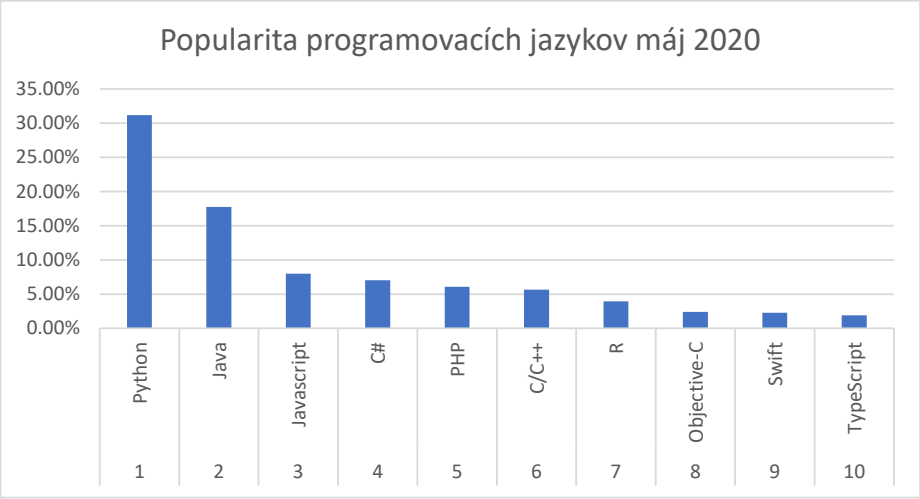
V tejto kapitole budú popísané použité technológie a nástroje, ktoré boli využité k vypracovaniu skriptu, ktorý slúži k identifikácii mobilných zariadení.

4.1 Použité nástroje

Skript bol implementovaný v jazyku `Python 3.8.2`[1], a využíva `MySQL`[2] relačný databázový server na ukladanie dát. Ako vizualizačný nástroj bol použitý `MySQL Workbench`[3].

4.1.1 Python

Python je vysokoúrovňový skriptovací programovací jazyk. Ponúka dynamickú kontrolu dátových typov a podporuje rôzne programovacie paradigmy, vrátane objektovo orientovaného, imperatívneho, procedurálneho alebo funkcionálneho. V roku 2018 vzrástla jeho popularita a v súčasnosti patrí medzi najpopulárnejšie a najobľúbenejšie jazyky, ako je možné vidieť na obrázku grafu 4.1.



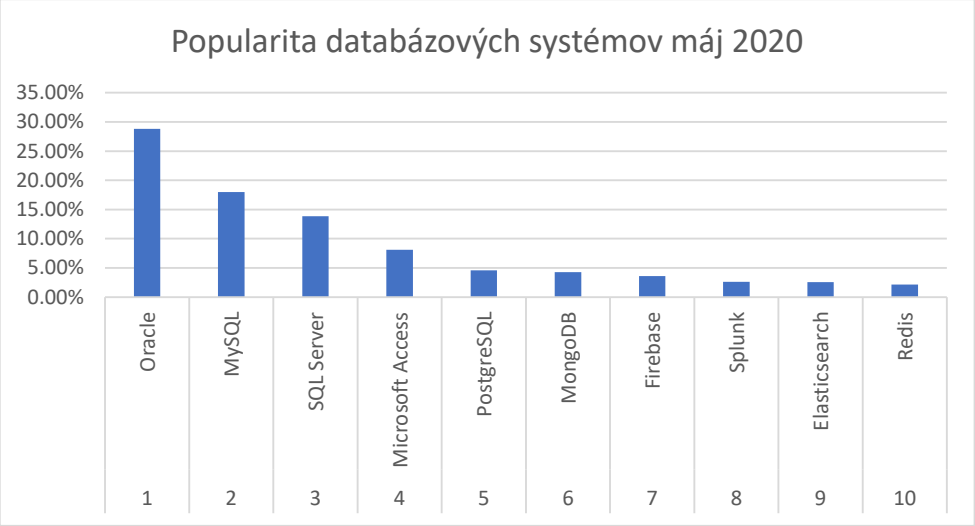
Obr. 4.1: Najpopulárnejšie programovacie jazyky - máj 2020, zdroj: [2]

[1] https://www.python.org/
[2] https://www.mysql.com/
[3] https://www.mysql.com/products/workbench/

Za posledné dekády sa jazyk Python stal hlavným nástrojom pre vedecké výpočetné práce, vrátane analýzy a vizualizácie rozsiahlych dátových sád. Hlavnú výhodu oproti iným programovacím jazykom predstavujú balíčky tretích strán, ako *NumPy* pre manipuláciu s dátami založenými na homogénnych poliach, *Pandas* pre manipuláciu s heterogénnymi dátami, *SciPy* pre rôzne výpočetné úlohy, *Mathplotlib* pre vizulizáciu dát, atď. . . [16]

4.1.2 MySQL

MySQL je otvorený systém riadenia bázy dát, ktorý uplatňuje relačný databázový model. MySQL je podporovaný na viacerých platformách (Linux, Windows, Solaris), je implementovaný v jazykoch `C`, `C++`. Každá databáza je v MySQL tvorená z jednej alebo z viacerých tabuliek, ktoré sa skladajú z riadkov a stĺpcov. V riadkoch sa rozoznávajú jednotlivé záznamy, stĺpce udávajú dátový typ jednotlivých záznamov. Práca s MySQL databázou je vykonávaná pomocou dotazov, ktoré vychádzajú z programovacieho jazyka SQL. MySQL sa nachádza medzi prvými troma najpopulárnejšími databázovými systémami, ako je možné vidieť na obrázku grafu 4.2.



Obr. 4.2: Najpopulárnejšie databázové systémy - máj 2020, zdroj: [2]

4.1.3 MYSQL Workbench

MySQL Workbench je vizualizačný nástroj pre návrh databáz, podporuje vytváranie, administráciu a návrh v jednom integrovanom vývojovom prostredí pre MySQL databázový systém. Predstavuje priameho nástupcu nástroju DBDesigner 4 a prvýkrát bol uvedený do prevádzky v roku 2005.

Kapitola 5

Dátové sady a techniky spracovania dát

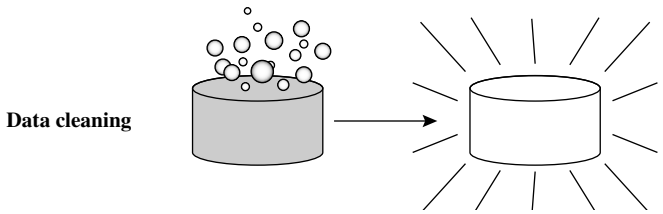
Ako bolo v úvodnej kapitole 1 spomenuté, táto práca je zameraná na identifikáciu mobilných zariadení na základe analýzy DNS dát. Tieto DNS dáta boli poskytnuté Fakultou infočmačných technológií VUT v Brne vo forme dátových sád. V sekcii 5.1 budú popísané hlavné techniky spracovania dát, medzi ktoré patrí čistenie dát 5.1.1, transformácia dát 5.1.2, integrácia dát 5.1.3 a redukcia dát 5.1.4. V sekcii 5.2 budú popísané dátové sady a spôsob ich spracovania.

5.1 Techniky spracovania dát

V dnešnom svete sú databázy zvhľadom na ich rozsiahlu veľkosť vysoko náchylné na zašumené, chýbajúce a nekonzistentné dáta. Pri získavaní znalostí z dát, dáta nízkej kvality potom typicky vedú k slabým výsledkom. Existuje viacero techník na spracovanie dát.

5.1.1 Čistenie dát - data cleaning

Čistenie dát (*data cleaning*) = Táto technika spočíva v odstránení šumu a oprave nekonzistentných dát. Čistenie dát zvyčajne prebieha v iteratívnom dvojkrokovom, postupe, ktorý pozostáva z detekovania nesúladných dát a transformácie dát. Ukážka čistenia dát je znázornená na obrázku 5.1.



Obr. 5.1: Grafická ukážka čistenia dát, obrázok je prevzatý z: [5]

Odstránenie chýbajúcich hodnôt

Za nekonzistentné dáta sa môžu považovať dáta, kde chýba veľký počet hodnôt, čo môže mať za následok negatívny vplyv na konečnú kvalitu výsledku. Existuje mnoho spôsobov, ako sa vysporiadať s chýbajúcimi hodnotami. Medzi tieto spôsoby patrí [5]:

1. **Ignorácia celej n-tice**: Táto metóda nie je veľmi efektívna. Použiteľná je iba v prípade, pokiaľ v danej n-tici chýbajú hodnoty mnohých atribútov. Táto metóda môže viesť k príliš veľkej redukcii dát.

2. **Doplnenie chýbajúcej hodnoty manuálne**: Tento prístup je vo všeobecnej praxi nepoužíteľný, vzhľadom na rozsiahlosť dát a časovú náročnosť.

3. **Vyplnenie chýbajúcej hodnoty globálnou konštantou**: V tomto prístupe sa chýbajúce hodnoty nahradia konštantou. Pokiaľ by sa nová konštanta vyskytovala v dátach príliš často, mohol by ju dolovací algoritmus považovať za významnú, čo by mohlo viesť k negatívnemu ovplyvneniu výsledku.

4. **Použitie priemernej hodnoty atribútov**: V tomto prístupe sa chýbajúce hodnoty nahradia aritmetickým priemerom (mediánom) prítomných hodnôt.

5. **Použitie priemernej hodnoty všetkých vzoriek, ktoré patria do rovnakej triedy ako daná n-tica**: Tento spôsob sa najlepšie demonštruje na praktickom príklade. Majme servis, v ktorom sa opravujú automobily. V servise pracujú rôzni zamestnanci (manažér, technici, účtovníci, atď...). Majme n-ticu `plat` a `pozícia`. Hodnota pozície bude predstavovať technika a hodnota platu bude chýbať. Tento plat sa nahradí priemerom platov všetkých technikov, ale nie priemerom platu všetkých zamestnancov.

6. **Použitie najpravdepodobnejšej hodnoty**: Tento spôsob je zložitejší, nahradenie chýbajúcej hodnoty môže byť prevedené za pomoci metód regresie alebo rozhodovacích stromov.

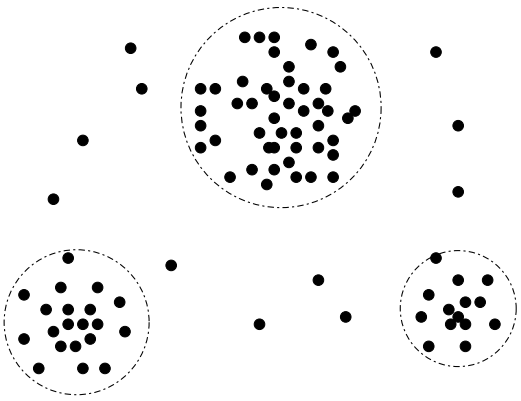
Odstránenie šumu

Šum môže predstavovať náhodnú chyby alebo vysokú odchýlku v hodnote atribútu. Šum môže byť negatívnym faktorom pri ovplyvnení kvality výsledku dolovania. Existuje viacero metód, ako sa s týmto problémom vysporiadať:

- **Odľahlá analýza (outlier analysis)**: Táto metóda sa používa v prípade zhlukovania, keď sa podobné hodnoty organizujú do tzv. zhlukov (clusters) a odľahlé hodnoty spadajú mimo týchto zoskupení. Ukážku zhlukovania je možné vidieť na obrázku 5.2.

- **Plnenie (binning)**: Táto metóda zohľadňuje okolie pri vyhladzovaní zoradených hodnôt. Zoradené hodnoty sú najprv rozdelené do tzv. vedier/košov. V každom vedre je približne rovnaký počet hodnôt. Potom sa hodnota každého prvku nahradí buď priemerom celého vedra alebo hraničnou hodnotou, ku ktorej je najbližšie. Tento proces sa nazýva *lokálne vyhladenie* (local smoothing). Tento postup je znázornený na obrázku 5.3.

- **Regresia (regression)**: Táto metóda vyhladzuje dáta pomocou krivky. V prípade lineárnej regresie sú hodnoty preložené priamkou.



Obr. 5.2: Grafická ukážka zhlukovania, za odľahlé body sú považované tie body, ktoré sa nachádzajú mimo troch hlavných zhlukov. Obrázok je prevzatý z: [5]

Partition into (equal-frequency) bins:
Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:
Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin boundaries:
Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Obr. 5.3: Grafická ukážka plnenia, obrázok je prevzatý z: [5]

5.1.2 Transformácia dát - data transformation

Transformácia dát (*data transformation*) = transformácia dát alebo tzv. normalizácia môže byť aplikovaná tam, kde sú dáta škálované tak, aby zapadli do menšieho intervalu, napr. od 0.0 do 1.0

Data transformation $-2, 32, 100, 59, 48 \longrightarrow -0.02, 0.32, 1.00, 0.59, 0.48$

Obr. 5.4: Grafická ukážka transformácie dát, obrázok je prevzatý z: [5]

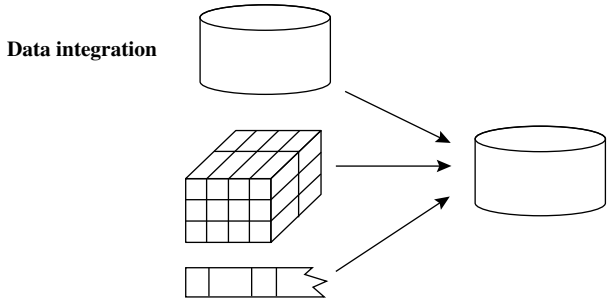
Pri transformácii dát sú dáta transformované na vhodnú formu pre dolovanie. Medzi stratégie transformácie dát patrí [5]:

1. **Vyhladzovanie (smoothing)**: Je to odstránenie šumu z dát. Podrobne je táto technika popísaná v 5.1.1.
2. **Konštrukcia atribútu (attribute construction)**: Nové atribúty sú vytvorené z poskytnutých, napr. adresa môže byť rozdelená na ulicu a číslo domu.

3. **Agregácia (aggregation)**: Agregačné operácie sú aplikované na dáta, ktoré zlučujú viacero hodnôt dohromady, napr. denné tržby môžu byť agregované do mesačných.
4. **Normalizácia (normalization)**: Dáta sú prepočítané tak, aby zapadli do menšieho intervalu, napr: $\langle -1.0, 1.0 \rangle$ alebo $\langle 0.0, 1.0 \rangle$.
5. **Diskretizácia (discretization)**: Dáta numerických atribútov sú nahradené intervalmi (0-10, 11-20).
6. **Generalizácia (generalization)**: Dáta na nízkej úrovni môžu byť nahradené vyšším konceptom, napr. ulica za mesto alebo kraj.

5.1.3 Integrácia dát - data integraton

Integrácia dát (*data integration*) = Táto technika spočíva v zlúčení dát z viacerých zdrojov do súvislého dátového celku. Ukážku názornej integrácie reprezentuje obrázok 5.5.

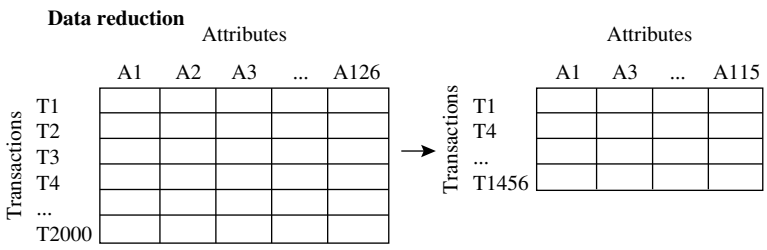


Obr. 5.5: Grafická ukážka integrácie dát, obrázok je prevzatý z: [5]

Integrácia môže pomôcť pri redukovaní redundantných (nadbytočných) a inkonzistentných dát [5].

5.1.4 Redukcia dád - data reduction

Redukcia dát (*data reduction*): Táto technika spočíva v redukovaní veľkosti/rozsiahlosti dát, ideálne bez straty informácií. Možné spôsoby sú: agregácia, odstránenie prebytočných vlastností alebo zhlukovanie. Redukcia dát je bezstratová, ak sa pôvodné dáta dajú zrekonštruovať zo zredukovaných dát bez straty informácií, inak je stratová. Ukážku redukcie dát je možné vidieť na obrázku 5.6.



Obr. 5.6: Grafická ukážka redukcie dát, obrázok je prevzatý z: [5]

Tieto techniky môžu výrazne pomôcť pri presnosti a efektivite dolovacích algoritmov, ktoré zahŕňajú meranie vzdialeností vo vektorovom modeli. [5]

5.2 Dátové sady

Ako bolo uvedené v úvode kapitoly 5, za účelom vyhotovenia tejto práce mi boli poskytnuté DNS dáta v podobe dátových sád. Týchto dátových sád je dokopy päť a obsahujú 6620 DNS záznamov o užívateľskom chovaní. Toto chovanie spočíva v návšteve rôznych doménových mien (internetových stránok a odkazov).
Príklad záznamu
10.42.0.232;10.42.0.1;1;pametove-karty.heureka.cz;

- **10.42.0.232** je IP adresa užívateľa, ktorý sa pýta (dotazuje)
- **10.42.0.1** je IP adresa DNS serveru, na ktorý zasiela dotaz
- **1** je typ DNS záznamu, v tomto prípade sa jedná o A záznam, ktorý slúži na mapovanie doménového mena na IP adresu, tento proces je podrobne popísaný v 2.4
- **pametove-karty.heureka.cz** je URL adresa, na ktorú užívateľ pristupuje

Pri implementácii skriptu a pri bližšom skúmaní obdržaných dátových sád bol zistený výskyt nasledujúcich typov záznamov DNS, ktoré je možné vidieť v tabuľke 5.1. Jednotlivé DNS záznamy sú podrobnejšie popísané v sekcii 2.4.1.

Typ záznamu	Typ ID	Názov	Štandard
A	1	IPv4 Address	RFC 1034
AAAA	28	IPv6 Address	RFC 3539
PTR	12	Domain Name Pointer	RFC 1034
*	255	All cached records	RFC 1035

Tabuľka 5.1: Záznamy DNS, ktoré sa vyskytujú v dátových sadách

5.2.1 Predspracovanie dátových sád

Pred samotnou implementáciou skriptu bolo potrebné si jednotlivé dátové sady vhodným spôsobom predspracovať. Na tento účel bola vytvorená MySQL databáza, ktorá obsahovala sedem tabuliek, ktoré je možné rozdeliť do troch hlavných kategórií:

- **Table of devices (tabuľka zariadení)**: V prvom rade bolo potrebné si vytvoriť tabuľku, ktorá by obsahovala zoznam všetkých unikátnych MAC adries zariadení naprieč všetkými dátovými sadami. K jednotlivým MAC adresám boli priradené IP adresy zariadení, pod ktorými vystupujú dané zariadenia v jednotlivých dátových sadách. Účel tejto tabuľky slúži predovšetkým k overeniu, či sa správne podarilo identifikovať dané zariadenia. Ukážku tejto tabuľky je možné vidieť v tabuľke 5.2. Keďže pri MAC adrese sa jedná o citlivý údaj, nebudú zobrazené v celom tvare, avšak tento fakt nijak jednoznačnosť MAC adresy neovplyvní.
- **Table of queries for dataset(1-5) (Tabuľka dotazov pre dataset)**: Ďalšou tabuľkou (alebo tabuľkami) v poradí je tabuľka dotazov. Táto tabuľka sa skladá z termov[1], z IP adresy zariadenia, ktoré sa dotazovalo, z počtu dotazov na konkrétnu stránku a z hodnoty TF pre daný term. Skrátenú ukážku danej tabuľky je možné vidieť v tabuľke 5.3.

[1] term = typ dns dotazu + dotaz na stránku | 1+a.centrum.cz

- **Table of all queries for all datasets (Tabuľka všetkých dotazov pre všetky dátové sady)**: Táto tabuľka sa skladá z termov, z počtu zariadení, ktoré sa na daný term pýtali v danej dátovej sade a z hodnoty IDF pre daný term. Ukážka tejto tabuľky je prezentovaná v tabuľke 5.4.

# id	MAC_address	IP_DS1	len_DS1	IP_DS2	len_DS2	IP_DS3	len_DS3	IP_DS4	len_DS4	IP_DS5	len_DS5
1	08:xx:xx:xx:xx:fc	10.42.0.199	16		0		0		0		0
2	10:xx:xx:xx:xx:19	10.42.0.232	807		0		0		0		0
3	10:xx:xx:xx:xx:bb	10.42.0.97	89	192.168.42.151	61	10.42.0.151	185	10.42.0.151	82		0
4	10:xx:xx:xx:xx:e5	10.42.0.162	75		0		0		0		0
5	40:xx:xx:xx:xx:73	10.42.0.205	98	192.168.42.76	91		0		0		0
6	70:xx:xx:xx:xx:a3	10.42.0.156	34	192.168.42.82	81	10.42.0.82	328	10.42.0.82	137	10.42.0.82	203
7	8c:xx:xx:xx:xx:2a	10.42.0.253	784		0		0		0		0
8	a4:xx:xx:xx:xx:c0	10.42.0.171	31		0		0		0		0
9	b4:xx:xx:xx:xx:c0	10.42.0.100	191	192.168.42.41	24		0		0		0
10	c0:xx:xx:xx:xx:72	10.42.0.134	54	192.168.42.77	3		0		0		0
11	c8:xx:xx:xx:xx:7c	10.42.0.76	382	192.168.42.16	203		0		0		0
12	d8:xx:xx:xx:xx:f9	10.42.0.161	4		0		0		0		0
13	dc:xx:xx:xx:xx:9f	10.42.0.85	485		0		0		0		0
14	0c:xx:xx:xx:xx:03		0	192.168.42.157	42	10.42.0.157	52		0		0
15	b4:xx:xx:xx:xx:ae		0	192.168.42.1	2	10.42.0.1	6	10.42.0.1	6	10.42.0.1	7
16	00:xx:xx:xx:xx:34		0		0	10.42.0.118	133		0		0
17	a0:xx:xx:xx:xx:52		0		0	10.42.0.52	428		0		0
18	cc:xx:xx:xx:xx:d5		0		0	10.42.0.161	678		0		0
19	48:xx:xx:xx:xx:88		0		0		0	10.42.0.234	15		0
20	ec:xx:xx:xx:xx:d4		0		0		0	10.42.0.193	60		0
21	f8:xx:xx:xx:xx:04		0		0		0	10.42.0.229	22		0
22	00:xx:xx:xx:xx:57		0		0		0		0	10.42.0.230	24
23	00:xx:xx:xx:xx:27		0		0		0		0	10.42.0.16	366
24	14:xx:xx:xx:xx:2f		0		0		0		0	10.42.0.154	27
25	18:xx:xx:xx:xx:13		0		0		0		0	10.42.0.38	132
26	a0:xx:xx:xx:xx:b4		0		0		0		0	10.42.0.160	88
27	a0:xx:xx:xx:xx:f6		0		0		0		0	10.42.0.56	46
28	e4:xx:xx:xx:xx:04		0		0		0		0	10.42.0.84	38

Tabuľka 5.2: Table of Devices (Tabuľka zariadení)

# id	term	ip_address_which_asked	frequency	term_frequency
1	1+DB5SCH101101813.wns.windows.com	10.42.0.100	1	0.005235602
2	1+DB5SCH101110123.wns.windows.com	10.42.0.100	1	0.005235602
168	1+www.gstatic.com	10.42.0.100	2	0.01047120411
184	1+www.google.com	10.42.0.134	9	0.166666666
202	1+android.clients.google.com	10.42.0.156	3	0.088235294
217	1+api-watch-us.huami.com	10.42.0.161	1	0.25
228	1+aktualne.disqus.com	10.42.0.162	26	0.34666666666

Tabuľka 5.3: Table of queries for dataset(1-5) (Tabuľka dotazov pre dátové sady(1-5))

# id	term	n_DS1	IDF_DS1	n_DS2	IDF_DS2	n_D3	IDF_DS3	n_DS4	IDF_DS4	d_DS5	IDF_DS5
1	1+DB5SCH101101813.wns..	1	0.81291	0	0.90308	0	0.84509	0	0.77815	0	0.95424
2	1+DB5SCH101110123.wns...	1	0.81291	0	0.90308	0	0.84509	0	0.77815	0	0.95424
168	1+www.gstatic.com	7	0.21085	1	0.60205	0	0.84509	0	0.77815	5	0.17609
184	1+www.google.com	10	0.07255	5	0.12493	3	0.24303	3	0.17609	8	0
202	1+android.clients.google.com	8	0.15970	2	0.42596	2	0.36797	3	0.17609	6	0.10914
217	1+api-watch-us.huami.com	2	0.63682	0	0.90308	0	0.84509	0	0.77815	0	0.95424
228	1+aktualne.disqus.com	1	0.81291	0	0.90308	0	0.84509	0	0.77815	0	0.95424

Tabuľka 5.4: Table of all queries for all datasets (Tabuľka všetkých dotazov pre všetky dátové sady)

Následná manipulácia s databázou pri implementácii kosínusovej podobnosti sa ďalej riešila už len pomocou SQL dotazov.

Kapitola 6

Návrh a implementácia riešenia

Skript pre identifikáciu mobilných zariadení, o ktorom pojednáva táto práca, bol implementovaný pomocou programovacieho jazyka `Python 3`, ktorý je detailnejšie popísaný v sekcii 4.1. Skript je navrhnutý procedurálne a je možné ho spustiť prostredníctvom linuxového terminálu pomocou pripravených prepínačov.
Príklad spustenia: `python -dsX -dsY [-a] [-h]`
kde `X` je celé číslo zdrojovej dátovej sady v rozmedzí $\langle 1, 5\rangle$ a `Y` je celé číslo cieľovej dátovej sady v rozmedzí $\langle 1, 5\rangle$. Nie je možné porovnávať zariadenia z rovnakej dátovej sady. Parameter `a` je nepovinný, ponúka rozšírený výpis kosínusovej podobnosti medzi jednotlivými dátovými sadami, zoradenými od najpodobnejších po menej podobné dvojice. Parameter `h` zobrazí nápovedu. Na implementáciu boli využité nasledovné knižnice: `sys`, `math`, `argparse` a `mysql.connector`.

6.1 Návrh riešenia

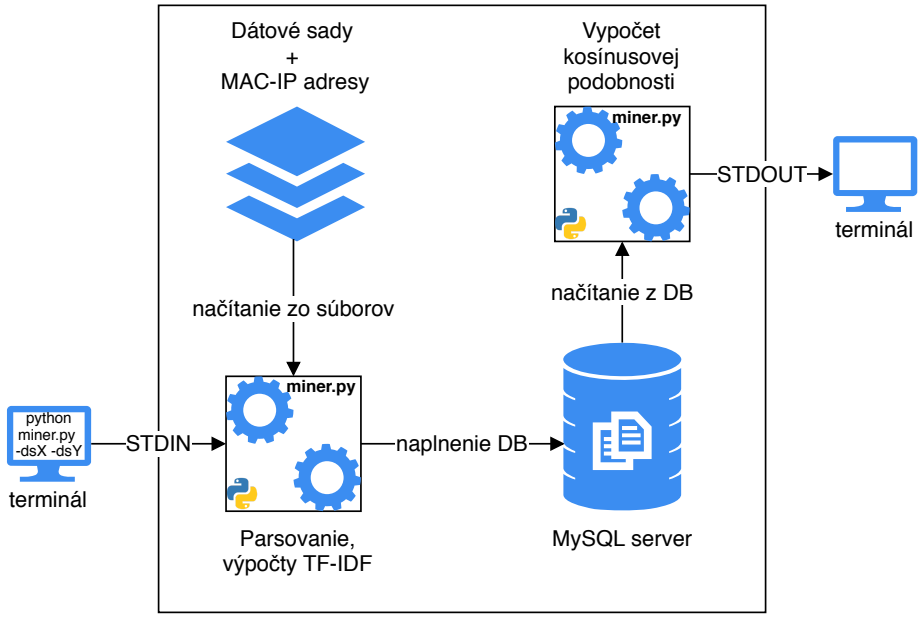
Návrhové riešenie sa skladá zo štyroch hlavných fáz, ktoré sú demonštrované na obrázku 6.1.

6.1.1 Fáza 1 - nadviazanie spojenia a načítanie vstupných súborov

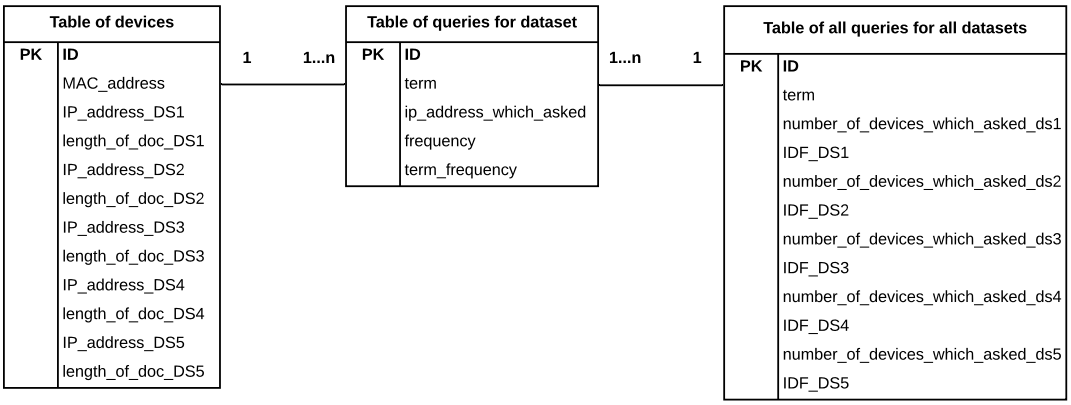
Táto fáza pozostáva z nadviazania spojenia s databázou pomocou importovaného ovládača `mysql.connector`, z vytvorenia inštancie kurzora a z vytvorenia samotnej databázy v prípade, ak neexistuje.

```
mydb = mysql.connector.connect(
    host = "localhost",
    user = "root",
    passwd = "root",
    database = "data_sets"
)
# Creating a new cursor istance
my_cursor = mydb.cursor()
# Creating a new database with a name "data_sets"
my_cursor.execute("CREATE DATABASE data_sets")
```

Vstupným bodom skriptu je funkcia `main()`, ktorá po zavolaní skontroluje syntaktickú správnosť vstupných argumentov a zavolá funkcie `drop_tables()` a `create_tables()`, ktorých účelom je vytvorenie jednotlivých tabuliek pre dátové sady. Návrh týchto tabuliek je demonštrovaný na ERD (Entity Relationship Diagram) 6.2.



Obr. 6.1: Obrázok zobrazuje návrh a logiku skriptu



Obr. 6.2: Obrázok zobrazuje návrh tabuliek v ERD

Vstupné súbory (dátové sady a zoznam MAC adries s prislúchajúcimi IP adresami) sú načítané pomocou funkcií `get_identity_io()` a `get_dataset_content_io()`, ktoré vracajú obsah jednotlivých dátových sád v zozname. Obsah týchto vstupných súborov je potrebné spracovať na vhodný tvar. Tento proces bude popísaný v sekcii 6.1.2.

6.1.2 Fáza 2 - spracovanie vstupných súborov

Táto sekcia pojednáva o spracovaní dátových sád. V prvom rade je potrebné si uvedomiť, že dátové sady sa nachádzajú v jednoduchej textovej forme a je potrebné si ich vhodným spôsobom spracovať. Na toto spracovanie boli vytvorené nasledujúce funkcie:

- `get_table_of_devices_content()` = Funkcia vracia slovník, kde unikátne MAC-adresy (unikátne zariadenie) predstavujú kľúč a prislúchajúce IP adresy, pod kto-

rými vystupujú zariadenia v dátových sadách, `{'MAC-address': ['IP', IP, IP, IP, IP]}` predstavujú hodnotu tohto slovníka. Táto funkcia bude ďalej využitá pri plnení databázy a pri získaní počtu zariadení v jednotlivých dátových sadách. Táto číselná hodnota bude ďalej spracovaná pri výpočte hodnôt IDF

parsed_data_table_1 = get_table_of_devices_content(io_data_mac_and_ip_address)

- `get_table_of_queries_content()` = Funkcia vracia zoznam o piatich položkách `[[ds1], [ds2], [ds3], [ds4], [ds5]]`. Každá položka predstavuje dátovú sadu, ktorá sa nachádza v spracovanom tvare v zozname. V tejto funkcii bolo potrebné aplikovať techniku redukcie dát 5.1.4 a zbaviť sa nadbytočnej položky IP adresy DNS serveru, pretože pre samotný výpočet kosínusovej podobnosti nepredstavuje hlbší význam.
  `10.42.0.100;10.42.0.1;1;a.centrum.cz;` → `10.42.0.100;1;a.centrum.cz;`
  Každá spracovaná dátová sada obsahuje záznamy v tvare `(term, IP_of_device, number_of_requests)`, kde `number_of_requests` predstavuje počet dotazov daného zariadenia na term. Táto hodnota bude ďalej využitá pri výpočte hodnoty TF v sekcii 6.1.3.

  parsed_data_table_2 = get_table_of_queries_content(io_data_all_content_from_all_datasets)

- `get_table_of_all_queries_for_all_datasets()` = Funkcia vracia slovník, kde unikátne termy predstavujú kľúč, a hodnotu slovníka predstavuje pole, ktorého obsahom je 5 položiek. Tieto položky sú celočíselné nezáporné hodnoty, ktoré reprezentujú počet zariadení, ktoré sa v jednotlivých dátových sadách dotazovali na daný term, `{'term' : [int, int, int, int, int]}` Vrátená hodnota tejto funkcie bude ďalej využitá pri výpočte IDF a plnení databázy.

  parsed_data_table_3 = get_table_of_all_queries_for_all_datasets(parsed_data_table_2)

Vrátené hodnoty z funkcií budú využité pri výpočte hodnôt TF a IDF, ktoré budú popísané v nasledujúcej sekcii 6.1.3

6.1.3 Fáza 3 - výpočet hodnôt TF, IDF a plnenie databázy

Táto sekcia pojednáva o výpočte hodnôt TF a IDF, ktoré budú v záverečnej fáze 6.1.4 použité k výpočtu kosínusovej podobnosti. Na výpočty boli vytvorené nasledujúce funkcie:

- `get_number_of_occurrences()` = Účelom tejto funkcie je získať hodnoty dĺžok jednotlivých dokumentov (celkový počet DNS dotazov daného zariadenia) pre jednotlivé zariadenia v danej dátovej sade. Tieto hodnoty je možné vidieť v tabuľke 5.2 ako `len_DSn` (kde n je číslo dátovej sady), ktoré reprezentujú dĺžky jednotlivých dokumentov pre jednotlivé zariadenia. Tieto hodnoty dĺžok dokumentov slúžia ako menovateľ vzorca 3.7. Funkcia vracia pole, v ktorom prvá položka reprezentuje IP adresu zariadenia a druhá položka reprezentuje počet výskytov daného zariadenia v dátovej sade, inými slovami dĺžku dokumentu
  `['IP', int] | ['10.42.0.100', 191]`.

  number_of_occurrences = get_number_of_occurrences(parsed_data_table_2)

- `get_term_frequency()` = Účelom tejto funkcie je vypočítať hodnoty TF pre jednotlivé termy v dátových sadách podľa vzorca 3.7. Čiteteľ je reprezentovaný počtom dotazov daného zariadenia a menovateľ dĺžkou dokumentu. Vypočítané hodnoty TF je možné vidiet v tabuľke 5.3.

  term_frequency = get_term_frequency(parsed_data_table_2, number_of_occurrences)

- `get_nominators()` = Účelom tejto funkcie je získať počet všetkých zariadení, ktoré sa nachádzajú v jednotlivých dátových sadách. Počet všetkých zariadení v konkrétnej dátovej sade reprezentuje čitateľa vzorca 3.9, ktorý sa používa na výpočet hodnôt IDF.

  idf_nominators = get_nominators(parsed_data_table_1)

- `calculate_idf()` = Účelom tejto funkcie je vypočítať hodnoty IDF pre jednotlivé termy podľa vzorca 3.9. Vypočítané hodnoty IDF je možné vidieť v tabuľke 5.4. Z hľadiska implementácie bolo dôležité exiplicitne nastaviť hodnotu logaritmu na základ 10, aby nedošlo k skresleniu výsledkov.

  idf = calculate_idf(parsed_data_table_3, idf_nominators)

Plnenie databázy

Účelom MySQL databázy je priniesť jednoduchšiu manipuláciu s dátami (termy, hodnoty TF, hodnoty IDF, atď...). V tejto fáze sa tabuľky v databáze naplnia hodnotami dát. Výsledkom tohto procesu je sedem tabuliek, ktoré je možné využiť pri finálnom výpočte kosínusovej podobnosti pri porovnávaní mobilných zariadení. Skrátenú formu tabuliek je možné vidieť v sekcii 5.2.1. Na plnenie tejto databázy slúžia funkcie:

- `fill_table_of_devices()`
- `fill_table_of_queries()`
- `fill_table_of_all_queries_for_all_datasets()`

Na plnenie tabuliek bol použitý ovládač mysql.connector. Pri zaslaní dotazu do databázy je vhodné použiť tzv. *placeholder* (zástupný znak) `%s`, aby sa predišlo útoku typu *SQL Injection*[1]. Ukážka kódu plnenia tabuľky 5.4:

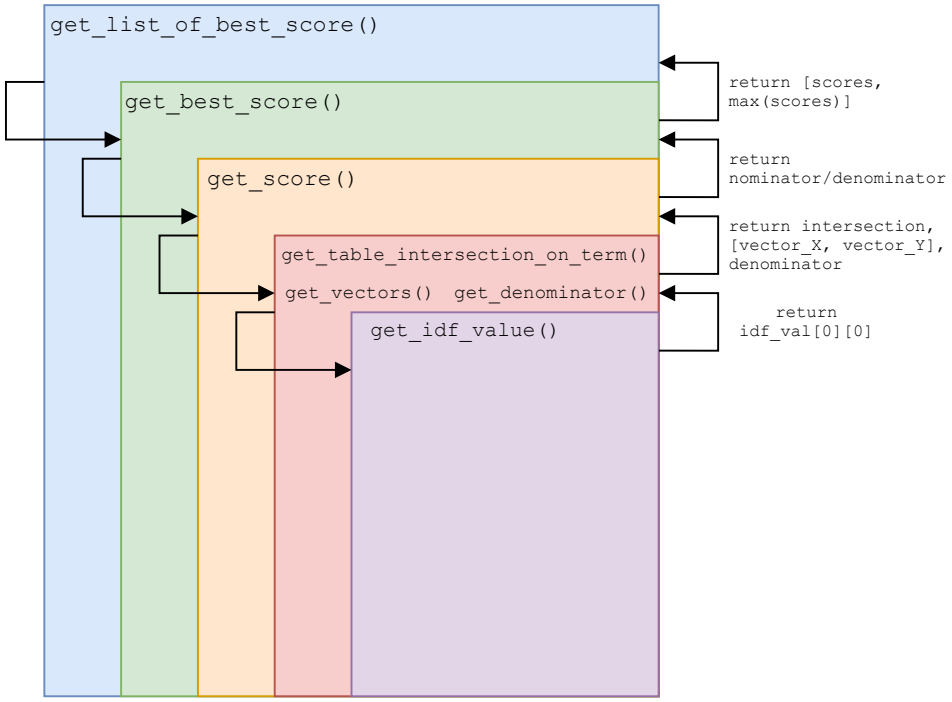
```
# Function fills table "table_of_all_queries_for_all_datasets" with data
def fill_table_of_all_queries_for_all_datasets(parsed_data_table_3, idf):
    index = 0
    for key, value in parsed_data_table_3.items():
        sql = "INSERT INTO table_of_all_queries_for_all_datasets (term,
            number_of_devices_which_asked_ds1, IDF_DS1, number_of_devices_which_asked_ds2
            ,\
            IDF_DS2, number_of_devices_which_asked_ds3, IDF_DS3,
                number_of_devices_which_asked_ds4, IDF_DS4,
                number_of_devices_which_asked_ds5, IDF_DS5)\
            VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)"
        val = (key, value[0], idf[0][index], value[1], idf[1][index], value[2], idf[2][index],\
            value[3], idf[3][index], value[4], idf[4][index])
```

[1]SQL injection je bezpečnostná chyba založená na možnosti manipulácie s dátami v databáze bez nutnosti vlastníctva legitímnych prístupových údajov. https://portswigger.net/web-security/sql-injection

```
my_cursor.execute(sql, val)
mydb.commit()
index = index + 1
```

6.1.4 Fáza 4 - výpočet kosínusovej podobnosti a určenie podobnosti mobilných zariadení

V tejto sekcii sa pojednáva o implementácii kosínusovej podobnosti a určení podobnosti mobilných zariadení.



Obr. 6.3: Obrázok zobrazuje logickú postupnosť volania funkcií k výpočtu kosínusovej podobnosti

- `get_list_of_best_scores()` = Účelom tejto funkcie je získať hodnoty kosínusovej podobnosti medzi zdrojovými a cieľovými zariadeniami. Parametre funkcie `dsX` a `dsY` reprezentujú zdrojovú a cieľovú dátovú sadu, z ktorých sú vybrané zariadenia na porovnanie. V tejto funkcii je nutné iterovane volať funkciu `get_best_scores()`, pre každé zariadenie zo zdrojovej dátovej sady. Tejto funkcii sa potom predávajú parametre: ip adresa zdrojového zariadenia, zdrojová dátová sada a cieľová dátová sada. Funkcia vracia zoznam, ktorého položky predstavujú výsledok celkového porovnania medzi všetkými dvojicami zariadení (parameter **-a** pri spustení skriptu), a výsledok porovnania medzi vybranými top skóre medzi zariadeniami (spustenie skriptu bez parametru **-a**).
  Ukážka kódu:

```
def get_list_of_best_scores(dsX, dsY):
    sql = f"SELECT ip_address_which_asked FROM table_of_queries_for_ds{dsX} GROUP
        BY ip_address_which_asked"
    my_cursor.execute(sql)
```

```
    ip_addresses_X = my_cursor.fetchall()
    ip_addresses_X = [item[0] for item in ip_addresses_X]

    best_scores = {}
    all_scores = {}
    for ip_address_X in ip_addresses_X:
        val = get_best_score(ip_address_X, dsX, dsY)
        all_scores[ip_address_X] = val[0]
        best_scores[ip_address_X] = val[1]

    return [all_scores, best_scores]
```

- `get_best_score()` = V tejto funkcii sú získané všetky IP adresy cieľových zariadení pomocou SQL dotazu. Funkcia volá iterovane pre každú cieľovú IP adresu funkciu `get_score()` pre výpočet podobnosti medzi dvojicami zdrojových a cieľových adries. Okrem IP adries cieľových zariadení sú v tejto funkcii parametre: IP adresa zdrojového zariadenia, zdrojová dátová sada a cieľová dátová sada. Funkcia vracia zoznam, ktorého položky predstavujú výsledky porovnania zariadení z dátových sád rovnako ako v prípade nadradenej funkcie `get_list_of_best_scores()`, kde vrátená hodnota `scores` predstavuje výsledky všetkých porovnaní a hodnota `max(scores)` len výsledky najlepších porovnaní.
  Ukážka kódu:

```
def get_best_score(ip_address_X, dsX, dsY):
    sql = f"SELECT ip_address_which_asked FROM table_of_queries_for_ds{dsY} GROUP
        BY ip_address_which_asked"
    my_cursor.execute(sql)
    ip_addresses_Y = my_cursor.fetchall()
    ip_addresses_Y = [item[0] for item in ip_addresses_Y]

    scores = []
    for ip_address_Y in ip_addresses_Y:
        scores.append([get_score(ip_address_X, ip_address_Y, dsX, dsY), ip_address_Y])

    return [scores, max(scores)]
```

- `get_score()` = Účelom tejto funkcie je získať výsledok podielu hodnôt medzi čitateľom a menovateľom rovnice 3.11. Funkcia volá zanorené pomocné funkcie na výpočet hodnôt čitateľa a menovateľa. Funkcia vracia podiel týchto dvoch hodnôt.
  Ukážka kódu:

```
def get_score(ip_address_X, ip_address_Y, dsX, dsY):
    intersection = get_table_intersection_on_term(f"table_of_queries_for_ds{dsX}", f"
        table_of_queries_for_ds{dsY}", "ip_address_which_asked", ip_address_X,
        ip_address_Y)
    if len(intersection) == 0:
        return 0
    vectors = [[],[]]
    vectors = get_vectors(intersection, dsX, dsY)
    nominator = 0
    for item_X, item_Y in zip(vectors[0], vectors[1]):
        tmp_val = item_X * item_Y
        nominator = nominator + tmp_val

    denominator = get_denominator(ip_address_X, ip_address_Y, dsX, dsY)

    return nominator/denominator
```

  - `get_vectors()` = Účelom tejto funkcie je vypočítať skalárny súčin hodnôt *tf-idf* vektorov podľa čitateľa vzorca 3.11 a vrátiť výsledné vektory nadradenej funkcii.
  - `get_denominator()` = Účelom tejto funkcie je vypočítať Euklidovskú dĺžku vektora podľa menovateľa vzorca 3.11 a vrátiť výslednú hodnotu nadradenej funkcii.
  - `get_table_intersection_on_term()` = Účelom tejto funkcie je získať z databázy spoločné termy (ich prienik) pre porovnávané zariadenia z daných dátových sád, hodnoty TF pre daný term a IP adresu daného zariadenia. Tento prienik termov bude využitý pri skalárnom súčine vektorov v čitateli vzťahu 3.11.

- `get_idf_value()` = Účelom tejto funkcie je získať IDF hodnoty z databázy pomocou SQL dotazu pre konkretný term z konkrénej dátovej sady. Funkcia vracia hodnotu IDF nadradeným funkciám `get_vectors()` a `get_denominator()`.

Výstupom tohto procesu výpočtov je miera kosínusovej podobnosti medzi zariadeniami z porovnávaných dátových sád. Bližší popis k implementácii sa nachádza v zdrojovom kóde.

Kapitola 7

Vyhodnotenie experimentov

V tejto kapitole bude popísané vyhodnotenie experimentov. Tieto experimenty pozostávali z porovnávania profilov zariadení z dátových sád. Profily zariadení boli zostavené na základe termov, ktoré sa skladajú z dotazov jednotlivých zariadení a typu DNS dotazu. Rozhodol som sa prezentovať 3 ukážkové experimenty. Väčší počet by bol zdĺhavý a pre riešenie problematiky by nepredstavoval väčší prínos. Výsledky experimentov sú znázornené v tabuľkách 7.4, v tabuľke 7.5 a v tabuľke 7.6. Pri experimentoch existujú štyri rôzne scenáre:

- **Zariadenie sa podarilo úplne identifikovať**: Zariadenie, ktoré sa podarilo úplne identifikovať sa v tabuľke vyznačilo zelenou farbou. Zariadenie sa nachádzalo v oboch dátových sadách.
- **Zariadenie sa identifikovalo ako nesprávne zariadenie**: Zariadenie, ktoré sa identifikovalo ako nesprávne zariadenie sa v tabuľke vyznačilo červenou farbou. Toto zariadenie by sa malo nachádzať v oboch dátových sadách.
- **Zariadenie sa nepodarilo identifikovať**: Zariadenie, ktoré sa nepodarilo identifikovať predstavuje nulové záznamy v prislúchajúcom riadku tabuľky. Tieto zariadenia by sa mali nachádzať v zdrojovej dátovej sade, a skript k nim nedokázal nájsť žiadne podobné zariadenie z cieľovej dátovej sady.
- **Zariadenie sa identifikovalo ako najpravdepodobnejšie zariadenie**: Zariadenie, ktoré sa identifikovalo ako najpravdepodobnejšie zariadenie, sa vyznačilo žltou farbou. Ak zdrojová dátová sada obsahavala zariadenia navyše, ktoré sa nenachádzali v cieľovej dátovej sade, tak bolo týmto zariadeniam priradené najpravdepodobnejšie zariadenie z cieľovej dátovej sady.

7.1 Experiment 1

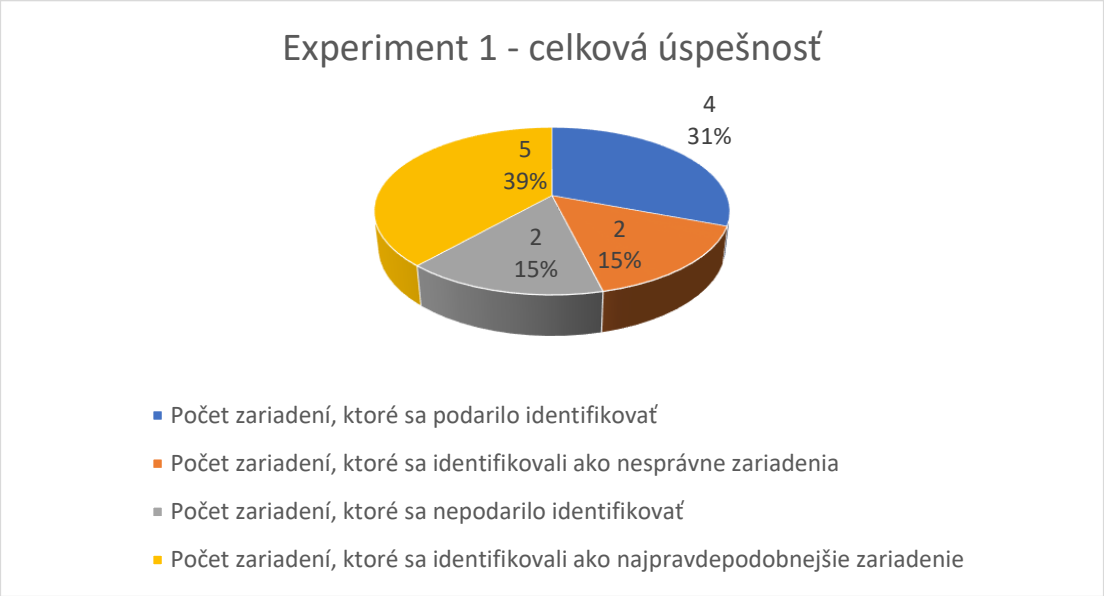
Výsledky tohto experimentu je možné vidieť v tabuľke 7.4. Na porovnanie boli vybrané zariadenia z prvej a druhej dátovej sady. V tomto experimente celkovo participovalo 13 zariadení. Na obrázku grafu 7.1 je možné vidieť celkový podiel identifikovaných zariadení. Z tohto grafu vyplývajú nasledovné skutočnosti:

- Z celkového počtu 13 zariadení boli 4 zariadenia, ktoré sa podarilo jednoznačne identifikovať ako identické zariadenia z dátových sád DS1 a DS2. Jedná sa o nasledujúce zariadenia: **10:xx:xx:xx:xx:bb**, **b4:xx:xx:xx:xx:c0**, **c0:xx:xx:xx:xx:72**

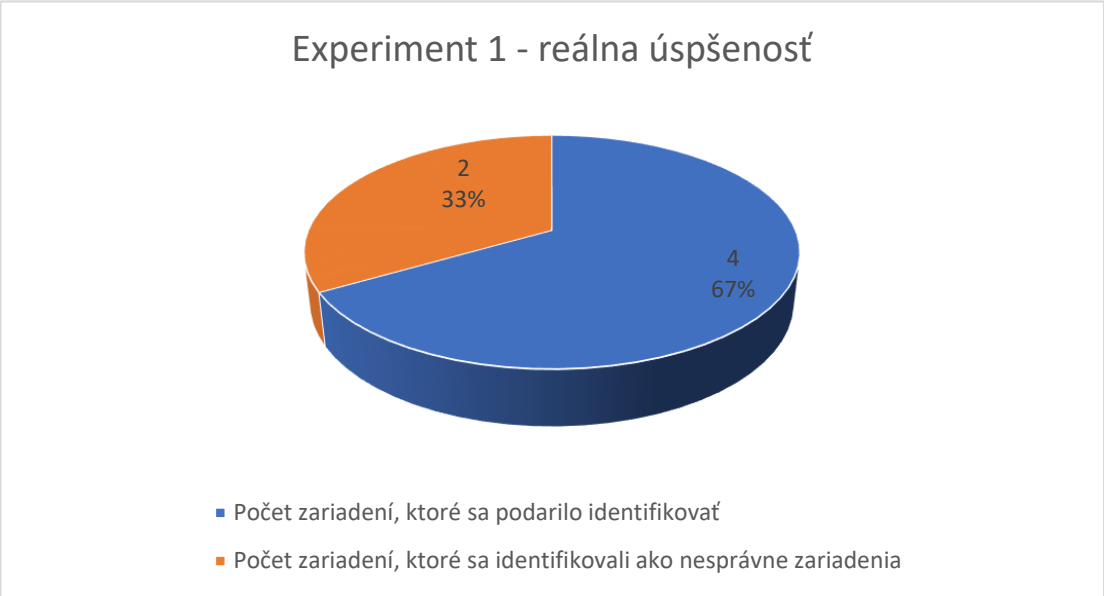
a **c8:xx:xx:xx:xx:7c**. V tabuľke 7.4 sú znázornené zelenou farbou a v grafe 7.1 modrou farbou.

- Zo zvyšných 9 zariadení boli nesprávne identifikované 2 zariadenia. Došlo k zámene zariadení. Jedná sa o nasledujúce zariadenia: **40:xx:xx:xx:xx:73**, ktoré bolo nesprávne identifikované ako zariadenie **10:xx:xx:xx:xx:bb** a zariadenie **70:xx:xx:xx:xx:a3**, ktoré bolo nesprávne identifikované ako zariadenie **10:xx:xx:xx:xx:bb**. Jeden z faktorov, ktorý zapríčinil zlú identifikáciu, mohol byť príliš veľký počet všeobecných dotazov. V tabuľke 7.4 sú znázornené červenou farbou a v grafe 7.1 oranžovou farbou.
- Zo zvyšných 7 zariadení boli 2 zariadenia, ktoré sa nepodarilo identifikovať vôbec. Jedná sa o nasledujúce zariadenia: **08:xx:xx:xx:xx:fc** a **d8:xx:xx:xx:xx:f9**. Tieto zariadenia nezdieľali totiž žiadne spoločné dotazy, na základe ktorých by ich profily mohli byť porovnané. V grafe 7.1 sú vyznačené šedou farbou.
- Posledných 5 zariadení sa identifikovali ako najpravdepodobnejšie zariadenia. Inými slovami: ak dátová sada **X** obsahavala zariadenia navyše, ktoré sa nenachádzali v dátovej sade **Y**, tak bolo týmto zariadeniam priradené najpravdepodobnejšie zariadenie z dátovej sady **Y**. Jedná sa o nasledujúce zariadenia:
  - **10:xx:xx:xx:xx:19** → **b4:xx:xx:xx:xx:c0**
  - **10:xx:xx:xx:xx:e5** → **10:xx:xx:xx:xx:bb**
  - **8c:xx:xx:xx:xx:2a** → **40:xx:xx:xx:xx:73**
  - **a4:xx:xx:xx:xx:c0** → **c0:xx:xx:xx:xx:72**
  - **dc:xx:xx:xx:xx:9f** → **70:xx:xx:xx:xx:a3**

  V tabuľke 7.4 sú znázornené žltou farbou a v grafe 7.1 ravnako žltou farbou.



Obr. 7.1: Obrázok zobrazuje výsledky prvého experimentu z tabuľky 7.4.



Obr. 7.2: Obrázok zobrazuje podieľ počtu správne identifikovaných zariadení a nesprávne identifikovaných zariadení v experimente 3.

Záver experimentu číslo 1

Záver prvého experimentu môže byť pokladaný za úspešný a uspokojivý. Na prvý pohľad sa môže zdať, že na základe grafu 7.1 je počet zariadení z celkového počtu, ktoré sa podarilo identifikovať malý. Štyri správne identifikované zariadenia z trinástich činí celkovú úspešnosť len 31%. Avšak, treba brať ohľad na viacero faktorov,napríklad na zariadenia, ktoré sa nachádzali len v jednej z dátových sád. Preto by bolo vhodnejšie rátať uspešnosť z grafu 7.2, ktorý zobrazuje pomer zariadení, ktoré sa podarilo správne identifikovať a zariadení, ktoré boli nesprávne identifikované a došlo k zámene týchto zariadení. Reálna úspešnosť identifikácie mobilných zariadení na základe DNS dát z dátových sád DS1 a DS2 by mala činiť 67%. Výsledky prvého experimentu sú zhrnuté v tabuľke 7.1.

Celková úspešnosť	Reálna uspšnosť
31%	67%

Tabuľka 7.1: Vyhodnotenie experimentu číslo 1

7.2 Experiment 2

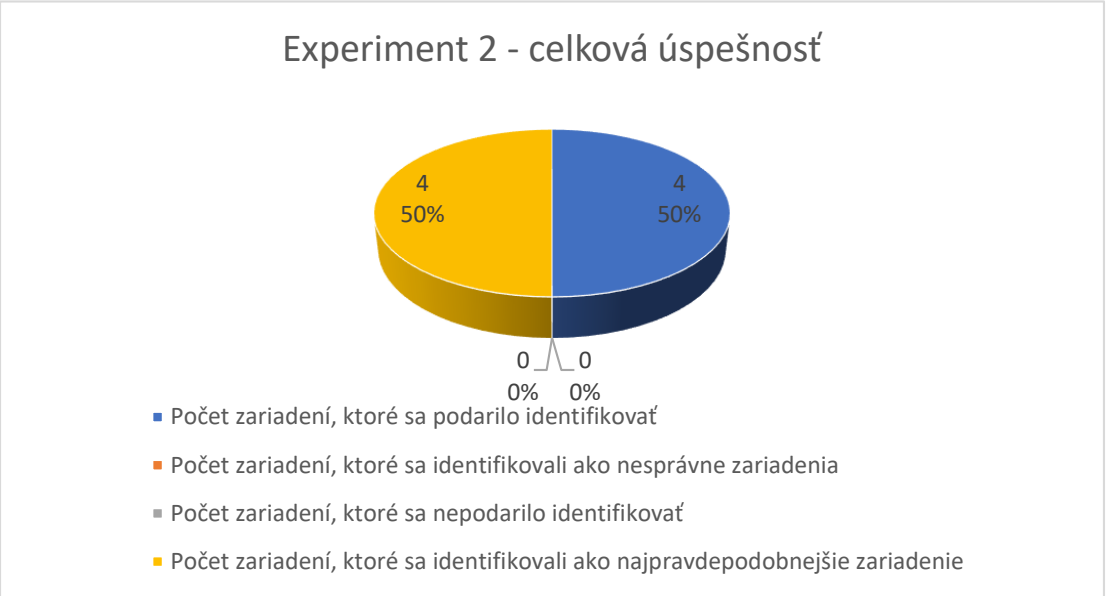
Výsledky tohto experimentu je možné vidieť v tabuľke 7.5. Na porovnanie boli vybrané zariadenia z druhej a tretej dátovej sady. V tomto experimente celkovo participovalo 8 zariadení. Na obrázku grafu 7.3 je možné vidieť celkový podiel identifikovaných zariadení. Z tohto grafu vyplývajú nasledovné skutočnosti:

- Z celkového počtu 8 zariadení boli 4 zariadenia, ktoré sa podarilo jednoznačne identifikovať ako identické zariadenia z dátových sád DS2 a DS3. Jedná sa o nasledujúce zariadenia: **10:xx:xx:xx:xx:bb**, **70:xx:xx:xx:xx:a3**, **0c:xx:xx:xx:xx:03**

a **b4:xx:xx:xx:xx:ae**. V tabuľke 7.5 sú znázornené zelenou farbou a v grafe 7.3 modrou farbou.

- Zo zvyšných 4 zariadení nebolo ani jedno zariadenie identifikované ako nesprávne (nedošlo k zámene zariadení). V grafe 7.3 sú vyznačené oranžovou farbou a nesú nulový podiel.
- Zo zvyšných 4 zariadení nebolo ani jedno zariadenie, ktoré by sa nedalo identifikovať ako v prípade experimentu 7.1.V grafe 7.3 sú vyznačené šedou farbou a nesú nulový podiel.
- Posledné 4 zariadenia sa identifikovali ako najpravdepodobnejšie zariadenia, rovnako ako v experimente 7.1. Jedná sa o nasledujúce zariadenia:
  - **40:xx:xx:xx:xx:73** → **cc:xx:xx:xx:xx:d5**
  - **b4:xx:xx:xx:xx:c0** → **00:xx:xx:xx:xx:34**
  - **c0:xx:xx:xx:xx:72** → **70:xx:xx:xx:xx:a3**
  - **c8:xx:xx:xx:xx:7c** → **70:xx:xx:xx:xx:a3**

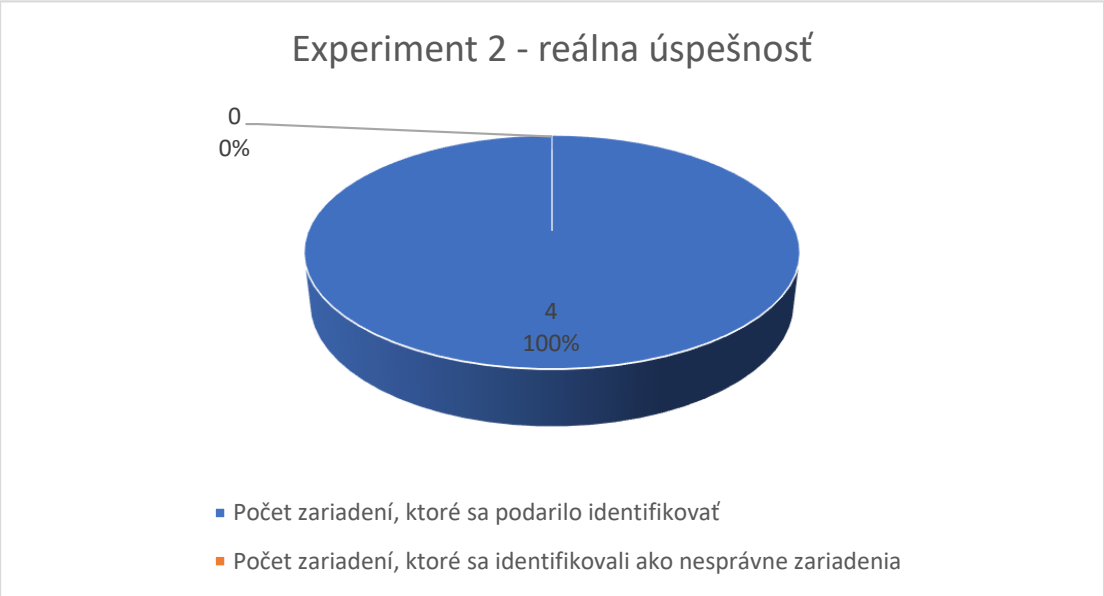
  V tabuľke 7.5 sú znázornené žltou farbou a v grafe 7.3 rovnako žltou farbou.



Obr. 7.3: Obrázok zobrazuje výsledky druhého experimentu z tabuľky 7.5.

Záver experimentu číslo 2

Záver druhého experimentu môže byť pokladaný za úspešný a uspokojivý, rovnako ako v prípade experimentu 7.1. Na základe grafu 7.3 je možné vidieť, že celková úspšnosť identifikovaných zariadení činí 50%, čo zodpovedá štyrom identifikovaným zariadeniam. Na rozdiel od experimentu 7.1, nedošlo k žiadnemu nesprávne identifikovanému zariadeniu (nedošlo k zámene zariadení), ako značí graf 7.4. Reálna úspešnosť identifikácie mobilných zariadení na základe DNS dát z dátových sád DS2 a DS3 by mala činiť 100%. Výsledky druhého experimentu sú zhrnuté v tabuľke 7.2.



Obr. 7.4: Obrázok zobrazuje podieľ počtu správne identifikovaných zariadení a nesprávne identifikovaných zariadení v experimente 3.

Celková úspešnosť	Reálna uspšnosť
50%	100%

Tabuľka 7.2: Vyhodnotenie experimentu číslo 2

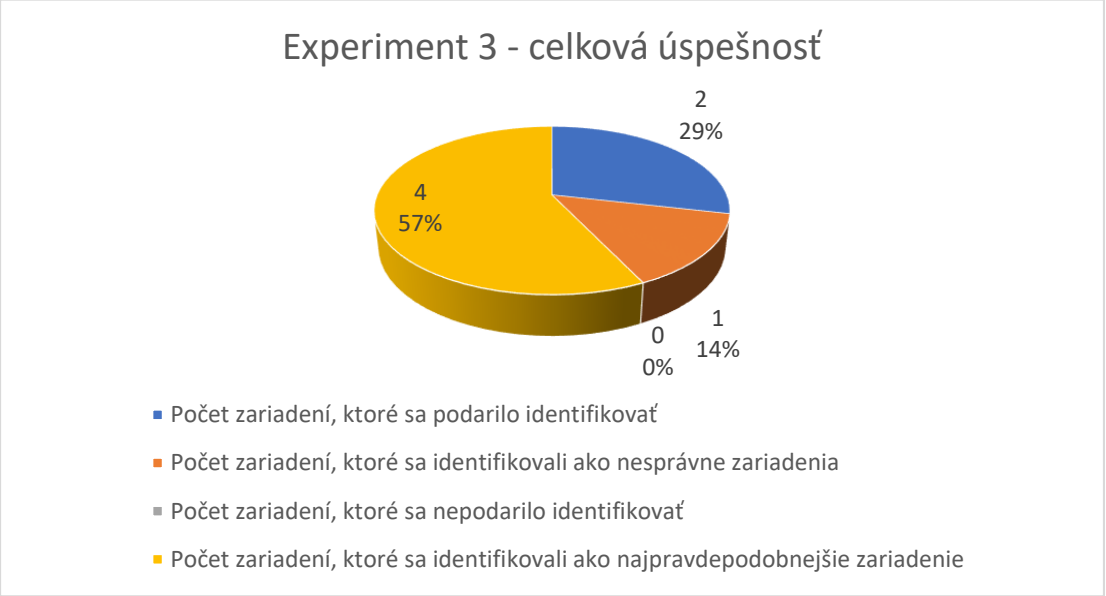
7.3 Experiment 3

Výsledky tohto experimentu je možné vidieť v tabuľke 7.6. Na porovnanie boli vybrané zariadenia z tretej a štvrtej dátovej sady. V tomto experimente celkovo participovalo 7 zariadení. Na obrázku grafu 7.5 je možné vidieť celkový podiel identifikovaných zariadení. Z tohto grafu vyplývajú nasledovné skutočnosti:

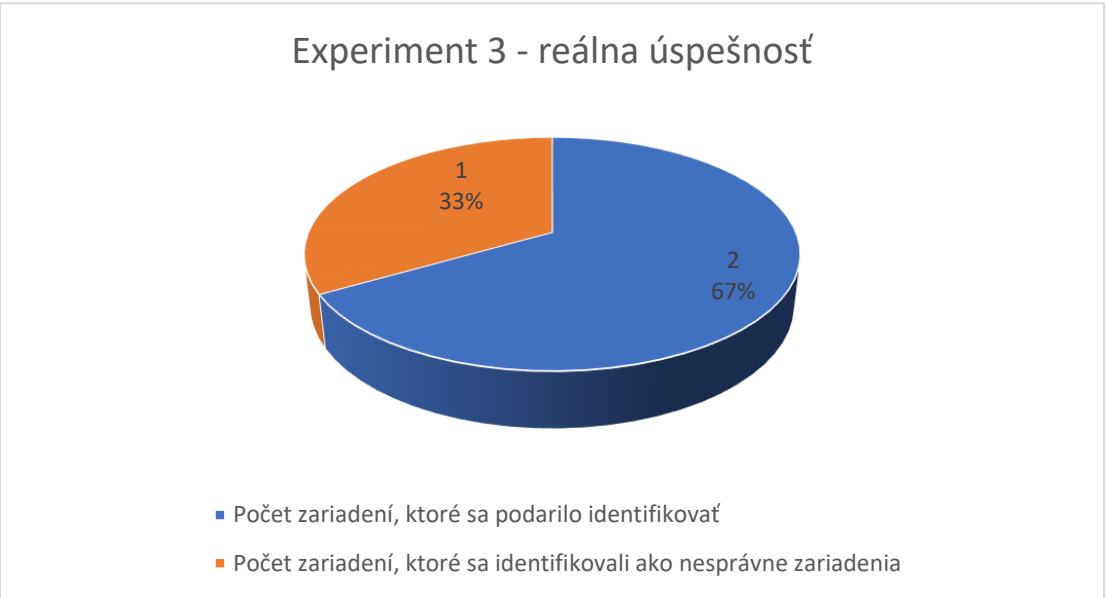
- Z celkového počtu 7 zariadení boli 2 zariadenia, ktoré sa podarilo jednoznačne identifikovať ako identické zariadenia z dátových sád DS3 a DS4. Jedná sa o nasledujúce zariadenia: **10:xx:xx:xx:xx:bb** a **70:xx:xx:xx:xx:a3**. V tabuľke 7.6 sú znázornené zelenou farbou a v grafe 7.5 modrou farbou.
- Zo zvyšných 5 zariadení bolo 1 zariadenie identifikované ako nesprávne, resp. nebolo identifikované vôbec. Dôvodom tohto neúspechu mohol byť minimálny počet (6) dotazov zariadení, ako je možné vidieť v tabuľke 5.2 Naviac, jednalo sa o úplne rozličné dotazy. V tabuľke 7.6 je tento stav znázornený červenou farbou a v grafe 7.5 je znázornený oranžovou farbou.
- Zo zvyšných 4 zariadení nebolo ani jedno zariadenie, ktoré by sa nedalo identifikovať. V grafe 7.3 je znázornené šedou farbou a nesie nulový podiel.
- Posledné 4 zariadenia sa identifikovali ako najpravdepodobnejšie zariadenia. Rovnako ako v experimente 7.1 a 7.2, jedná sa o nasledujúce zariadenia:
  - **0c:xx:xx:xx:xx:03** → **b4:xx:xx:xx:xx:ae**

– **00:xx:xx:xx:xx:34 → 48:xx:xx:xx:xx:88**
– **a0:xx:xx:xx:xx:52 → 70:xx:xx:xx:xx:a3**
– **cc:xx:xx:xx:xx:d5 → 70:xx:xx:xx:xx:a3**

V tabuľke 7.6 sú znázornené žltou farbou a v grafe 7.5 ravnako žltou farbou.



Obr. 7.5: Obrázok zobrazuje výsledky tretieho experimentu z tabuľky 7.6.



Obr. 7.6: Obrázok zobrazuje podieľ počtu správne identifikovaných zariadení a nesprávne identifikovaných zariadení v experimente 3.

Záver experimentu číslo 3

Záver tretieho experimentu môže byť pokladaný za úspešný a uspokojivý, rovnako ako v pre-

došlých prípadoch. Rovnako ako v prípade experimentu 7.1 a 7.2, na základe grafu 7.5 je možné vidieť, že celková úspšnosť identifikovaných zariadení činí 29%, čo zodpovedá dvom identifikovaným zariadeniam. Došlo k nesprávnej identifikácii jedného zariadenia, resp. nebolo identifikované vôbec. Dôvodom tohto stavu mohol byť fakt, že zariadenie disponovalo len malou vzorkou dotazov v dátových sadách, konkrétne šiestimi, ktoré naviac boli úplne rozličné. Rovnako ako v predošlých príkladoch, reálnu úspešnosť by bolo vhodnejšie rátať z grafu 7.6. Reálna úspešnosť identifikácie mobilných zariadení na základe DNS dát z dátových sád DS3 a DS4 by mala činiť 67%. Výsledky tretieho experimentu sú zhrnuté v tabuľke 7.3.

Celková úspešnosť	Reálna uspšnosť
29%	67%

Tabuľka 7.3: Vyhodnotenie experimentu číslo 3

7.4 Celkové vyhodnotenie experimentov

Na experimentoch: 7.1, 7.2 a 7.3 bolo možné pozorovať značné odchýlky pri celkovej a reálnej úspešnosti. Jedným z faktorov, ktorý tento stav zapríčinili, je zvolená metodika TF-IDF, ktorá nepredstavuje úplne ideálny spôsob profilovania mobilných zariadení. Ako najväčší faktor pôsobí samotná aktivita jednotlivých užívateľov, na základe ktorého prebieha zostavenie profilov, ktoré sú porovnávané. K zlepšeniu výsledkov by mohol viesť presnejší profil týchto zariadení, ktorý by obsahoval viac DNS dotazov.

	10:xx:xx:xx:xx:bb	**70:xx:xx:xx:xx:a3**	**40:xx:xx:xx:xx:73**	**b4:xx:xx:xx:xx:c0**	**c0:xx:xx:xx:xx:72**	**c8:xx:xx:xx:xx:7c**	**0c:xx:xx:xx:xx:03**	**b4:xx:xx:xx:xx:ae**
08:xx:xx:xx:xx:fc	0	0	0	0	0	0	0	0
10:xx:xx:xx:xx:19	0.0001244859	0.0000862240	0.0000870387	**0.0005964884**	0	0.0002484269	0	0
10:xx:xx:xx:xx:bb	**0.0061658528**	0.0009675407	0.0025588357	0.0000642257	0	0.0005068458	0	0
10:xx:xx:xx:xx:e5	**0.0067453429**	0.0002842555	0.0008406489	0.0000355470	0	0.0014010872	0	0
40:xx:xx:xx:xx:73	**0.0105696807**	0.0007741730	0.0035389993	0.0000787887	0.0012199722	0.0018006296	0	0
70:xx:xx:xx:xx:a3	**0.0110317708**	0.0016078452	0.0007543295	0.0002660566	0.0067817588	0.0023388460	0	0
8c:xx:xx:xx:xx:2a	0.0002792941	0.0001413114	**0.0455217927**	0.0000700563	0.0001171213	0.0002938747	0	0
a4:xx:xx:xx:xx:c0	0.0093758400	0.0015351503	0.0003299793	0	**0.0139445290**	0.0018263596	0	0
b4:xx:xx:xx:xx:c0	0.0000024489	0.0000599971	0.0041932070	**0.0068341506**	0	0.0010508372	0	0
c0:xx:xx:xx:xx:72	0.0004144299	0.0002475091	0.0011057396	0.0002563201	**0.0584847410**	0.0002914042	0	0
c8:xx:xx:xx:xx:7c	0.0016892383	0.0003433689	0.0091595056	0.0000194408	0.0005370036	**0.0573477617**	0	0
d8:xx:xx:xx:xx:f9	0	0	0	0	0	0	0	0
dc:xx:xx:xx:xx:9f	0.0008835921	**0.0899002219**	0.0001118804	0.0000636753	0.0028056108	0.0001226859	0	0

Tabuľka 7.4: Experiment 1: porovnanie zariadení v DS1 a DS2, prvý stĺpec reprezentuje MAC adresy zdrojových zariadení a prvý riadok MAC adresy cieľových zariadení, číselné hodnoty reprezentujú mieru kosínusovej podobnosti

	10:xx:xx:xx:xx:bb	**70:xx:xx:xx:xx:a3**	**0c:xx:xx:xx:xx:03**	**b4:xx:xx:xx:xx:ae**	**00:xx:xx:xx:xx:34**	**a0:xx:xx:xx:xx:52**	**cc:xx:xx:xx:xx:d5**
10:xx:xx:xx:xx:bb	**0.0076956121**	0.0009394778	0	0	0	0.0003554816	0.0000094382
40:xx:xx:xx:xx:73	0.0022386395	0.0016841755	0	0	0	0.0016071279	**0.0177374059**
70:xx:xx:xx:xx:a3	0.0007781500	**0.0500543371**	0	0	0	0.0000155066	0.0089122421
b4:xx:xx:xx:xx:c0	0.0001183892	0.0002448629	0	0	**0.0024166366**	0.0001551296	0.0000036467
c0:xx:xx:xx:xx:72	0	**0.0059433827**	0	0	0	0	0.0001625686
c8:xx:xx:xx:xx:7c	0.0007900621	**0.0008820204**	0	0	0	0.0003421808	0.0000074970
0c:xx:xx:xx:xx:03	**0.1221996000**	0	0	0	0	0	0
b4:xx:xx:xx:xx:ae	0	0	0	**0.2857984378**	0	0	0

Tabuľka 7.5: Experiment 2: porovnanie zariadení v DS2 a DS3, prvý stĺpec reprezentuje MAC adresy zdrojových zariadení a prvý riadok MAC adresy cieľových zariadení, číselné hodnoty reprezentujú mieru kosínusovej podobnosti

	10:xx:xx:xx:xx:bb	**70:xx:xx:xx:xx:a3**	**b4:xx:xx:xx:xx:ae**	**48:xx:xx:xx:xx:88**	**ec:xx:xx:xx:xx:d4**	**f8:xx:xx:xx:xx:04**
10:xx:xx:xx:xx:bb	**0.0085215470**	0.0029841538	0	0.0003700883	0.0031559853	0.0014092241
70:xx:xx:xx:xx:a3	0.0031071855	**0.0361156613**	0	0	0.0005647915	0.0014612824
0c:xx:xx:xx:xx:03	0	0	**0.0637915638**	0	0	0
b4:xx:xx:xx:xx:ae	0	0	0	0	0	0
00:xx:xx:xx:xx:34	0	0	0	**0.0007396478**	0	0
a0:xx:xx:xx:xx:52	0.0000584977	**0.0003041598**	0	0	0.0001450534	0
cc:xx:xx:xx:xx:d5	0.0000035605	**0.0014686824**	0	0	0.0000082302	0.0000183436

Tabuľka 7.6: Experiment 3: porovnanie zariadení v DS3 a DS4, prvý stĺpec reprezentuje MAC adresy zdrojových zariadení a prvý riadok MAC adresy cieľových zariadení, číselné hodnoty reprezentujú mieru kosínusovej podobnosti

Kapitola 8

Záver

Cieľom tejto bakalárskej práce bolo identifikovať mobilné zariadenia na základe analýzy DNS dát, ktoré boli zachytené pri sieťovej komunikácii týchto zariadení. Na splnenie tohto zámeru bolo potrebné implementovať skript, ktorý by umožnil spracovať obrovské množstvo dát. Tento zámer bol úspešne splnený, ako je možné vidieť v kapitole 6 a 7. Z možných profilovacích techník bola vybraná práve technika váhovania vektorov TF-IDF v kombinácii s kosínusovou podobnosťou. Aj keď sa nejedná o úplne ideálnu metódu, ako bolo naznačené v sekcii 7.4, z opísaných spôsobov váhovania v sekcii 3.2 bolo dokázané, že z predstavených metód je metóda TF-IDF jednoznačne v kombinácii s kosínusovou podobnosťou najpresnejšia. Pomocou kombinácie týchto metód bola úspešne vypočítaná podobnosť ukážkových dokumentov z tabuľky 3.1, ako je možné vidieť v sekcii 3.2.2.

Z osobného hľadiska mi práca umožnila aplikovať a rozšíriť nadobudnuté vedomosti z oblasti sieťových technológií a databázových systémov v praktickej forme. Vystavila ma doposiaľ pre mňa neprebádanému odvetviu informatiky, dolovaniu dát a získavaniu znalostí z databáz. Rovnako mi aj prehĺbila programátorské zručnosti v jazykoch Python a SQL.

Vhodným rozšírením tejto práce by bola možnosť vysporiadať sa s nedostatkami metód TF-IDF, ktoré boli opísané v celkovom vyhodnotení experimentov 7.4. Jedným z možných spôsobov by bolo zostaviť presnejší profil zariadení. Na zostavenie tohto profilu by mohli byť využité dotazy jednotlivých zariadení z rozličných dátových sád, a neobmedzovalo by sa len na jednu konkrétnu.

Literatúra

[1] Bush, R., Karrenberg, D., Kosters, M. a Plzak, R. *Root Name Server Operational Requirements* [online]. 2000. Jún 2000 [cit. 15. mája 2020]. Dostupné z: https://tools.ietf.org/html/rfc2870.

[2] Carbonnelle, P. *PopularitY of Programming Language Index* [online]. 2020. Máj 2020 [cit. 21. mája 2020]. Dostupné z: http://pypl.github.io/PYPL.html.

[3] Clement, J. *Number of apps available in leading app stores 2020* [online]. 2020. Máj 4 2020 [cit. 20. mája 2020]. Dostupné z: https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/.

[4] Grossman, D. A. a Frieder, O. *Information Retrieval: Algorithms and Heuristics*. 2. vyd. Chicago, IL, U.S.A.: Springer, 2004. 332 s. ISBN 978-1-4020-3004-8.

[5] Han, J., Kamber, M. a Pei, J. *Data mining: concepts and techniques*. 3. vyd. 225 Wyman Street, Waltham, MA 02451, USA: Morgan Kaufmann Publishers, 2012. 703 s. ISBN 978-0-12-381479-1.

[6] Hoffman, P., Sullivan, A. a Fujiwara, K. *DNS Terminology* [online]. 2015. December 2015 [cit. 15. mája 2020]. Dostupné z: https://tools.ietf.org/html/rfc7719.

[7] Kurose, F. J. a Ross, W. K. *Computer Networking A Top-Down Approach*. 6. vyd. One Lake Street, Upper Saddle River, New Jersey 07458: Addison-Wesley, 2013. 862 s. ISBN 978-0-13-285620-1.

[8] Kurtz, A., Gascon, H., Becker, T., Rieck, K. a Freiling, F. Fingerprinting Mobile Devices Using Personalized Configurations. *Proceedings on Privacy Enhancing Technologies*. 2016, zv. 2016, č. 1, s. 4–19. ISSN 2299-0984.

[9] Matoušek, P. *Síťové aplikace a jejich architektura*. 1. vyd. Brno: VUTIUM, 2014. 396 s. ISBN 978-80-214-3766-1.

[10] Mockapetris, P. *DOMAIN NAMES - CONCEPTS AND FACILITIES* [online]. 1987. November 1987 [cit. 14. mája 2020]. Dostupné z: https://tools.ietf.org/html/rfc1034.

[11] Mockapetris, P. *DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION* [online]. 1987. November 1987 [cit. 14. mája 2020]. Dostupné z: https://tools.ietf.org/html/rfc1035.

[12] Noris, I. *Príručka systémového administrátora* [online]. 2008. Máj 2015 [cit. 15. mája 2020]. Dostupné z: http://deja-vix.sk/sysadmin/dns.html#zone_files_records.

[13] Postel, J. *ASSIGNED NUMBERS* [online]. 1981. September 1981 [cit. 15. mája 2020]. Dostupné z: https://tools.ietf.org/html/rfc790.

[14] Schütze, H., Manning, C. a Raghavan, P. *Introduction to Information Retrieval,*. 1. vyd. New York, NY, USA.: Cambridge University Press, 2008. 506 s. ISBN 978-0521865715. Dostupné z: https://nlp.stanford.edu/IR-book/pdf/06vect.pdf.

[15] Thomson, S., Huitema, C., Ksinant, V. a Souissi, M. *DNS Extensions to Support IP Version 6* [online]. 2003. Október 2003 [cit. 14. mája 2020]. Dostupné z: https://tools.ietf.org/html/rfc3596#section-2.1.

[16] VanderPlas, J. *Python Data Science Handbook*. 1. vyd. 1005 Gravenstein Highway North, Sebastopol, CA 95472.: O'Reilly Media, Inc, 2016. 529 s. ISBN 978-1-491-91205-8.

Príloha A

Obsah CD

Priložené CD obsahuje nasledovné súbory:

- **src** - adresár so zdrojovým kódom a dátovými sadami
- **readme.md** - užívateľský manuál na spustenie skriptu
- **doc** - adresár obsahujúci súbory textovej časti
- **xsporn01_BP.pdf** - správa v elektronickej forme