VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ
DEPARTMENT OF INFORMATION SYSTEMS

VYHODNOCOVÁNÍ RIZIK INTERNETOVÝCH DOMÉN
INTERNET DOMAIN RISK EVALUATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **JAN POLIŠENSKÝ**
AUTHOR

VEDOUCÍ PRÁCE **Mgr. Ing. PAVEL OČENÁŠEK, Ph.D.**
SUPERVISOR

BRNO 2022

Ústav informačních systémů (UIFS) Akademický rok 2021/2022

Zadání bakalářské práce

25126

Student: **Polišenský Jan**
Program: Informační technologie
Název: **Vyhodnocování rizik internetových domén**
Internet Domain Risk Evaluation
Kategorie: Web

Zadání:

1. Seznamte se s principy vyhodnocování rizik doménových jmen.
2. Analyzujte požadavky na systém, který bude vyhodnocovat rizika internetových domén.
3. Navrhněte systém, který bude analyzovat vybraná relevantní dostupná data ke známým a škodlivým doménám a aplikovat strojové učení na tato data.
4. Navržený systém implementujte.
5. Implementovaný systém otestujte na reálných datech.
6. Získané výsledky vyhodnoťte a navrhněte další možnosti rozšíření.

Literatura:

- Kurose, J. F. Computer networking: A top-down approach. Pearson, Essex, 2017. ISBN 978-1-292-15359-9.
- Stallings, W. Network security essentials: Applications and standards. Hoboken, 2016. ISBN 978-0-13-452733-8.
- Ahmed, M., Mahmood Naser, A., Hu, J. A survey of network anomaly detection techniques. Journal of network and computer applications. Elsevier, 2016, 60(C), s. 19-31. ISSN 1084-8045.
- Ma, J., Savage, L. S., S., Voelker, G. M. Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, s.1245-1254. ISBN: 978-1-60558-495-9.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 3 zadání.

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

Vedoucí práce: **Očenášek Pavel, Mgr. Ing., Ph.D.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 11. května 2022
Datum schválení: 1. listopadu 2021

Abstrakt

Internetové domény hrají klíčovou roli při poskytování webových služeb. Je tedy nezbytně nutné umět škodlivé domény odhalit. V této práci jsou rozebrány přístupy ke klasifikaci internetových domén, jakožto i výběr a zdroje dat, použitých k této klasifikaci. Pozornost je především zaměřena na metody strojového učení neuronovými sítěmi a metodou podpůrných vektorů. Pomocí těchto metod je implementován systém, který vykazuje 96,3% přesnost na testovacích datech.

Abstract

Internet domains play a crucial role in providing web services. It is necessary to be able to detect malicious domains. The aim of this thesis is to summarize current work on domain classification and develop improved system of classification. This system is implemented using support vector machines a neural networks and shows 96.3% accuracy on test data.

Klíčová slova

rizika domén, klasifikace doménových jmen, neuronové sítě, support vector machines

Keywords

Internet domains, domain classification, neural networks, support vector machines

Citace

POLIŠENSKÝ, Jan. *Vyhodnocování rizik internetových domén.* Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Mgr. Ing. Pavel Očenášek, Ph.D.

Vyhodnocování rizik internetových domén

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením mgr. Ing. Pavla Očenáška Ph.D. Další informace mi poskytl Ing. Petr Chmelař ze společnosti Greycortex. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

. . . .
Jan Polišenský
3. května 2022

Poděkování

Chtěl bych především poděkovat Petru Chmelaři za jeho ochotu při konzultacích a za poskytnuté prostředky pro realizaci. Dále bych chtěl poděkovat mému vedoucímu Ing. Pavlu Očenáškovi Ph.D za formální vedení práce.

Obsah

Kapitola 1

Úvod

Doménová jména a jejich rezoluce jsou ve světě internetu naprosto klíčové. Hlavní funkcí systému DNS[1] je, jak již název napovídá, překládát lidmi zapamatovatelná doménová jména na adresy IP, které pak jednoznačně identifikují počítače v síti. DNS se tedy přímo nestará o samotné zajištění chodu služby, ale o určité zpřístupnění služeb uživatelům přes doménová jména.

Můžeme říci, že webové domény se staly centrálními body internetu, přes které probíhá nezanedbatelná část komunikace. Je proto nezbytně nutné řešit bezpečnost i na úrovni doménových jmen. Mezi triviálnější příklady zneužití tohoto systému může být prosté šíření spamu či reklamy. Sofistikovanějším zneužitím pak může být tvoření kopií renomovaných webů, tzv. evil twins. Zde jsou většinou uživatelé lákáni, aby zadali své přihlašovací údaje či jiné pro útočníka cenné informace. Dalším příkladem může být řízení botnetů skrze doménová jména. Jako botnet označujeme síť malwarem infikovaných počítačů obvykle řízených z centrálního bodu, které jsou využívány například pro útoky typu DDOS. Pro generování domén za tímto účelem jsou využívány speciální algoritmy, DGA, které budou rozebrány níže.

Autoři maligních domén následně často spoléhají na poměrně jednoduchý princip. Prostřednictvím emailu, jeho příloh nebo reklamy donutí uživatele navšívit danou doménu, přičemž je vždy nutná alespoň nějaká interakce, např. kliknutí na odkaz, na reklamu či zprávu. Tomuto by bylo možné předejít, kdyby mohli být uživatelé předem informování, že doména, kterou hodlají navštít, nemusí být bezpečná. Musíme tedy řešit problém, zda je možné z identifikátoru URL určit, jestli je doména potenciálně nebezpečná nebo nebo tomu tak není. [15]

Právě tato otázka bude předmětem řešení mé bakalářské práce. Nejprve budou v kapitole 2 představeny existující teoretické přístupy k této problematice. Dále budou definovány obecné požadavky na funkčnost a architekturu systémů v kapitole 3. Diskutovány budou také vhodné klasifikační metody. Na tomto teoretickém základu bude následně navržen vlastní systém vyhodnocování rizik internetových domén, tento návrh je pak možné nalézt ve 4 kapitole. V kapitole 5 bude rozebrána samotná implementace tohoto systému. Na závěr v kapitole číslo 6 jsou pak uvedeny výsledky a způsob testování implementovaného systému.

---

[1] Domain Name System

Kapitola 2

Současné přístupy ke klasifikaci doménových jmen

2.1 Doménové jméno

Doménové jméno bychom v nejobecnější rovině mohli nazvat unikátním identifikátorem v síti, přes který lze přistoupit ke konkrétnímu počítači [15]. Na doménové jméno se ale dá také nahlížet jako na DNS záznam ve stejnojmenné databázi. Konkrétně pak jde o záznamy typu A, pokud mapuje na adresu IPv4 nebo AAA pokud překládá IPv6 adresu.

Obecný formát doménového jména můžeme definovat jako **name.domain.tld**, kde **name** označuje jméno samotné domény, část **.domain** pokud je přítomna, pak značí doménu druhé nebo nižší úrovně. Konečně **tld** pak značí doménu nejvyšší úrovně. Tyto domény pak registruje přímo asociace ICANN[1].

Jako příklad může být uvedena adresa **fit.vutbr.cz**, kde name je **fit**, doména druhé úrovně pak **.vutbr** a tld **.cz**. Této hierarchické struktury se s výhodou využívá nejen při dns rezoluci, ale dovoluje také pro každé doménové jméno zjistit odpovědného registrátora, mailový server, dns server a mnoho dalšího. Toto bude obzvlášť důležité při sběru a zpracování dat. [11]

2.2 Existující přístupy ke klasifikaci

Následně budou představeny tři existující přístupy ke klasifikaci domén a výběru dat pro tuto klasifikaci. Jedná se o dva zástupce vyhodnocení na základě malého množství dat pocházejících převážně z lexikální analýzy. Dále o jednu práci představující princip klasifikace na základě velkého množství údajů. Na závěr budou rozebrány dvě práce diskutující možnosti využití strojového učení při ochraně před kybernetickými útoky na webu a detekci maligního DNS provozu.

2.2.1 Segmentace doménového jména

Klasifikace za použití segmentace doménového jména je ze tří uváděných přístupů přístupem nejvíce odlehčeným. Možnosti tohoto řešení shrnují ve své práci W. Wang a K.E Shirley [20]. Jádrem jejich řešení je segmentace doménového jména na lexikální tokeny. Například doména "duckduckgo.com"bude segmentována na tokeny "duck"a "go". Na základě četnosti

---

[1] Internet Corporation for Assigned Names and Numbers

výskytu jednotlivých tokenů v souboru názvů vygenerovaných maligních domén jsou následně prováděny predikce. K samotnému vyhodnocení neslouží celý identifikátor URL, ale pouze TLD a doména druhé úrovně. Tedy z kompletní URL *http://www.more.example.com/path-to-url.html* bude extrahováno TLD *.com* a doménové jméno *example*. Kromě výše zmíněných lexikálních tokenů jsou ke klasifikaci využity čtyři základní charakteristiky s uvedenou střední hodnotou výskytu.

1. **Počet písmen -** Počet písmen v doménovém jménu bez TLD, střední hodnota 11.5
2. **Počet spojovníků** - Počet výskytu znaku "-", střední hodnota 0.12
3. **Počet číslic -** Myšleno jako počet znaků "0-9", střední hodnota 0.09
4. **Počet čísel -** Soubor po sobě následujících číslic, střední hodnota 0.04

Čím větší odchylka od těchto hodnot pro je danu doménu identifikována, tím spíše bude klasifikována jako maligní. Vyhodnocován je také počet konkrétních písmen v názvu domény. Obdobným přístupem byly klasifikovány také TLD.

Využití této odlehčené metody může být v implementacích real-time systémů, kdy se využije její rychlost, která ovšem je na úkor přesnosti. Tato metoda nebyla vyvíjena k použití jako samostatný systém detekce, ale spíše jako hrubé síto pro následnou aplikaci přesnějších klasifikačních systémů. [20]

2.2.2 Využití URL charakteristik

Širší využítí charakteristik extrahovaných z URL, a také zapojení veřejné databáze Whois, demonstrují D. Kevin McGrath a Minaxi Gupta[13] v práci s názvem *Behind Phishing: An Examination of Phisher Modi Operandi*. Jak již je z názvu patrné, autoři se více zaměřují na domény zapojené v phishingových kampaních.

I zde je URL segmentováno na TLD a samotné doménové jméno, jako při klasifikaci podle [20], postup vyhodnocování je však odlišný. Místo rozdělování názvu domény do lexikálních tokenů je doména vyhodnocována jako celek. Navíc je hodnocena i cesta k danému souboru. Tedy pro příklad *http://www.xyz.example.com/dir/doc.html* je cesta *dir/doc.html*.

Pozornost je zaměřena na skladbu domény s ohledem na konkrétní znaky a jejich unikátnost, dále je uvažována reputace TLD a reputace případných domén nižší úrovně. Další zkoumanou součástí pak je přítomnost názvu známých značek v doménovém jméně. Toto je možno označit jako věc čistě specifickou pro phishingové domény, neboť autoři takovýchto domén se pokoušejí co nejvíce navodit dojem renomované stránky.

Vyhodnoceny jsou také údaje z databáze Whois. Autoři se zaměřují především na datum registrace, registrátora a datum expirace. Tyto ukazatele pomáhají podchytit specifickou dobu života phishingových domén, stejně tak nižžší průměrnou reputaci jejich registrátorů.

Autoři pak docházejí k zajimavým výsledkům, které je možné shrnout do následujících bodů.[13]

- Ačkoli názvy phishingových domén se od názvů domén benigních liší jak délkou tak složením, ve většině případů phishingové domény obsahují název značky nebo domény, kterou se pokouší napodobit.
- Častým jevem je zneužívání zdarma dostupných hostingových služeb a služeb nabízejících URL aliasy, jako například TinyURL[2].

---

[2] `https://tinyurl.com/app`

- Dalším rysem phishigových domén je, že jsou aktivní téměř okamžitě po své registraci, čímž se zkracuje čas dostupný pro jejich odhalení.

2.2.3 Klasifikace na základě rozsáhlých dat

Výsledky klasifikace za použití rozsáhlých zdrojů dat jako DNS, Whois, vlastních měření a dalších je dobře demonstrována např. v práci J.Ma, K.Lawrence a kolektivu [12]. Tato práce bude sloužit jako hlavní opora implementace vlastního systému v rámci této bakalařské práce. Zpracovávaná data autoři seskupují do čtyřech kategorií:

1. **Vlastnosti IP adresy -** DNS záznamy, jejich hodnoty, přítomnost na blacklistech,
2. **Data z Whois** - Registrátor, datum registrace a expirace a další
3. **Charakteristiky doménového jména -** RTT, lexikální analýza
4. **Geografické údaje -** Geografická lokalita, rychlost přenosu

Geografické údaje a rychlost přenosu jsou velmi užitečnými údaji. Výsledky ukazují, že maligní stránky mívají zpravidla menší rychlost přenosu a větší odezvu, což je i intuitivně možno očekávat. Systém dále využívá rozsáhlá data z volně dosupných blacklistů.

Prováděna je také anaýza lexikálního složení domén, ovšem na rozdíl od [20] je tato klasifikace spíše doplňkem k vyhodnocení ostatních parametrů. Dalším rozdílem je velká pozornost věnovaná výběru vhodného klasifikačního modelu. Porovnávány jsou metody strojového učení jako Support Vector Machines (SVM), Naivní Bayesovské klasifikátory a Logistická regrese. Autoři prokázali, že statistické modely vykázaly větší chybu díky velké dimenzi vstupních dat, naopak pomocí SVM metody došli k excelentním výsledkům.

Práce odpovídá na základní otázku, a totiž, zda zvyšování množství dostupných dat vede k přesnější klasifikaci. Autoři v závěru demonstrují, že takovýto přístup umožňuje klasifikovat domény s 95-99% přesností s pouze malým počtem falešně pozitivních výsledků. [12]

2.2.4 Využití strojového učení k detekci maligních domén

Širokou škálu využití strojového učení při ochraně před kybernetickými útoky prezentuje F. Strasak ve své práci s názvem *Strojové učení k ochraně před kybernetickými útoky na webu* [6]. Podstatná část práce je věnována právě detekci maligních doménových jmen a vhodným metodám strojového učení pro tuto detekci.

Na rozdíl od již uvedených metod jsou domény klasifikovány na základě samotného obsahu. Zpracováván je HTML DOM pomocí grafových algoritmů a tato data jsou následně klasifikována konvolučními neuronovými sítěmi. K tomu je využita knihovna Pytorch.

Tato implementace je volně dostupná ve formě webové aplikace, která nabízí možnost ohodnotit malignost zvoleného doménového jména. Aplikace je přístupná na adrese shouldiclick[3].

2.2.5 Techniky pro detekci maligního DNS provozu

Principy detekce maligního DNS provozu prezentují ve své práci G.Chukwunonso, D. Jaye S. Agron a kolektiv [2]. Obvzvlášť zajimavý je návrh jejich architektury neuronové sítě,

---

[3] `https://www.shouldiclick.org/`

kdy úspěšně využívají architekturu **long short-term memory**. Ačkoli v rámci této BP bude implementován systém pro detekci podezřelých doménových jmen a nikoli pro detekci podezřelého provozu. Princip zpracování DNS je z velké části přenositelný, a tak může sloužit k inspiraci. Mezi nejpodstatnější parametry použité pro klasifikaci patří například **rychlost odpovědi** nebo **medián a průměrná délka paketů** při DNS dotazu.

2.3 Maligní domény

Škodlivé, nebo maligní domény, je pojem zahrnující více typů internetových domén. Všechny pak mají za cíl uškodit koncovému uživateli.

Maligní domény můžeme rozdělit do dvou základních typů. Jedním jsou phishingové domény, jejichž cílem je ukrást uživateli přihlašovací údaje, či jakákoli jiná data. Druhým typem jsou domény, které obsahují malware. Ten se zpravidla pokouší dostat do zařízení uživatele a tím ho infikovat.

2.3.1 Phishingové domény

Phishingové domény se pokouší ukrást uživateli data, jako například hesla, emailové adresy nebo čísla kreditních karet. Ve většině případů se využívá prosté nepozornosti uživatelů. Odkazy na takové stránky se mohou šířit například emaily.

"Evil twin"domény

Nejznámějším druhém phishingových domén jsou takzvané *Evil twin* domény. Jedná se o klony, či spíše imitace zavedených důvěryhodných stránek. Ačkoli kvalita takovýchto napodobenin v posledních letech stoupá, stále můžeme pozorovat zásadní rozdíly v provedení takovýchto stránek oproti jejich legitimním předlohám. Mezi nejčastější rozdíly pak patří:

- Nízká kvalita použitých obrázků. Toto je velmi nezvyklé u jakékoli legitimní stránky.[6]
- Na stránce se objevují falešné odkazy, které nikam nesměřují. Většinou se jedná o odkazy na licenční podmínky či podmínky užívání. [6]

2.3.2 Domény obsahující malware

Cílem těchto domén je místo prostého odcizení údajů, přímo infikovat zařízení uživatele. Toho může být dosaženo buď využitím chyby v prohlížeči uživatele, bez jeho vědomí. Tento scénář pak nebývá tak častý, jelikož musí cílít na konkrétní chybu v konkrétní verzi prohlížeče. Častějším způsobem pak je donucení uživatele aby si přímo škodlivý software nainstaloval.

Zajímavým zjištěním prezentovaným F.Strasak [6] je fakt že většina domén obsahujících malware jsou původně legitimní domény, které byly napadeny útočníky. To může významně ztížit jejich odhalení, jelikož tyto domény nebyly vytvářeny za účelem maligního chování.

Častým trikem takovýchto stránek bývá informovat uživatele o nějaké výhře, například nového počítače, telefonu atd. kdy zbývá jen určitý čas a uživatel musí kliknout na daný odkaz nebo stáhnout požadovaný software.

Reklamní domény

Za maligní aktivitu je také považována nevyžádaná reklama. Jedná se sice o mírnější typ útoku na koncového uživatele, stále se však jedná o maligní aktivitu. Zajimavý přístup k detekci takovýchto domén se nabízí v práci Zhang a kol. [22]. Prezentuje zde hypotézu, že reklamní domény jsou dobře detekovatelné na základě jejich URL.

Na základě softwaru Nutch[4] a Modsecurity[5] je implementován vlastní systém detekce reklamních domén, který ve srovnání s referenčním systémem *360 secured browser* vykazuje více než dvojnásobně vyšší přesnost detekce.

---

[4] Apache Nutch je modifikovatelný a snadno škálovatelný softwarový systém s otevřeným zdrojovým kódem

[5] ModSecurity je webový aplikační firewall. Je určen pro Apache HTTP servery a má opět otevřený zdrojový kód.

Kapitola 3

Zdroje dat internetových domén a metody jejich zpracování

Správný výběr a ohodnocení dat jsou alfou i omegou každého úspěšného klasifikačního modelu. Je jim tedy třeba věnovat náležitou pozornost. Z konceptu klasifikace na maligní a benigní domény vyplývá požadavek na dva datasety. První s daty nezávadných domén, druhý s daty škodlivých domén, a to škodlivých z jakéhokoli důvodu, např. spam, botnet, reklama, atd.. Pokud jde o obsáhlost datasetů, J.Ma, K.Lawrence a kolektiv [12] využívají soubory dat o obsáhlosti cca 15000 domén. Tento počet bude v této BP vyšší, protože metoda strojového učení neuronovými sítě a metodou podpůrných vektorů vyžaduje řádově větší množství dat. Na základě práce A. Gupty a kolektivu [8], pojednávající o klasifikaci vícedimenzionálních dat, je spodní hranice velikosti datasetů cca 100 až 300 tisíc domén na jeden soubor dat.

Primárním zdrojem benigních domén v předkládané BP byl seznam 1 000 000 nejvyhledávanějších webů, který uveřejňuje společnost Amazon[1]. Z tohoto seznamu bylo vyfiltrováno přibližně 400 000 domén, které budou dále implicitně považovány za benigní. Zdroje maligních domén byly dvojího typu. Jednak bylo využito veřejných blacklistů, odkud bylo staženo zhruba 150 000 závadných domén. Další část, cca 250 000 domén mi ochotně poskytl Ing. Petr Chmelař ze společnosti Geycortex, za což mu velmi děkuji.

Při samotném zpracování dat bylo třeba vzít v úvahu jednu zásadní skutečnost. Domény jsou velmi variabilní v množství dostupných dat a vyvstává otázka, zda aplikovat strojové učení i na vektory proměnlivé délky nebo zda nedostupná data nahradit speciální hodotou nesoucí informaci o jejich nedostupnosti nebo domény s nedostatečným množstvím dat ze zpracování vyloučit. Vzhledem k poměrně rozsáhlému množství dat byla zvolena třetí možnost, a totiž filtrování domén s nízkým počtem zjistitelných parametrů. Více podrobností k této metodě se pak nalézá v kapitole číslo 5, která obsahuje popis implementace.

---

[1] https://www.alexa.com/topsites

3.1 Zdroje dat k analýze

Data pro analýzu malignosti domén je možno čerpat z celé řady zdrojů. Pro účely této Bakalářské práce budou tyto zdroje dat rožděleny do čtyř kategirií. Toto rozdělení umožní pohodlnější zpracování a ohodnocování dat. V této sekci se pak nalézá popis konkrétních využívaných dat.

1. **Whois**
2. **Geografická data**
3. **DNS data**
4. **SSL/TSL data**

První skupinou jsou data z veřejně přístupné databáze Whois. Zde je zpracováván registrátor a datum registrace a expirace. Tato data využili k úspěšné detekci phishingových domén např. D. Kevin McGrath a Minaxi Gupta [13]. V tomto směru na jejich práci tato BP navazuje. Nad rámec jejich práce jsou v předkládané BP zpracovány údaje o přítomnosti DNSSEC u dané domény a přidružené emailové adresy. Využití dat z databáze Whois je popsáno v kapitole 3.1.1.

Dalšími zpracovávanými údaji jsou geografická data, čímž je míněno umístění, tj. fyzická lokalita dané IP adresy, do této kategorie jsou zařazeny i fyzické parametry serveru, např. RTT a konektivita. Podrobně jsou rozepsány v kapitole 3.1.2.

K široce využívané skupině dat patří jednotlivé DNS záznamy a hodnoty z nich vyčtené. Stahovány jsou nejpoužívanější záznamy jako A, AAAA, MX, atd., více ke zpracovávání DNS dat je popsáno v kapitole 3.1.3.

Poslední skupina dat zahrnuje údaje o existenci zabezpečeného připojení SSL. V případě, že jej doména poskytuje, je dále zkoumán certifikát a příslušná certifikační autorita. Množství zkoumaných dat je opět rozsáhlé, popis je v kapitole 3.1.4.

3.1.1 Whois

Whois je označení pro distribuovanou databázi, která uchovává informace o majitelích internetových domén a IP adresách. Motivací pro vytvoření tohoto systému byla regulace registrace doménových jmen a nutnost transparentnosti těchto záznamů. Mezi základní informace uchovávané v rámci Whois patří kontaktní informace majitele dané domény, registrátor, datum registrace, datum poslední změny, nameservery a další. Existují dva základní typy Whois záznamů, a to "thin"a "thick". První typ obsahuje pouze jméno a odkaz na konkrétní Whois server, který spravuje údaje o dotazované doméně. Ve druhém typu je pak možné nalézt již kompetní sadu údajů.

V globálním měřítku za správu zodpovídá organizace ICANN[2]. Struktura Whois je hierarchická, tedy každý národní registrátor má povinnost vést vlastní Whois server a udržovat informace o doménách pod ním registrovaných. [4]

3.1.2 Geografická data

Geografická data pomáhají určit korelaci mezi servery hostujícími maligní stránky a jejich fyzickými parametry. Pomocí nich je možné ověřit např. intuitivní předpoklad, že hostingové

---

[2] Internet Corporation for Assigned Names and Numbers

služby s menší reputací, hojně využívané pro tento účel, budou mít horší parametry jako jsou konektivita a RTT. K účelu sběru těchto dat byla využita volně dostupná služba ipinfo[3]. Ke sběru údajů o konektivitě posloužil příkaz *ping*, běžně dostupný v unixových operačních systémech.

3.1.3 DNS

DNS je databáze umožňující lokalizovat libovolné doménové jméno nebo službu k ní přidruženou. Jedná se o decentralizovaný a hierarchický systém, kdy je ke každé doméně přiřazen DNS server spravující informace o této doméně. Narozdíl od systému Whois, který má spíše informativní účel, je DNS využívano téměř všude, kde je třeba provádět směrování v rámci internetu.

Jádrem celého systému je skupina kořenových DNS serverů, které obsahují odkazy na DNS servery TLD domén. Dotazem na tyto kořenové servery je tedy možné zjistit adresu libovolné domény.

Hlavním úkolem a příčinou jeho vzniku jsou vzájemné převody doménových jmen a IP adres uzlů sítě. Avšak rozsah služeb zajišťovaný tímto standartem je daleko větší. Využívá se například také při směrování elektronické pošty nebo navazování VoIP hovorů. [11]

Data z DNS představují klíčovou část zpracovávaných dat. Je například nezbytné, aby byl u každé domény přitomen záznam typu A nebo AAAA, a to za účelem přeložení daného jména na adresu IP. Bez těchto záznamů je doména v praxi nepřístupná a navíc nám znemožňuje provádět dotazy týkající se fyzických parametrů serveru (geografická lokalita, RTT, konektivita).

Výčet DNS záznamů využitých v rámci BP a jejich specifikace:

- **A -** Překlad doménového jména na adresu typu IPv4.
- **AAAA -** Obdobně jako A záznam, pro adresy typu IPv6.
- **CNAME -** Mapuje alias serveru na kanonické jméno počítače.
- **SOA -** Název primárního serveru a emailovou adresu správce.
- **NS -** Určuje autoritativní server.
- **MX -** Mailový server pro danou doménu.
- **TXT -** Různé použití, mohou se zde vyskytovat autorizace služeb třetích stran.

3.1.4 SSL/TLS data

Protokol SSL a jeho nástupce TLS jsou protokoly poskytující šifrování na aplikační vrstvě. Velmi často se používají k zabezpečení právě HTTP komunikace. Toto zabezpečení zaručuje integritu dat, autentizaci serveru a možnou, avšak ne příliš často využívanou autentizaci klienta. [11] Údaje související se zabezpečeným připojením jsou poslední skupinou analyzovaných údajů v rámci BP. jejich využití se opírá o hypotézu, že méně důvěryhodné weby budou toto šifrování postrádat nebo budou využívat certifikáty s kratší platností. Pokud tedy internetová doména poskytuje zabezpečené připojení, jsou staženy údaje o vydavateli certifikátu, samotný certifikát, datum jeho vydání a datum expirace. Klíčovým údajem je pak délka platnosti certifikátu a jeho vydavatel.

---

[3] `https://ipinfo.io/`

3.1.5 Public sufix list (PSL)

Public sufix list, dále jen PSL. Je veřejný katalog přípon internetových domén. Ke každé doméně jsou pak přiřazeny záznamy nížších úrovní. Cílem je tedy vytvořit strom doménových jmen, kdy jsou každému registrátorovi přiřazeny přípony, které spravuje. PSL byl původně vyvíjen prohlížečem Mozila pro jeho potřeby. Dle oficiálního webu `https://publicsuffix.org/` umožňuje: Zvýraznění nejdůležitější části url v prohlížeči, vyhnout se problémům se soukromím při použití *supercookies* nebo také přesněji řadit historii vyhledávání pro konkrétní stránky. PSL je dnes volně dosupný a udržovaný komunitou.

3.2 Metody klasifikace vícerozměrných vektorů

Z formátu zpracovávaných dat je zjevné, že jde o data s vyšší dimenzí. Tento fakt je třeba vzít v úvahu při výběru vhodné klasifikační metody. Podle [8] nejsou vhodné prosté statistické klasifikátory, jako například logistická regrese. Zvažovanou metodou pak mohou být bayesovské klasifikátory, které úspěšně využívá [12]. Tato metoda by dokázala zpracovat data o větší dimenzi, ovšem stále se jedná o statistickou metodu a tedy by byl potřeba daleko větší poměr dimenze vstupního vektoru a množství dat, aby se začaly projevovat statistické skutečnosti. Z tohoto důvodu tato metoda také není pro daný účel vhodná. [17]

3.2.1 Support Vector Machine

Potenciálně vhodnou metodou je Support vector machines, přeloženo pak metoda podpůrných vektorů, dále již jen SVM. SVM řadíme do metod strojového učení s učitelem. Metoda byla původně konstruována jako podpůrná metoda pro učení neuronových sítí. Základem metody SVM je lineární klasifikátor do dvou tříd. Cílem je rozdělit prostor příznaků do dvou podprostorů. Hlavním jádrem úspěchu SMV je transformace do vyšší dimenze, dokud rovinu separující prostor příznaků nelze rozdělit ve vyšší dimenzi. [18]

Kernel SVM

Kernel SVM je funkce, pomocí které tato metoda provádí klasifikaci. Volba kernelu pak určuje základní klasifikační schopnosti celého modelu. Jinak řečeno, určuje jaká data bude metoda schopna klasifikovat. Mezi nejpopulárnější typy kernelů patří **Lineární**, **Polynomický**, **RBF** a **Sigmoid**. Srovnání použití jednotlivých typů kernelů na škále dat ve své práci s názvem *Role of Kernel Parameters in Performance Evaluation of SVM* prezentují A.Goel a S. Srivastava [7].

Využití SVM

SVM se využívá pro regresi a klasifikaci. Obvzlášť vhodná je tato metoda pro vstupní data s vysokou dimenzí. Metoda neztrácí na přesnosti ani když dimenze vstupních dat překoná počet vzorků. Příklad využití pak prezentuje M.H Selemat v práci s názvem *Image face recognition using Hybrid Multiclass SVM* [16]. Zde je pomocí metody SVM trénován klasifikátor na rozpoznávání obličejů.

3.2.2 Neuronové sítě

Neuronové sítě, představují starší a obecnější přístup ke klasifikaci než SVM. Koncept neuronových sítí pak vychází ze studia lidské nervové soustavy. Základním stavebním prvkem neuronových sítí je **neuron**. Tyto neurony jsou navzájem propojeny do sítě a platí, že každy může mít více vstupů ale pouze jeden výstup. Každý vstup neuronu má pak určitou váhu. Tyto vstupní hodnoty jsou v neuronu transformovány dle příslušné **aktivační funkce**. Z hlediska konceptu je možno říci, že neuronové sítě provádí určitou transformaci do vyšší dimenze. Tato metoda je pak obvzvlášť vhodná na datasety opravdu velkých rozměrů, kdy metoda SVM přestává být efektivní. [1]

Aktivační funkce

Aktivační funkce transformuje vnitřní potenciál neuronu na normalizovaný výstup. Zjednodušeně, je to funkce pomocí které provadí neuron transformaci vstupů na výstupy. Volba vhodné aktivační funkce je důležitým prvkem při návrhu každé neuronové sítě. Nejjednodušším příkladem může být prostá **lineární funkce**. Jelikož ale tato funkce degraduje neuronovou síť na lineární klasifikátor, není moc využívána. Mezi nejpoužívanější funkce pak patří **sigmoida**. Ta je obvzvlášť vhodná při binární klasifikaci, kdy škáluje svůj vnitřní potenciál do rozsahu 0 až 1.

3.2.3 Architektury neuronových sítí

V následující sekci bude představeno několik architektur neuronových sítí. Pozornost pak bude zaměřena na ty, které by mohly najít potenciální využití při naší doménové klasifikaci.

Konvoluční neuronové sítě

Konvoluční neuronové sítě svůj název odvozují od speciální **konvoluční vrstvy**, kterou obsahují. Tato vrstva extrahuje ze vstupních dat užitečné informace (při zpracování obrazu například detekce hran) a předává je zbytku klasifikátoru. Toto předzpracování pak dramaticky snižuje množství trénovacích dat nutných k vytrénování neuronové sítě. Efektivně tedy zpracovávají vstupy o velké dimenzi. Typicky se využívají při rozeznávání písma, obrazu, obecně pak v odvětví strojového vidění. [1]

Long short-term memory

Long short-term memory, dále jen LSTM, je architektura neuronových sítí, která se používá v oblasti hlubokého učení. Rozdíl oproti standartním neuronovým sítím je to, že LSTM má zpětnovazebná propojení. Tato architektura je pak obvzvlášt vhodná na zpracování sekvencí dat, jakožto i zpracování přirozeného jazyka. V rámci BP se tato architektura využije při lexikální analýze názvu domény. [1]

Modulání neuronové sítě

Modulární neuronové sítě je architektura, která se skládá z více nezávislých neuronových sítí. Princip je pak takový, že každá dílčí síť provádí úkol, na který je specializována. Jedná se o další paralelu s lidským nervovým systémem, kdy přesně takto jsou neurony organizovány v lidském mozku. [5]

3.2.4 Výběr a filtrování klasifikačních dat

Ačkoli jsou výše zmíňené metody vhodné pro klasifikaci vícerozměrných dat, jsou situace, kdy se úpravě dat před samotným zpracováním zkrátka nevyhneme. Úpravou se zde myslí redukce nebo nějaké zmenšení. Většinou pak z důvodu snížení výpočetních nároků. Otázka kterou je pak nutné se zabývat je následující: Redukovat dimenzi dat, nebo z dat vybrat vhodné parametry? V této sekci je nabídnuto porovnání obou přístupu. [3]

Výběr parametrů

Výběr parametrů (feature selection) je jednoduchý přístup, kdy ze vstupního vektoru vybereme pouze některé položky bez toho aniž bychom je měnily. Metod jak provádět tuto selekci je celá řada. Obecně bychom je mohli rozdělit do tří kategorií: [9]

- **Filter-based** jsou metody založené na filtrování dat s nastavenou metrikou. Můžeme například odebrat parametry kterým chybí hodnoty nebo parametry s nízkou variabilitou hodnot.
- **Wrapper-based** metody nahlíží na výběr parametrů jako na problém vyhledávání. Příkladem může být metoda *Recursive Feature Elimination*.
- **Embedded** metody využívají algoritmy, které mají již jako součást své funkcionality zabudovaný výběr parametrů. Například algoritmus *LASSO*[4].

Redukce dimenze

Redukování dimenze (dimension reduction) je proces transformace dat do nižší dimenze. Ačkoli je cíl podobný jako při výběru parametrů, data jsou zde převedena jako celek. Jednou z nejpoužívanějších metod je **Principal component analysis (PCA)**. Metoda PCA se využívá pro extrahování důležitých informací ze sady dat a jejich transformace do menšího počtu hlavních komponent (principal components). Tyto nové komponenty korespondují s linární kombinací originálních dat. [21]

3.2.5 Srovnání redukce dimenze a výběru parametrů

Redukce dimenze i výběr parametrů jsou obě efektivní metody jak snížit množství vstupních dat pro model strojového učení. Výběr parametrů je podstatně jednodušší, ovšem hrozí zde riziko ztráty dat. Metody redukce dimenze se naoapk snaží zachovat co největší množství informace. [3]

---

[4] Algoritmus upravuje parametry modelu pomocí snižování koeficientu regrese

Kapitola 4

Návrh a požadavky na systém

Tato kapitola je věnována samotnému návrhu systému. V první části budou nejprve definovány požadavky a očekávaná funkcionalita systému. Na tomto základě bude systém konceptuálně navrhnut. Dále budou rozpracované jednotlivé části a jejich celková architektura.

4.1 Požadavky systému

Základní požadavek na systém vyhodnocující rizika internetových domén je samotná přesnost klasifikace. Tedy aby byl systém opravdu schopen rozeznávat mezi maligními a benigními doménami. K tomuto funkčnímu požadavku se následně váže několik nefunkčních požadavků, ty by se daly shrnout takto:

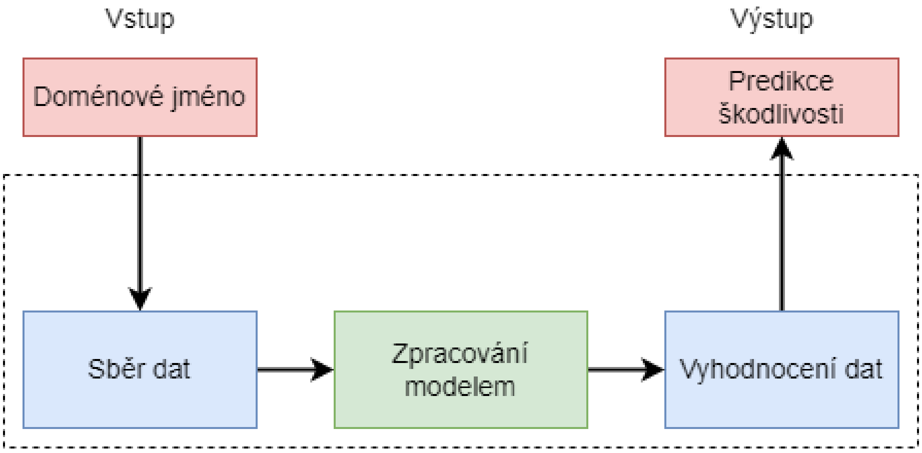
- **Přesnost**
- **Časové odezva**
- **Interpretovatelnost**

Nejedná se o kompletní výčet požadované funkcionality, ale o hlavní prvky, kterými musí systém disponovat, aby mohl být považován za funkční. Přesností klasifikace je myšlena míra spolehlivosti při predikci maligních domén. Zvláštní pozornost by měla být věnována eliminaci falešně negativních výsledků, tedy doménám, jež jsou klasifikované jako bezpečné, ale nejsou. Také eliminace falešně pozitivních výsledků nesmí být přehlížena, ale představuje mnohem menší problém, neboť nemá potenciál ohrozit koncového uživatele. Toto úzce souvisí s výběrem klasifikační metody proto budu v implementační části bakalářské práce (BP) navazovat na J.Ma, K.Lawrence a kolektivu [12] a využívat především metody strojového učení SVM, a také neuronové sítě.

Časové odezva je spíše obecným požadavkem a při návrhu systému nepřímo souvisí s konstrukcí samotného modelu. Faktem však je, že systém musí poskytovat rozumnou časovou odezvu. Hlavním prvkem ovlivňující rychlost vyhodnocení je množství dat zpracovávaných o každé doméně. Odlehčený systém prezentovaný W. Wang a K.E Shirley [20] sice zaručuje vyhodnocení v téměr reálném čase, ovšem neposkytuje uživateli příliš přesnou zpětnou vazbu. Jde tedy o nalezení kompromisu mezi zpracovávaným množstvím dat a dobou výsledné odezvy, která by měla být v řádu jednotek či nižších desítek sekund. Posledním požadavkem je interpretovatelnost, a totiž, že uživatel musí být jednoznačně informován o nalezených skutečnostech. Současná řešení jako například **shouldiclick.org** zobrazují potenciální riziko pro čtyři základní kategorie: "Evil Twin", "Scam", "Unencrypted data sent"a "Dangerous behaviour". Dále se můžeme setkat s prostým hodnocením na stupnici 1 až 100.

4.1.1 Koncepce systému

Na základě definice obecných požadavků na systém je možné sestavit jeho konceptuální návrh.



Obrázek 4.1: Koncept systému

Diagram na obrázku 4.1 zobrazuje průběh samotného vyhodnocení, vstupy, výstupy a základní funkční bloky navrhovaného systému. Nejprve je do systému uživatelem zadáno doménové jméno. Toto je ve funkčním bloku **Sběr dat** přeloženo na IP adresu a pomocí řady dotazů (DNS, Whois, atd...) jsou shromážděna relevatní data. Definice a obecně formát těchto dat bude popsán v dalších částech BP. Následně jsou data zpracována a přeložena do podoby vhodné pro strojové zpracování. Dalším blokem systému je pak **Zpracování modelem**, kdy jsou data ohodnocena samotným klasifikátorem. Poslední funkční blok, **Vyhodnocení dat** reprezentuje interpretaci predikce modelu do srozumitelné podoby. Na závěr jsou data o malignosti domény zobrazena uživateli.

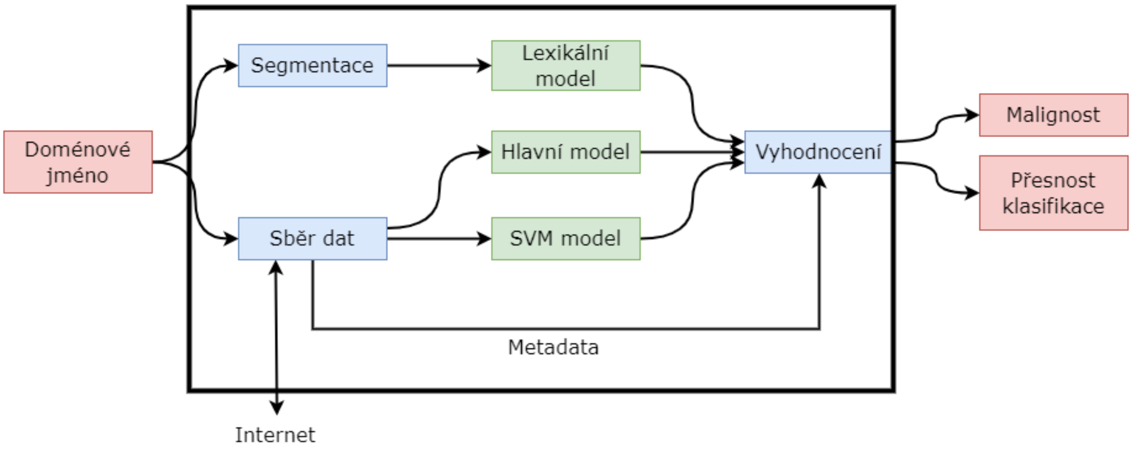
4.1.2 Architektura

Z porovnání metod strojového učení v předešlé kapitole plyne, že neexistuje univerzální metoda, která by pokryla všechna specifika při klasifikaci domén. Při návrhu způsobu klasifikace tedy bude využito více modelů, kdy každý bude vykonávat specifickou činnost. Tato architektura je inspirována modulárními neuronovými sítěmi. O zpracování dat domén se budou starat dva modely, jeden realizovaný pomocí **neuronových sítí** a druhý metodou **SVM**. Klasifikace dvěma modely nám dovolí porovnávat jejich výstup a využít naplno silné stránky každé metody.

Dále je nutné řešit zpracování samotného doménového jména. Tento úkol bychom mohli zařadit do problematiky strojového rozpoznávání textu. Pro tyto účely jsou opět vhodné neuronové sítě. Jelikož ale bude vyžadovat odlišnou architekturu než síť pro zpracování dat, je nutné vytvořit model pro lexikální zpracování jako separátní blok systému.

Jádro systému tedy bude zahrnovat následující moduly, sběr dat, hlavní datový model, SVM klasifikátor, model pro lexikální zpracování názvu domény a modul který bude hodnotit výstupy všech třech modelů strojového učení.

Při využití více modelů nastává otázka do jaké architektury tyto modely uspořádat. Intuitivně se může jevit sériové zapojení jako optimální, kdy hlavní model má k dispozici výstupy pomocných či vedlejších modelů. Ovšem není tomu tak. Klasifikace hlavním modelem selhává ve chvíli kdy dostane na vstup neočekávané hodnoty, tedy například že část dat se nemusí podařit stáhnout.



Obrázek 4.2: Architektura systému

4.2 Klasifikátory

Následující sekce se věnuje výběru klasifikačních metod pro tři klasifikátory představené výše. Dle studované literatury je možné do užšího výběru zařadit, jak již bylo zmíněno, **Support vector machines** a **Neuronové sítě**. Obě dvě metody se zdají být vhodné při klasifikaci doménových dat. Avšak každá z metod má své výhody a nevýhody. SVM ze vstupních dat vybírá nejvhodnější parametry, na základě kterých klasifikuje vstupní data. [18] Neuronové sítě využívají všechna data. Na druhou stranu když dimenze a objem dat přeroste určitou hranici, SVM se stává pomalá a neefektivní. Výsledky obou modelů, jak neuronových sítí tak SVM budou srovnány v poslední kapitole.

4.2.1 Lexikální model

Účelem lexikálního modelu, jak již název napovídá, je pak analýza doménového jména jako takového. Výhodou tohoto přístupu je rychlost zpracování a také to, že není třeba o doméně zjišťovat dodatečná data. Takovýto model se tedy využije například v situaci kdy je doména nedostupná. Model bychom tedy mohli zařadit do umělé inteligence zpracovávající přirozený jazyk. Z tohoto důvodu se jeví jako nejvhodnější využít architekturu neuronových sítí LSTM [17]. Konkrétní realizace modelu byla inspirována daty a prototypem, poskytnutým společností Greycortex, která již implementovala tento lexikální model.

Této architektuře je tedy nutné přizpůsobit formát vstupních dat. Dle [19] je vhodné použít místo prosté zakódované abecedy tzv. Bigramy.

Bigramy

Bigram je sekvence sousedních prvků z řetězce tokenů, kterými jsou obvykle písmena, slabiky nebo slova. Četnost výskytu takovýchto bigramů ve slovech a větách je bežně používaná metoda pro jednoduchou analýzu textu. Dělení slov na bigramy nabízí možnost anazyzovat text na vyšší úrovni oproti segmentaci na jednotlivá písmena. Slovník těcho bigramů se pak nachází ve složce *Data*.

4.2.2 Hlavní model

Model založený na zpracování dat domény je jádrem celého systému. Tedy jeho výstup bude z velké části definovat hodnocení dané domény, musí mu být tedy věnována náležitá pozornost. Z hlediska architektury neuronových sítí je nejprve nutné rozhodnout o aktivačních funkcích jednotlivých neuronů. Dále pak zvolit počet vrstev neuronové sítě a nakonec množství neurononů v každé vrstvě. Za tímto účelem bylo provedo několik experimentů na zmenšené sadě dat.

Aktivační funkce a škálování vstupů

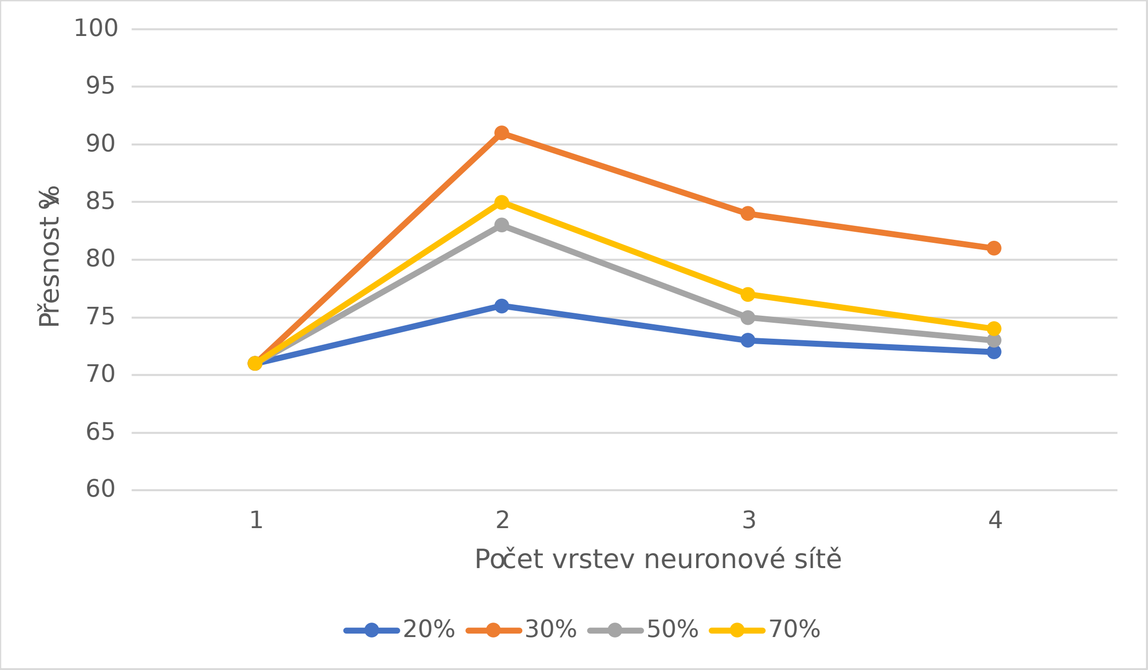
Otázku, kterou je nezbytně nutné se také zabývat je před architekturou samotného modelu také aktivační funkce a konkrétní škálování hodnot na vstupu. Co je aktivační funkce a proč je důležité se jejím výběrem zabývat bylo popsáno v druhé kapitole. Jelikož je prováděna binární klasifikaci, tak ideální aktivační funkcí pro poslední vrstvu bude funkce sigmoid. Ta zajistí škálování výstupních hodnot do intervalu 0 až 1, což je ideální výstup binární klasifikace. Toto rozpětí je možné interpretovat jako procento malignosti domény.

Počet a charakteristika vrstev

Při rozhodování o architektuře neuronové sítě, bylo čerpáno z práce P.Nocera a R.Quelavoine [14]. Ačkoli je zde prováděna studie na determinování nezbytného počtu vrstev a počtu neuronů v každé vrstvě pro rozeznávání řeči. Obecná fakta o návrhu neuronových sítí jako celku lze z jejich práce přenést do této.

Experimenty byly prováděny se zmenšenou datovou sadou, která obsahuje náhodně vybrancýh 10% dat původní datové sady. Informace o kompletní datové sadě se nalézají v kapitole 5.

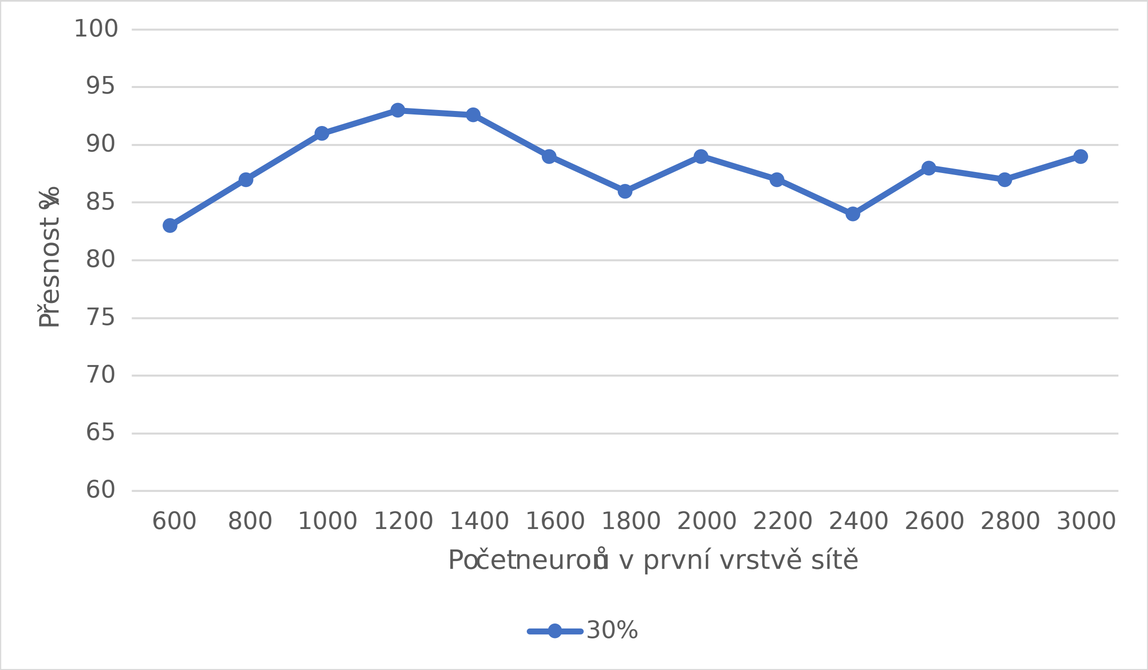
Takovou redukci je možno provést, jelikož ještě není trénován výsledný klasifikátor, pouze se rozhoduje o počtu vrstev. K hrubému odhadu takové množství stačí. Stejnou experimentální metotodou je určeno i optimální množství neuronů v jedné vrstvě. V první vrstvě bude neuronů řádově víc, pro tento experiment je velikost této vrstvy zafixována na 1000 neuronů. Množství neuronovů v dalších vrstvách se pak bude poměrově zmenšovat, rychlost tohoto zmenšování nesmí být moc velká, aby se v sítí dokázaly vyselektovat relevantní parametry, ale ani moc malá aby nedocházelo ke ztrátě dat. Koeficient tohoto zmenšování bude také předmětem experimentů.



Obrázek 4.3: Experiment s počtem vrstev

Jak vidíme z grafu 4.3, nejlepších výsledků dosahují neuronové sítě se dvě skrytými vrstvami. Počet neuronů v dalších vrstvách je pak ideální okolo 30%. Tedy pro první vrstvu o 1000 neuronech budeme mít druhou s cca 300-330 neurony. Podobným experimentem budeme hledat optimální množství neuronů v první vrstvě. Zde bude testováno větší rozmezí, mezi 600 a 3000 neurony pro vstupní vrstvu.

Na obrázku 4.4 je pak možné vidět výsledek tohoto experimentu. Největší přesnosti dosahuje síť s 1200-1400 neurony první vrstvy. Na základě této sady experimentů je již možné učit výslednou architekturu. Tvoří ji tři vrstvy neuronů. První vstupní, jejíž počet



Obrázek 4.4: Experiment s počtem neuronů v první vrstvě

odpovídá dimenzi vstupních dat. Dále dvě skryté vrstvy (hidden layers), které budou provádět samotnou klasifikaci. Nakonec pak výstupní vrstva pouze s jedním neuronem jelikož provádíme binární klasifikace na maligní/benigní.

4.2.3 SVM model

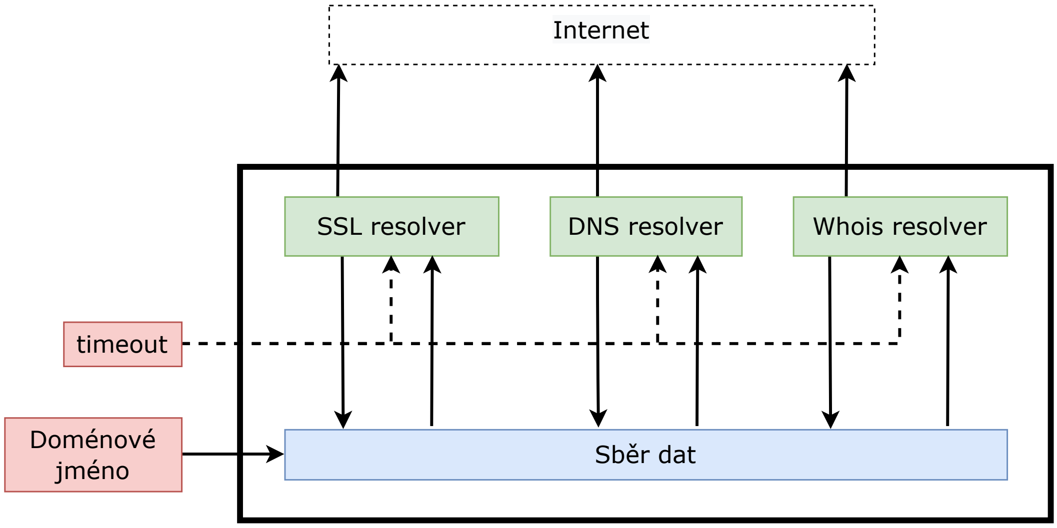
Metoda podpůrných vektorů (SVM) byla popsána v sekci výše. Nyní k samotnému trénování modelu touto metodou. Model bude trénován na stejných datech jako výše uvedený hlavní klasifikátor. Tento model částečně řeší problémy které nastávají při nekonzistentním množství dat, v takových situacích modely založené na neuronových sítích vykazují větší chybu než SVM [18]. SVM model je tedy využit jako kontrola výstupu datového modelu a jeho výstup se především promítá do vykázané přesnosti celého systému.

Výběr kernelu SVM

Důležitost samotného kernelu a jeho výběru při klasifikaci byla posána v kapitole 3. Samotný výběr byl pak inspirován prací [7]. V tomto případě bude nejvhodnější zvolit *Radial basis function kernel*, dále již RBF. Tento typ kernelu provádí se vstupními parametry podobnou operaci jako Gausovo rozložení. Je pak obvzvlášť vhodný pro datasety, které nejsou lineárně separovaténé. Z podobných důvodů by se mohl pro tyto účely hodit kernel založený na funkci sigmoid. S těmito dvěma kernely byly provedeny experimenty na testovacím datasetu, tedy 10% trénovacích dat. Zde vykazuje RBF kernel přesnost **92,4%** oproti **90,29%** při použití sigmoid kernelu. Pro implementaci byl tedy zvolen RBF kernel.

4.3 Systém sběru dat

Systém pro sběr dat, neboli první blok konceptuálního návrhu, bude zajišťovat sběr dat dle kapitoly 3. Nutným požadavkem na tento blok je rychlost. Data by měla být k dispozici k analýze co nejrychleji. Za tímto účelem je systém navrhnut paralelně.



Obrázek 4.5: Architektura systému pro sběr dat

Architektura se skládá ze čtyř bloků. Hlavní řídící blok je zde pojmenován sběr dat. Bude se jednat o funkci či třídu do které přijde požadavek na sběr dat. Tato třída pak aktivuje příslušné resolvery. Ty jsou ve schématu vyznačeny zeleně, a jejich volání bude paralelní. Naráz se tedy budou zjišťovat DNS data, data o SSL certifikátech a Whois data. Do těchto bloků ještě vstupuje timeout, který je jedním ze vstupních nastavení celého systému. Timeout určuje dobu po kterou bude resolver čekat na odpověď. Pokud do té doby nepřijde je vrácen prázdný objekt a daná data jsou označena za chybějící.

4.3.1 Experimenty se sekvenční architekturou

Pro ověření efektivity zvolené architektury byla provedena řada experimentů. Zajimavým zjištěním je, že pokud jsou data domény dobře dostupná a v žádném resolveru nedojde k aktivaci timeoutu, je doba potřebná pro sekvenční zpracování téměř shodná s paralelním zpracováním. Problém nastává když se systém pokouší získat informace o doméně neexistující nebo nedostupné. V případě sekvenčního zpracování musí postupně v každém resolveru dojít k vypršení onoho nastaveného timeoutu. V tomto případě je paralelní zpracování až třikrát rychlejší.

Kapitola 5

Implementace

V sekci o implementaci je nejprve pozornost věnována použitým knihovnám, zde je vysvětlen jejich konkrétní výběr, stejně tak jako výběr programovacího jazyka. Dále je postupně popsán systém sběru dat, samotná trénovací data a trénování modelů. Na závěr jsou diskutována možná rozšíření systému.

5.1 Využité technologie a knihovny

Využité technologie jsou důležitou součástí implementace. Moderní programovací jazyky a knihovny nabízí vysokoúrovňový přístup ke složitým strukturám jako jsou neuronové sítě nebo jiné klasifikátory se zachováním rychlosti a možnosti akcelerace na specializovaném hardware. Bez toho by poměrně přímočará implementace systému vyhodnocení rizik domén nebyla tak přímočarou.

5.1.1 Python3 a knihovny

Systém je implementován v programovacím jazyce Python3. Tento programovací jazyk byl zvolen především kvůli velkému množství knihoven pro strojové učení a také kvůli jeho architektuře, které umožňuje vysokoúrovňovou práci s daty. Toho je nezbytné pro efektivní implementaci zpracování tak rozsáhlého množství dat jako při klasifikaci domén. V sekci níže pak uvádím výčet a charakteristiku nejvýznamějších knihoven a nástrojů využitých při implementaci. Především se jedná o knihovny pro podporu numerického počítán, strojového učení a databází.

Pytorch

Pytorch[1] je open-source framework pro strojové učení. Je vyvijen laboratoří AI research ve firmě Facebook (nyní již Meta). Vychází pak z knihovny torch. Pytorch poskytuje možnost vysokoúrovňové práce především s neuronovými sítěmi pro strojové vidění. Výhodou této knihovny je to, že umožňuje akceleraci trénování na GPU, které je nezbytná při práci s rozsáhlejšími modely, což je také tento případ. V rámci BP je pytorch využit pro práci s hlavním, na datech založeným modelem.

---

[1] https://pytorch.org/

Sklearn

Sklearn[2] je další knihovnou pro podporu strojového učení pro jazyk python. Oproti výše uvedené knihovně Pytorch, se kterou je často srovnávána. Nabízí podporu širšího spekra metod strojového učení. Z tohoto důvodu je v této práci využita pro tvorbu a práci se SVM modelem. Sklearn využívá celou řadu běžně dostupných knihoven pro numerické počítání a práci s daty, mezi nejznámější patří Numpy, SciPy nebo Pandas.

Tensorflow

Tensorflow[3] je open-source knihovna pro hluboké strojové učení a numerické počítání. Systém je napsán v jazyce C++ a podporuje hardwarovou akceleraci na grafických kartách Nvidia za pomocí speciální platformy NVIDIA CUDA. Tohoto je opět využito pří trénování modelů. Tensorflow se používá ve frameworku Keras, ve kterém je sestaven lexikální model.

5.1.2 MongoDB

MongoDB[4] je NoSQL databáze, která místo tradičních tabulek ukládá dokumenty ve formátu podobném JSON. Tohoto se s výhodou dá využít při implementaci.

Pokud systém pracuje v režimu bez databáze, stále je nutné načítat data, toto je možno provést prostým načtením JSON souborů např s uloženým hodnocením TLD. Jelikož se v prakticky stejném formátu data ukládají i do databáze, při režimu s databází se toto uložiště jen dá dynamicky změnit na onu databázi.

Paralelní zpracování a síť

Na završ uvedu několik knihoven nezbytných pro paralelní zpracování, které se využívá na více místech této BP. Dále také nezbytné síťové knihovny, nutné pro získávání doménových dat. Pro účely paralelizace se využívá standartní knihovna **threading**. Pro účely zpracování DNS záznamů se využívá python implementace DNS resolveru **DNSpython**, obdobně pro whois data se využívá **WhoisPython**

5.2 Sběr dat a tvorba datasetů

Sběr dat je první věcí, kterou systém provede po zadání doménového jmena. V této sekci je popsána jeho implementace a konkrétní akce, které je nutné provést. Dále je pak rozebrána tvorba datasetů, neboť se o něco liší od zpracování dat při samotné klasifikaci.

5.2.1 Systém sběru dat

Funkce a třídy provádějící zpracování dat se nachází v souboru *Data_loader.py*. Pro doménu je vytvořen prázdný záznam do kterého jsou postupně doplňovány informace. Nejdříve je nutné přeložit doménu na IP adresu (nebo adresy), tedy nejprve jsou provedeny dotazy na DNS záznamy. K tomu pak slouží funkce *load_dns_data*.

---

[2] https://scikit-learn.org/stable/

[3] https://www.tensorflow.org/

[4] https://www.mongodb.com/

Pokud není k doméně nalezen platný A záznam, tedy neexistuje k ní IP adresa, systém provede znovu dodaz, již konkrétně na DNS A záznam. Jestli že je tento neúspěšný, sběr DNS dat a dalších dat závislých na IP adresách končí neúspěchem.

Pokud je tedy k doméně nalezena IP adresa, dále je prováděn dotaz na geografické údaje a poskytovatele, který spravuje konkrétní IP adresu. K tomuto je využita služba ipinfo[5]. Tento dotaz a jeho zpracování má na starosti funkce *load_geo_info*.

Dále je proveden dotaz do databáze Whois. Tento dotaz je směřován na doménové jméno, nikoli IP adresu. Je tedy možné například zjistit registrátora domény, která nemá přiřazený DNS A záznam. Vyhodnocení provádí funkce *load_whois_data*.

Posledním krokem je pak sběr dat ohledně zabezpečeného připojení SSL/TSL. Jelikož se jedná o komplexní proces, je mu věnována separátní třída, která se nachází v souboru *SSL_loader.py*. Sběr těchto dat je pak závislý na dostupnosti domény na portu vyhrazeným pro toto připojení, a to 443. Po úspěšném připojení na tomto portu, je stažen SSL certifikát. Z certifikátu se následně extrahují údaje o jeho vydavateli, době expirace a celkové délce platnosti. Certifikáty jsou ukládány do složky *certs* v adresáři *data*.

Po provedení těchto čtyř sad dotazů jsou data z třídy *Base_parse* buď vrácena do modulu *Core* pro okamžité vyhodnocení a nebo uložena do databáze v případě přípavy datasetů.

5.2.2 Trénovací datasety

Ke sběru trénovacích dat jsou využity výše popsané funkce, které slouží k obecnému doplňování dat k doménám. Pro zpracování datasetů jsou pak implementovány dodatečné funkce, které tento proces významě zrychlí. Zpracování trénovacích dat bychom tedy mohli rozdělit do následujících kroků:

1. Načítání a filtrování doménových jmen
2. Paralelní vyhodnocování a načítání dat těchto domén
3. Korekce a filtrování dat
4. Překlad dat do výsledných datasetů

V prvním kroku jsou z předem nastavených zdrojů načítána doménová jména. Zde je rozdíl oproti výsledné klasifikaci, neboť v tom případě je na vstupu pouze jedno doménové jméno zadané uživatelem.

Mezi povolené zdroje pro strojové zpracování patří prosté seznamy domén, odkazy na internetové blacklisty nebo komprimované seznamy domén. Následně jsou odkazy vyčištěny a jsou zahozeny nesmyslná jména jako *localhost*. Při načítání je také nezbytné definovat zda se jedná o maligní či benigní domény. Takto vyčištěná doménová jména jsou ukládána do databáze. Výše uvedený proces je implementován ve funkcích *get_links clean_links* a *get_hostname* ze třídy *Base_parser*.

V dalším kroku jsou postupně z databáze brána tato jména a jsou k nim doplňovány údaje. Při tomto doplňování je pak využita paralelizace, kdy jsou funkce třídy *Base_parse* spouštěny pro více domén zaroveň. Oproti výslednému vyhodnocení, jsou data kontrolována na množství chybějících dat. Pokud u domény schází velké procento dat, systém se je pokusí doplit, pokud to není možné, z důvodu nedostupnosti domény nebo jiných důvodů, systém

---

[5] https://ipinfo.io/

doménu z databáze odstraní. Toto se nebude dít v rámci výsledného vyhodnocení, protože na takové domény bude použit lexikální model. Zde je však trénován datový model.

Statistiky trénovacích dat

Na závěr si uvedeme statistky výsledných trénovacích datasetů. Především pak statistiky týkající se filtrování dat. Nejvíce vyloučených domén bylo mezi maligními doménami, jednalo se o 12,47% vyloučených domén. Mezi doménami benigními bylo vyloučeno 4,23%, tj. přibližně třikrát méně záznamů. V tabulce 2.1 je uvedeno složení výsledných datasetů. Procentuální hodnoty ve sloupcích značí podíl domén, u kterých byly dané charakteristiky úspěšně získány.

	Benigní domény	Maligní domény
Celkový počet	467 744	471 994
% domén s Whois záznamy	99,917%	98,358%
% domén s Geografickými daty	99,654%	99,388%
% domén s DNS daty	100%	100%

Tabulka 5.1: Výsledná data

5.3 Zpracování dat

Zpracování získaných dat je již totožné jak pro trénovací datasety, tak pro výslednou klasifikaci. Data jsou zpracovávána a ukládana ve formátu JSON, vhodném a často využívaném pro sběr dat. Tento formát je však nevhodný pro účely strojového učení a je tedy nutné provést transformaci dat. Rozdílné způsoby zpracování pro konkrétní lexikální i datový model jsou pak popsány níže.

5.3.1 Zpracování pro datový model

Zpracování JSON dat ať už z databáze nebo přímo po sběru dat, zajišťuje třída **Lexical_analysis** v souboru **Lex.py**. Zde jsou postupně ohodnoceny všechny kategorie dat, které byly k doméně sebrány. Hodnoty jsou škálovány do rozpětí -5 až 5.

```json
{
    "label" : 0,
    "domain" : "google.com",
    "data" : [1.105, -0.25, 0, 3, 0, 2.78, 0.4, 1, 2, 1, 0.5, 5, 5]
}
```

Tabulka 5.2: Příklad Ohodnocených a normalizovaných dat

Příklad formátu ohodnocených dat se pak nachází v tabulce 5.2. Položka *label* pak u trénovacích dat značí typ domény.

Doména nejvyššího řádu (TLD)

Doména nejvýšší úrovně, neboli TLD (top level domain) je jedním z klíčových faktorů při posuzování malignosti domény. Registrátoři prestižnějších domén jako .com, .eu, .cz, atd. vyžadují daleko větší množství informací při registraci domén. Je pak snažší spojit konkrétní osobu či organizaci s doménou a tedy i se škodlivým provozem na této doméně.

V první fázi je nejprve ohodnoceno TLD. Toto hodnocení se provádí na základě dat poskytnutých společností Greycortex. Tabulka k tomuto hodnocení se nachází v aresáři *data*, konkrétně v souboru *candidates.domain.csv*. Toto ohodnocení je následně naškálováno na výše uvedený rozsah. Příklad hodnot z tohoto souboru je pak uveden v tabulce 5.3

TLD	Výsledná reputace	Počet benigních domén	Počet maligních domén
.com	77,9	65332	19821
.net	82,5	5478	875
.eu	62,3	10341	4588
.soft32	0,4	1	559

Tabulka 5.3: Hodnocení TLD

Hodnocení DNS záznamů

Vyhodnocování DNS záznamů je ve většině studovaných systémech klasifikace opomijeno. Ačkoli S.Kogo a A.Kanai.[10] prezentují úspěšně detekují v DNS provozu škodlivý provoz nejedná se o použití v systému přímo pro klasifikaci domén. Analýza DNS záznamů představuje jednu z oblastí, které by stálo za to věnovat separátní práci a může být předmětem rozšíření této BP jakožto i dalšího studia. V tomto systému budou DNS záznamy pro jednoduchost klasifikovány binárně, tedy zda jsou přítomné nebo ne. Ohodnocení přítomnosti či nepřítomnosti je pak jiné podle typu záznamu. Například přítomnost MX záznamu u domény je dnes jiř spíše raritou než pravidlem a tak není jeho nepřítomnost penalizována. Ačkoli takovýto způsob hodnocení prakticky zahazuje obsah DNS záznamů, pro implementaci systému plně dostačuje.

Hodnocení SSL dat

Hodnocení SSL dat se rozpadá to několika kritérií, první je samotná přítomnost či nepřítomnost možnosti zabezpečeného připojení přes SSL. V dnešní době je již nezvyklé tento typ zabezpečení neposkytovat, tedy se jedná o velké mínus pro klasifikovanou doménu.

Pokud takovéto připojení je k dispozici je hodnocen nejprve vydavatel SSL certifikátu. Vydavatelům jako google, banky atd. Je přiřazeno lepší hodnocení a naopak poskytovatelem zdarma certifikátů jako letsencrypt je udělováno horší hodnoení.

S tímto úzce souvisí délka platnosti certifikátu. Věrohodnější certifikační autority poskytují delší platnost než ty zdarma.

Ohodnocení	Maximální délka platnosti
A, +5	300 dnů
B, +3	80 dnů
C, +0.5	30 dnů

Tabulka 5.4: Porovnání chyb

Pro účely hodnocení těchto dat byla platnost certifikátů rzdělena do 3 kategorií, kdy každé kategorii připadá příslušné skóre. Toto hodnocení je pak v tabulce 5.4.

Geografické a jiné údaje

Lokalita je ukládána ve formě geografických souřadnic. Tedy obsahují zeměpisnou šířku v rozsahu 0-90 °a zeměpisnou délku v rozsahu 0-180°. Tyto údaje jsou opět naškálovány na požadovaný rozsah. Mezi další zpracovávané charakteristiky, pak také patří **počet teček** v názvu domény a **celková délka názvu**. Ačkoli bychom tyto parametry spéíše řadily do sekce o lexikální analýze, jsou však natolik důležitým ukazatelem při klasifikaci, že je dobré je datovému modelu expliticně dát.

5.3.2 Zpracování pro lexikální model

Lexikální zpracování pak zajišťuje třída *Preprocessor*. Jelikož není tato analýza závislá na jiných datech, je možné ji provádět paralelně se sběrem dat.

Jedná se o podstatnou část doménové klasifikace, ze které je možné vyvodit široké spektrum závěrů s významnými dopady. Velkého významu pak nabývá pokud se systému nepodaří připojit se k dané doméně a tedy hlavní model nemá k dispozici téměř žádné údaje. V takovém případě nezbývá nic jiného než se spolehnout na lexikální analýzu a její výsledky.

Název je tedy nejprve převeden na pouze malá písmena a všechny číslice jsou nahrazena symbolem otazníku. Dále je název segmentován na bigramy, účel této segmentace je popsán v sekci o návrhu lexikálního modelu. Seznam bigramů obsahuje celkem 896 položek a byl opět poskytnut firmou Greycortex. Tyto bigramy jsou následně nahrazeny pořadím, které jim náleží v daném seznamu. Takto zakódovaný název domény je následně vstupem modelu.

Příklad segmentace doménového jména **google.com** se pak nachází v tabulce 5.5.

```
{
    "bigrams" : ['go', 'oo', 'og', 'gl', 'le', 'e.', '.c', 'co', 'om']
}
```

Tabulka 5.5: Příklad segmentace na bigramy

DGA algoritmy

Jedním z hlavních cílů této analýzy je identifikace domén generovaných algoritmy DGA. DGA neboli Domain Generation Algorithms slouží k automatickému generování jmen. Je běžnou praxí, že škodlivé programy využívají domény jako centrální body komunikace a tedy je nutné, aby byly domény registrovány automaticky, k tomuto účely DGA algoritmy. Tento vzor můžeme pozorovat například při řízení botnetů, kdy např. škodlivý program může odmítnout příkaz či update, který nepochází z konkrétní domény, platné pro daný časový úsek. Příklady takto generovaných domén mohou být: **intgmxdeadnxuyla.com** nebo **axwscwsslmiagfah.com**. Zde se využije segmentace na bigramy, jelikož takto generované domény zřídkakdy obsahují smysluplná slova.

5.4 Modely a jejich trénování

Skripty nutné k trénování všech tří modelů se pak nachází v adresáři *training*. Je nutné mít také nastavené připojení do databáze, kde se nachází kolekce s trénovacími daty. Samotné vytrénované modely se pak nacházení v adresáři *models*. Je možné do systému vložit vlastní modely strojového učení, jejich cestu je pak nutné nastavit v souboru .env.

Parametry datového modelu

Za zmínku také stojí volby parametrů pro učení datového modelu. Vzhledem k tomu že se implementuje binární klasifikace byla zvolena ztrátová funkce *Loss function Binary Cross Entropy*. Rychlost učení, neboli koeficient lr, byl experimentálně stanoven na 0.001. Celkový počet trénovacích epoch byl pak 25.

5.5 Systém vyhodnocení modelů

Posledním blokem systému je vyhodnocení výstupů modelů a metadat do výsledné predikce. Při implementaci tohoto bloku musíme zohlednit chování jednotlivých modelů při určité ztrátě dat. Nejprve je tedy třeba provést sadu experimentů při kterých do jednotlivých modelů dodáme pouze určité procento dat. Hlavním účelem tohoto bloku je aby systém při nedostatku dat, dal přednost vhodnému modelu, který pro dané množství dat bude vykazovat největší přesnost.

Logicky je možné se domívat že lexikální model jehož vstupem je pouze doménové jméno nebude závislý na datech. Tedy systém tento model upřednostní v případě nemožnost získat téměř žádná data. Experimentem je ale třeba zjistit kde se tato hranice nalézá. Hranice byla nalezena okolo 50% chybějících dat. Této problematice je věnována sekce v poslední kapitole, které se zabývá testováním lexikálního modelu a dala by se shrnout následovně:

- Při klesajícím množství dostupných dat, má hlavní model tendenci klasifikovat domény jako škodlivé.
- Výstup lexikálního modelu není závislý na množství dat. Jeho přesnost a výstup závisí pouze na jménu domény
- Lexikální model má tendenci klasifikovat dlouhé názvy jako maligní.

Ačkoli se chování dle prvního bodu může zdát korektní, je nutné v takovém případě hlavní model usměrňovat lexikálním a SVM modelem. Další metadata, která mohou být v bloku konečného vyhodnocení užitečná je délka domény a řád subdomény. Toto všechno bude sloužit ke korekci lexikálního modelu

5.6 Rozšíření

Možná rozšíření systému se odvíjí od směru kterým by se případný vývoj ubíral. Pokud by šlo o zpřesnění klasifikace jako takové, ve hře by byla reorganizace či přidání modelů. Na druhou stranu, jestliže bychom hovořili o vylepšování aplikace z uživatelského hlediska. Zde by se dalo hovořit například o vylepšování odezvy, či nejen binární klasifikaci.

5.6.1 Více modelů

Naprosto intuitivním rozšířením které se nabízí, je vytrénovat více modelů. Zajimavá architektura by jistě byla nahradit hlavní klasifikační model několika více specializovanými.

Obecně můžeme říct modely specializované na provádění konkrétní činnosti, vykazují lepší přesnost než modely obecné. Na tomto základě by pak bylo možné sestavit konkrétní modely pro hodnocení DNS záznamů, geografických dat nebo Whois dat.

Další rozdrobování těchto modelů na dílčí modely by pak dle mého názoru bylo neefektivní. Zvyšovat počet lexikálních modelů se zdá zbytečné. Název domény je velmi malý zlomek všech údajů, které je možné dostat.

Model pro hodnocení DNS

Kategorie dat, pro kterou by stálo za úvahu realizovat separátní klasifikační model jsou DNS záznamy. Rozšířil by se tím nyní používaný binární systém, tedy hodnocení by mohlo obsahovat více než hodnoty existuje a neexistuje. Bylo by tak možné analyzovat například obsahy TXT záznamů, které se mohou používat při ověřování vlastnictví domény. Analýza MX, CNAME nebo SOA záznamů by také jistě přinesla zajimavá zjištění.

Zajimavé poznaky z analýzy DNS dat pomocí strojového učení, prezentují ve své práci S.Kogo a A.Kanai [10]. Jejich přístup zahrnuje agregaci ještě většího počtu údajů, jako jsou délky DNS paketů, intervaly mezi pakety a nebo návratové kódy. Tyto údaje jsou následně analyzovány pomocí algoritmu strojového učení "Random Forest", což je kombinovaná metoda pro klasifikaci a regresi. Tento algoritmus pak vychází z rozhodovacích stromů.

5.6.2 Nejen binární klasifikace

Dalším vylepšením, které by bylo možné implementovat je rozšíření klasifikace z binární podoby na více kategorií. Nabízí se například rozdělění na phishingové domény, domény obsahující malware nebo "evil twin"domény. Tento přístup prezentuje F.Strašák [6]. Ačkoli takovýto přístup značně komplikuje tvorbu datasetů a vyžaduje rámcově mnohem více dat, myslím že by určitě stál za úvahu. Může ovšem nastat situce že pro učení neuronových sítí nebude dostupné dostatečném množství dat. Pak by musela být zvolena jiné metoda strojového učení. Dobrým kandidátem se může jevit SVM, které je zde již využito v podpůrné roli. Za zkoušku by stály i algoritmy založené na principech rozhodovacích stromů jako je například "Random Forest".

5.6.3 Databáze domén

Efektivním způsobem jak snížit průměrnou odezvu od zadání vstupní domény uživatelem je implementace cache systému.

Systém by mohl uchovávat několik miliónů již ohodnocených domén. Zde nastává problém že data domén se mohou rychle měnit. Toto by šlo řešit přidáním expirace jednotlivých záznamů. Tedy pokud by bylo ohodnocení domény starší než určitá meze, systém by data automaticky obnovil. To by se dalo provádět ve chvílích nižšího uživatelského vytížení. Takový přístup by se obvzlášť využil při implementaci webových systémů.

Kapitola 6

Testování na reálných datech

Tato kapitola popisuje testování, které bylo se systémem provedeno. To je možné rozdělit do tří fází. První fází bylo samostatné testování modelů během jejich vývoje na připravené validační sadě dat. Ve druhé fázi byl systém testován jako celek na stejné sadě validačních dat. Třetí fází bylo uživatelské testování pomocí webové implementace, na menším množství dat. Výsledky ukazují dobrou přesnost využitých klasifikačních metod a úspěšnost při odhalování maligních domén. Testovací data použitá pro prvotní testování jednotlivých modelů a testování celého systému byla vzata z hlavního datasetu, konkrétně se jedná o 20% tohoto datasetu. Základní metrikou pro měření úspěšnosti klasifikace je **F1 skóre**, které je definováno rovnicí na obrázku 6.1.

$$\mathbf{F1} = \frac{2 * TP}{2 * TP + FP + FN}$$

Obrázek 6.1: Vzorec pro výpočet F1 skóre

Dále jsou uváděny grafy pro validační chybu, **validation loss**, na testovacích datasetech. Při hodnocení celého systému je také zmíněna přesnost **accuracy**, jejíž vzorec je následující:

$$\mathbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Obrázek 6.2: Vzorec pro výpočet Accuracy

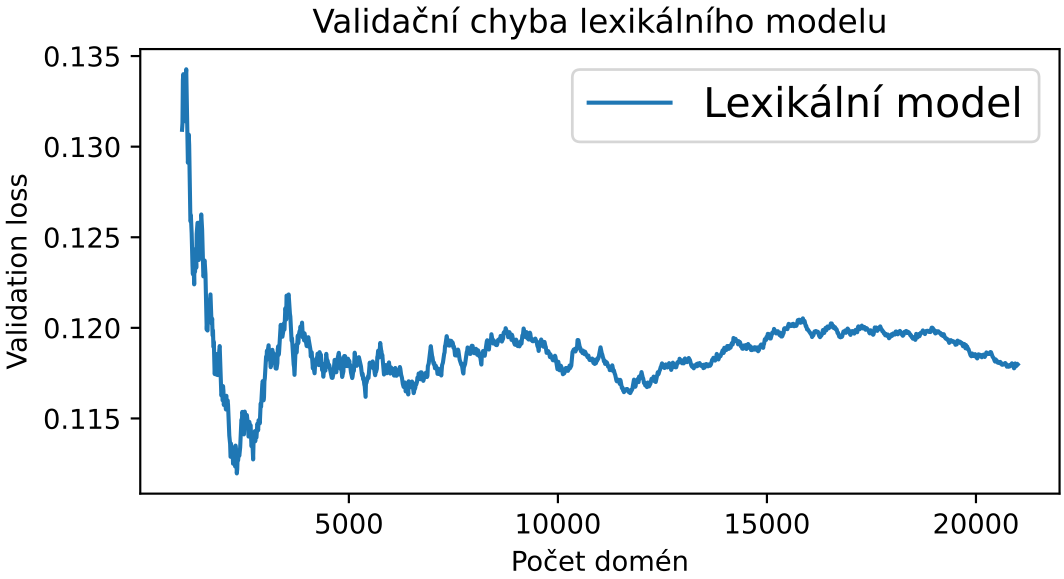
TP neboli **True positive** značí správně klasifikované pozitivní výsledky. Obdobně **TN**, **True negative** správně klasifikované negativní výsledky. **FN** a **FP** jsou nesprávně klasifikované negativní a pozitivní výsledky.

6.1 Testování jednotlivých modelů

Testování na 20% dat z trénovacího datasetu během vývoje bylo základní zpětnou vazbou pro úpravu architektury modelů.

6.1.1 Lexikální model

Výsledek testování je znázorněn na grafu 6.3. **Skóre F1** pro tento model je **0.882**. Ačkoli se tato naměřená hodnota může zdát nízká, je třeba si uvědomit, že model rozhoduje o malignosti domény pouze na základě jejího názvu a nevyžaduje tedy žádná dodatečná data.

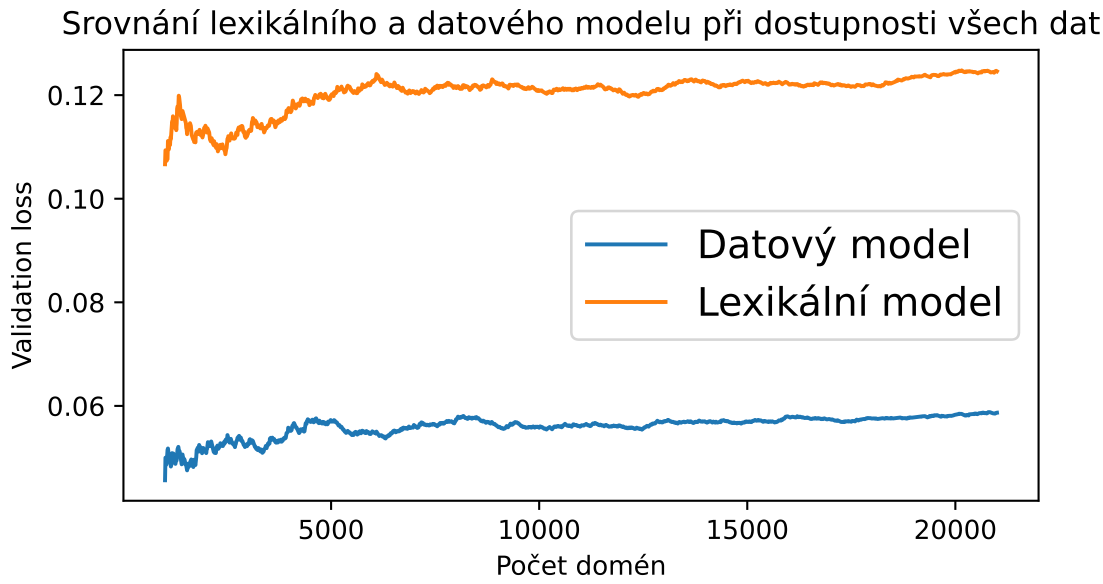


Obrázek 6.3: Chyba lexikálního modelu na testovací sadě

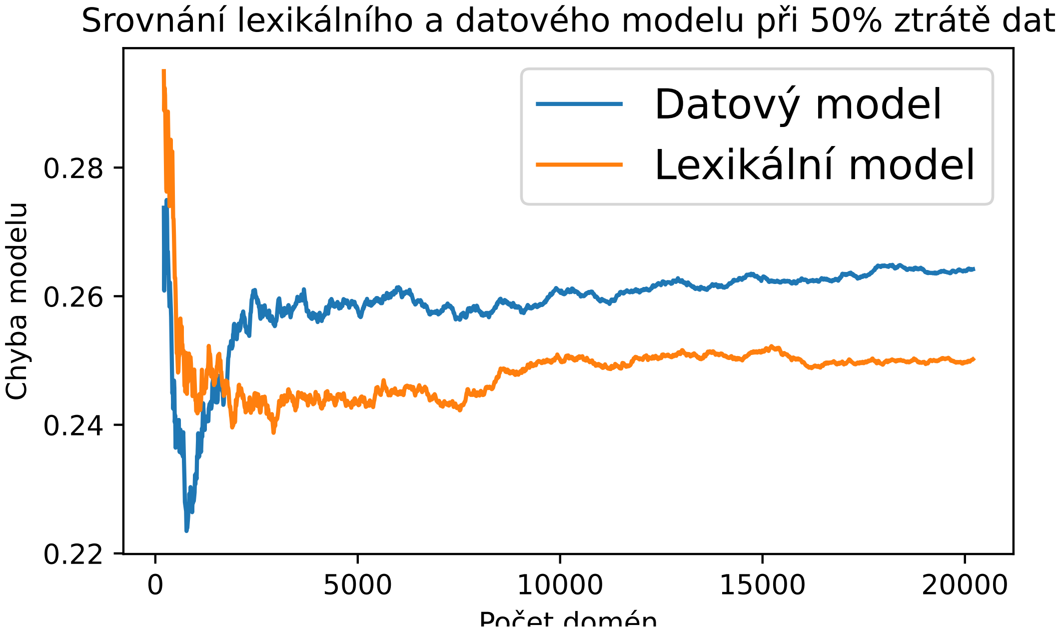
Existují specifické situace, kdy je přesnost lexikálního modelu srovnatelná nebo dokonce lepší, než přesnost hlavního datového modelu. Prvním případem je situace, kdy datový model nemá k dispozici dostatek dat pro spolehlivé vyhodnocení. Tento jev byl zmiňován při návrhu modulu pro vyhodnocování modelů v kapitole 5. Na grafu 6.4 je znázorněno srovnání chybovosti obou modelů, pokud má datový model k dispozici kompletní data. Situace se mění v grafu 6.5, kdy datovému modelu byla odebrána polovina dat. Toto simuluje situaci, kdy se nepodaří získat například žádná SSL data a část Whois dat. V takovém případě je jasně vidět, že přesnost lexikálního modelu je lepší.

Druhým případem může být klasifikace domén s dlouhými názvy. Toto je demonstrováno na grafu 6.6, kde byly klasifikovány domény s délkou větší než 10 znaků. I pro tuto délku je přesnost lexikálního modelu znatelně větší. Kdybychom hranici posunuli na 12 znaků, byl by tento rozdíl ještě výraznější. Závěry z testování lexikálního modelu je možné shrnout takto:

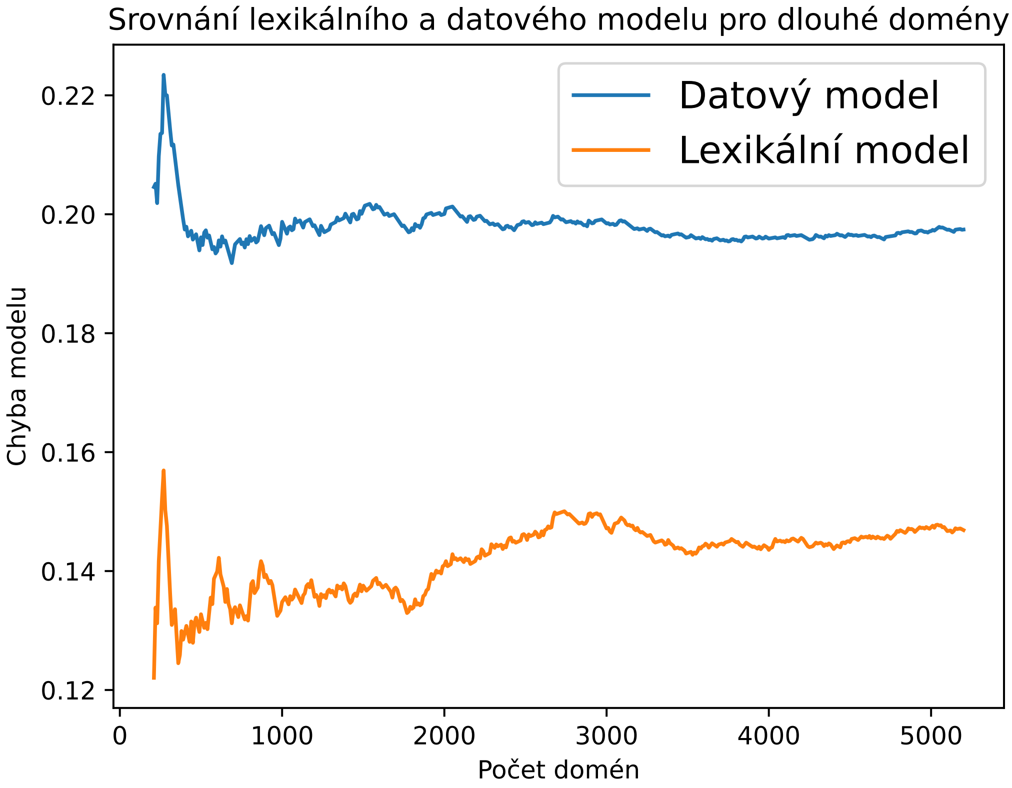
- Lexikální model vykazuje větší přesnost pro delší názvy domén (l>10).
- Přítomnost čísel či speciálních znaků vede, až na vyjímky, ke klasifikaci domény jako maligní.
- Model klasifikuje domény jako špatné, pokud se v jejich názvu vyskytují zkratky nebo náhodné shluky písmen, které netvoří slova.



Obrázek 6.4: Srovnání lexikálního a datového modelu pro kompletní data



Obrázek 6.5: Srovnání lexikálního a datového modelu pro 50% ztrátu dat



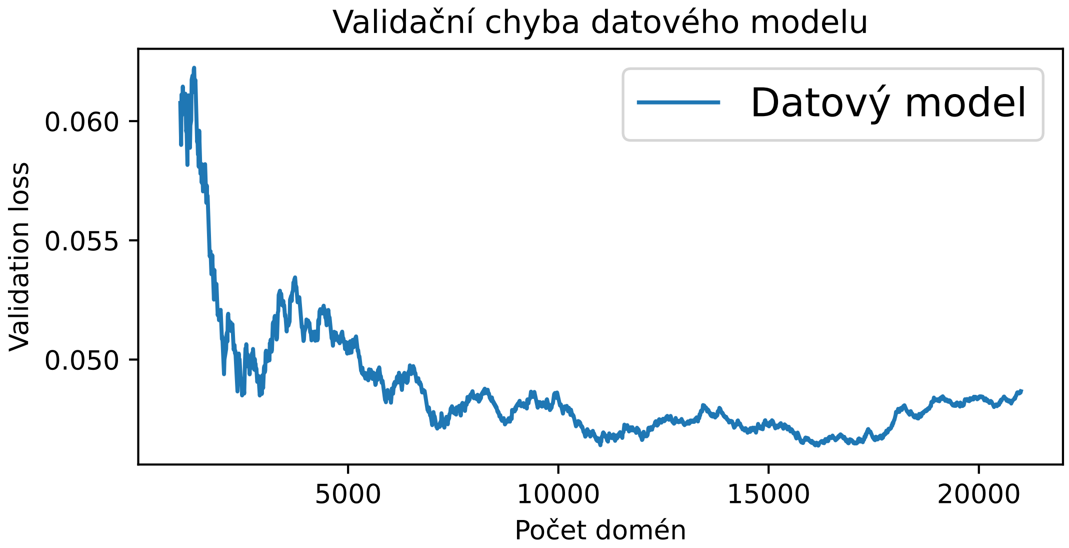
Obrázek 6.6: Srovnání lexikálního a datového modelu pro dlouhé domény ≥ 10

6.1.2 Hlavní model

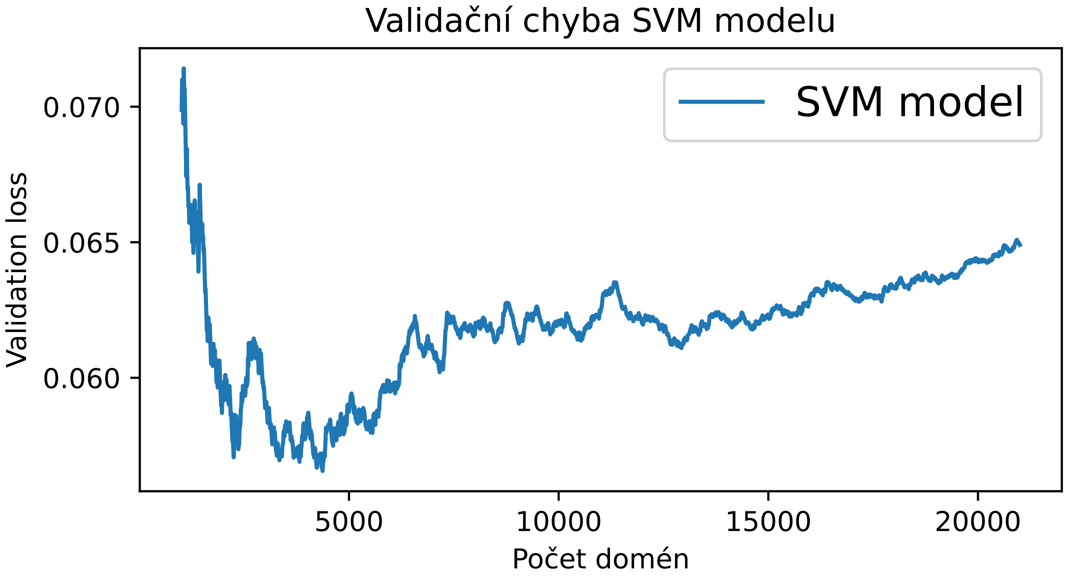
Na datech založený model, zde označovaný jako hlavní či datový model, je hlavním klasifikátorem. Na grafu 6.7 je zobrazena jeho validační ztráta (validation loss). **F1 skóre** tohoto modelu je **0.946**. Pro klasifikační model je to velmi slušná hodnota.

6.1.3 SVM model

Posledním modelem je model založený na metodě **Support Vector Machines**. Trénování modelu bylo prováděno na stejných datech jako u hlavního modelu. **F1 skóre** tohoto modelu je **0.934**. Jedná se o podstatně lepší přesnost než u lexikálního modelu a mírně horší než u datového modelu. Hlavním rozdílem oproti datovému modelu je vyšší odolnost na ztrátu dat. Ačkoli stále pracujeme s datovým modelem, tato odolnost není tak markantní, jako u lexikálního modelu. Větší přesnost SVM modelu se projevuje mezi 20-30% ztracených dat. Pro vyšší hodnoty ztrát dat jeho přesnost již klesá stejně jako u datového modelu. Pro hodnotu 30%, která se zdá být hraniční, je uvedeno porovnání těchto dvou modelů na grafu 6.9



Obrázek 6.7: Validační chyba datového modelu

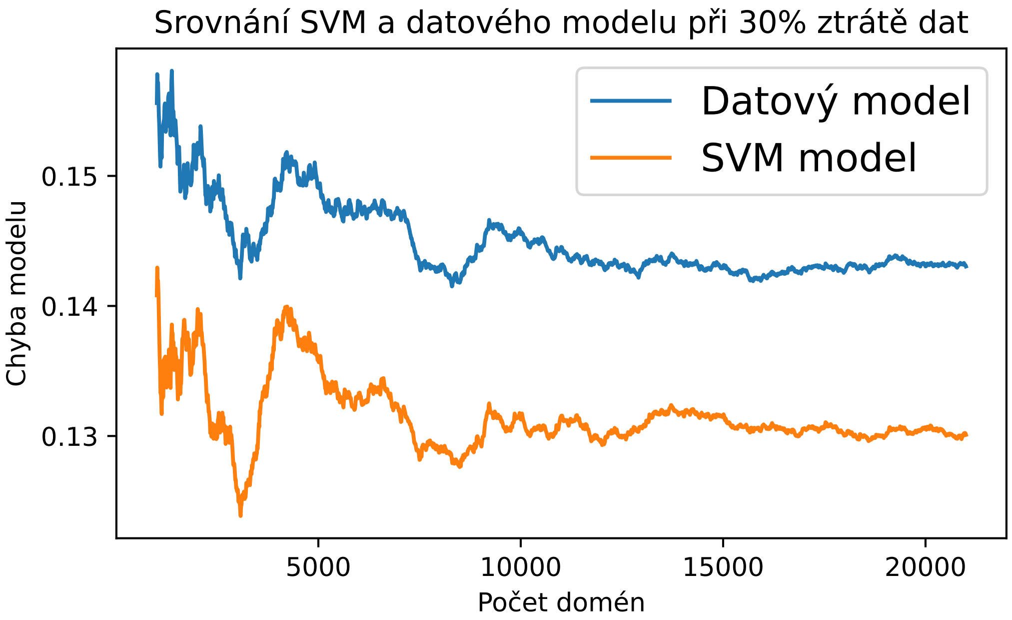


Obrázek 6.8: Validační chyba SVM modelu

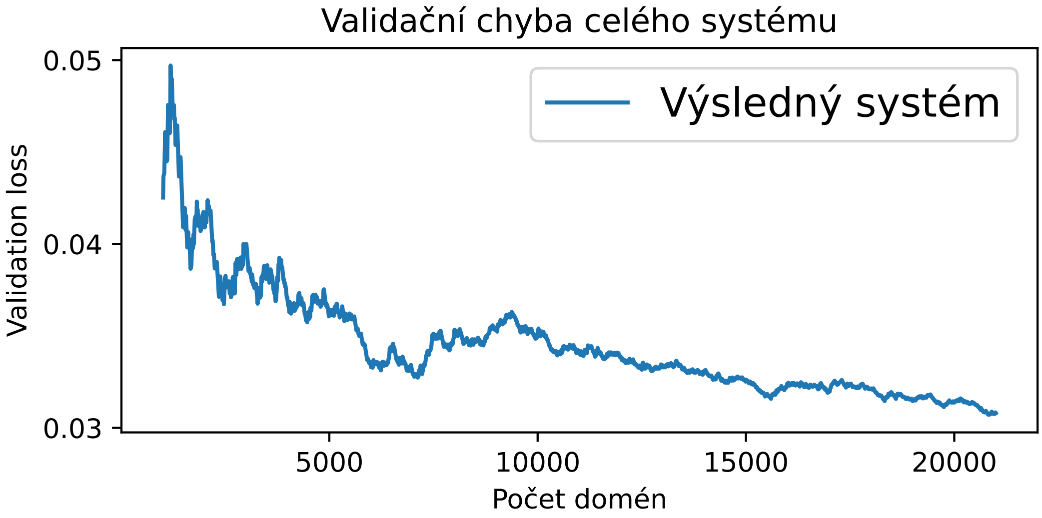
6.2 Testování celého systému

Cílem testování celého systému je ověřit správnou funkčnost modulu agregujícího výstupy dílčích modulů. Tedy ověřit zamýšlenou funkcionalitu, že při nedostatku dat by měl systém dát přednost lexikálnímu, případně SVM modelu, které jsou v případě ztráty dat přesnější.

Pro testování celého systému je znázorněna kromě validation loss (obrázek 6.10) také ROC křivka (obrázek 6.11). **F1 skóre** celého systému je **0.971**, přesnost **96,3%**.

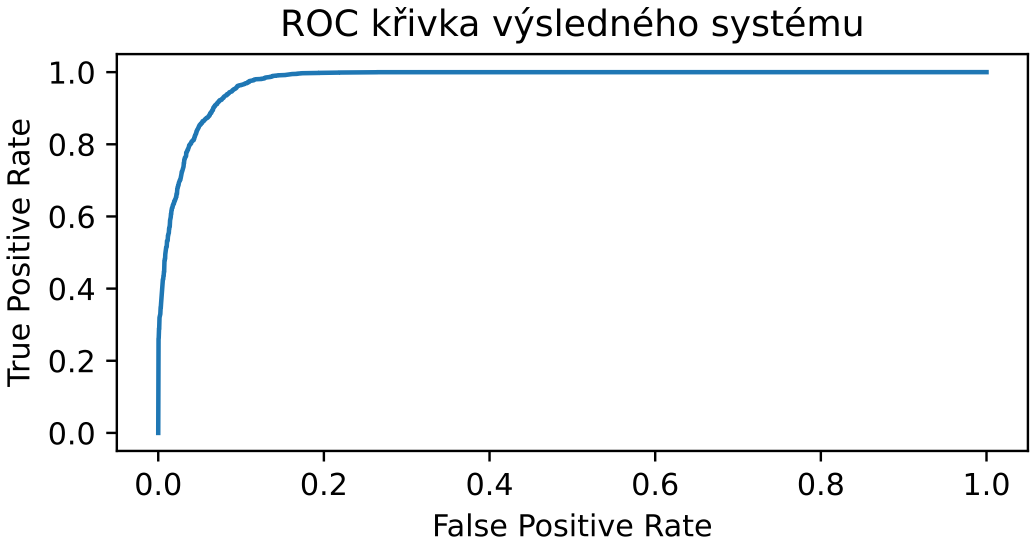


Obrázek 6.9: Srovnání SVM modelu a datového modelu při nedostupnosti 30% dat



Obrázek 6.10: Validační chyba celého systému

Z toho je zřejmé, že systém agregace více modelů opravdu zpřesnil klasifikaci celého systému nad úroveň všech použitých modelů. Konkrétní tabulku s procentuálním rozložením typů chyb je možné nalézt v sekci 6.2.2.



Obrázek 6.11: ROC křivka celého systému

6.2.1 Testování pomocí webové implementace

Testování pomocí webové implementace probíhalo od 10.4.2022 do 18.4.2022. Cílem testů s reálnými uživateli bylo jak změřit průměrnou odezvu systému, tak zejména správnost klasifikace. Jednalo se spíše o doplňkové testování vzhledem k menšímu dostupnému množství dat. Po zobrazení predikce byli uživatelé vyzváni k odeslání zpětné vazby, zda je poskytnutá predikce korektní. Na základě této zpětné vazby bylo prováděno hodnocení.

Jedním z měřených údajů při testování webové implementace byla odezva systému. Tyto naměřené údaje mohou být dobrou zpětnou vazbou pro větší paralelizaci systému. Uživatelé v tomto období do systému zadali celkem 244 domén. Data z tohoto testování jsou shrnuta v tabulce 6.1

Typ domény	počet	průměrná odezva	úspěšnost
Maligní	39	9,56 s	97,4%
Benigní	205	6,98 s	92,2%
Celkem	244	7,39 s	93,0%

Tabulka 6.1: Údaje z uživatelského testování

Dobrým zjištěním je nízká míra falešně negativních výsledků, neboť jsou nebezpečnější. Zajimavým poznatkem může být nepoměr mezi zadanými benigními a maligními stránkami. Z tohoto nepoměru může plynou nepoměr v úspěšnosti u jednotlivých kategorií. Další měřenou veličinou je odezva. I tyto naměřené hodnoty odpovídají intuitivním předpokladům. U maligních domén je daleko častější jejich nedostupnost, je zde tedy pravděpodobnější, že systém narazí na časový limit pro resolvery.

6.2.2 Typy nesprávné klasifikace a jejich eliminace

Na závěr sekce o testování je vhodné uvést poznámku o typech nesprávných výsledků při klasifikaci a zejména o způsobu přístupu k nim při klasifikaci domén.

Nesprávné výsledky můžeme rozdělit na falešně pozitivní a falešně negativní. Tím prvním případem by při klasifikaci domén bylo ohodnocení neškodného doménového jména jako škodlivého. Druhým pak ohodnocení maligní domény jako neškodné. Z principu je druhý typ chyby mnohem závažnější. Při vývoji byl tedy kladen důraz zejména na eliminaci falešně negativních výsledků. V tabulce 6.2 jsou uvedeny statistiky těchto nesprávných typů klasifikace.

Zdroj dat	True positive	False positive	True negative	False negative
Testovací dataset	32.72%	2,31%	63,58%	1,44%
Reálný provoz	13.38%	2,58%	83.63%	0,41%

Tabulka 6.2: Porovnání typů chyb výsledného systému

Kapitola 7

Závěr

Cílem této práce byl návrh a implementace systému pro vyhodnocení rizik doménových jmen. Tohoto cíle se povedlo dosáhnout. Práce obsahuje souhrn současných řešení při klasifikaci maligních domén. Dále pak návrh a implementaci vlastního systému využívajícího strojové učení. Tento systém je implementován za použití dvou neuronových sítí a klasifikační metody support vector machines. Díky této kombinaci přístupů systém dokáže detekovat maligní domény i s malým počtem dostupných dat. Dále bylo provedeno rozsáhlé testování systému, jehož závěry je možné nalézt v poslední kapitole. Nad rámec zadání Bakalářské práce se podařilo vytvořit webovou implementaci systému **urlcheck**[1]. Tato implementace umožnila testování a měření na datech od reálných uživatelů.

Klasifikace domén a jejich škodlivosti je považována za náročný úkol. Techniky, které využívají tvůrci škodlivých stránek, se rychle mění a není proto jednoduché všechny jejich rysy odhalit. I přes tyto náročné podmínky vykazuje uvedený implementovaný systém na testovacích datech celkově 96,3% úspěšnost, což je obecně považováno za dobrý klasifikační model. V práci jsou také diskutována možná rozšíření systému, která by mohla klasifikaci ještě zpřesnit. Tato rozšíření se nachází v kapitole 5 Další potenciál vidím v implementování uvedených rozšíření, ať už za účelem zpřesnění klasifikace nebo zlepšení uživatelského zážitku.

Při implementaci systému a zpracování BP jsem úzce spolupracovat s Ing. P.Chmelařem z firmy Greycortex, kterému bych chtěl tímto ještě jednou poděkovat. Tato spolupráce mi umožnila nahlédnout do praktického světa internetové bezpečnosti a motivovala mě do dalšího studia. Při zpracování práce jsem se také seznámil s principy strojového učení, zejména pak s principy návrhu a implementace neuronových sítí. Téma internetové bezpečnosti je v dnešní době více než aktuální a plánuji se mu i na dále věnovat.

---

[1] `https://urlcheck.eu`

Literatura

[1] ALPAYDIN, E. *Introduction to Machine Learning.* 2nd. The MIT Press, únor 2010. ISBN 026201243X.

[2] AMAIZU, G. C., AGRON, D. J. S., LEE, J.-M. a KIM, D.-S. Two-Stage Classification Technique for Malicious DNS Identification. In: *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC).* 2021, s. 068–072. DOI: 10.1109/ICAIIC51459.2021.9415225.

[3] BAKAR SIDDIK, M. A., MAZUMDER, M. M. I., ALAM, R. a KHAN, M. Performance Comparison Between Dimension Reduction and Feature Selection Approaches for Data Classification. In: *2021 5th International Conference on Computing Methodologies and Communication (ICCMC).* 2021, s. 893–898. DOI: 10.1109/ICCMC51019.2021.9418293.

[4] BIANZINO, A. P., PEZZUOLO, D. a MAZZINI, G. Who is whois? An analysis of results consistence. In: *2014 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCOM).* 2014, s. 289–292. DOI: 10.1109/SOFTCOM.2014.7039137.

[5] ELSHERIF, H. a HAMBABA, M. A modular neural network architecture for pattern classification. In: *Neural Networks for Signal Processing III - Proceedings of the 1993 IEEE-SP Workshop.* 1993, s. 232–238. DOI: 10.1109/NNSP.1993.471865.

[6] FRANTIŠEK, S. *Should I click on a link? Machine Learning to Protect from Cyber Attacks on the Web.* 2019. [cit. 2022-03-15]. 89 s. Diplomová práce. České vysoké učení technické v Praze, Fakulta elektrotechnická. Vedoucí práce GARCÍA, S.

[7] GOEL, A. a SRIVASTAVA, S. K. Role of Kernel Parameters in Performance Evaluation of SVM. In: *2016 Second International Conference on Computational Intelligence Communication Technology(CICT).* 2016, s. 166–169. DOI: 10.1109/CICT.2016.40.

[8] GUPTA, A., CHETTY, N. a SHUKLA, S. A classification method to classify high dimensional data. In: *2015 International Conference on Computing, Communication and Security (ICCCS).* 2015, s. 1–6. DOI: 10.1109/CCCS.2015.7374132.

[9] JINGYANG, G., QUNXIONG, Z. a CHENGLIZHAO, C. A new method of evaluation and optimization for feature extraction. In: *2010 International Conference on Computer and Information Application.* 2010, s. 40–43. DOI: 10.1109/ICCIA.2010.6141532.

[10] KOGO, S. a KANAI, A. Detection of Malicious Communication Using DNS Traffic Small Features. In: *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE).* 2019, s. 125–126. DOI: 10.1109/GCCE46687.2019.9015292.

[11] Kurose, J. *Computer networking : a top-down approach.* Harlow, England Boston: Pearson, 2017. ISBN 978-1-292-15359-9.

[12] Ma, J., Saul, L., Savage, S. a Voelker, G. Beyond blacklists: learning to detect malicious Web sites from suspicious URLs. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009.* Leden 2009, s. 1245–1254. DOI: 10.1145/1557019.1557153.

[13] Mcgrath, D. K. a Gupta, M. Behind Phishing: An Examination of Phisher Modi Operandi. In: *In Proceedings of the USENIX Workshop on Large-scale Exploits and Emergent Threats.* Leden 2008.

[14] Nocera, P. a Quelavoine, R. Diminishing the number of nodes in multi-layered neural networks. In: *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94).* 1994, sv. 7, s. 4421–4424 vol.7. DOI: 10.1109/ICNN.1994.374981.

[15] Pope, M., Warkentin, M., Mutchler, L. a Luo, R. The Domain Name System: Past, Present, and Future. *Communications of the Association for Information Systems.* Květen 2012, sv. 30, s. 329–346. DOI: 10.17705/1CAIS.03021.

[16] Selamat, M. H. a Md Rais, H. Image face recognition using Hybrid Multiclass SVM (HM-SVM). In: *2015 International Conference on Computer, Control, Informatics and its Applications (IC3INA).* 2015, s. 159–164. DOI: 10.1109/IC3INA.2015.7377765.

[17] Stanevski, N., Tsvetkov, D. a CLASSIFIER, M. Using Support Vector Machine as a Binary Classifier. In: *International Conference on Computer Systems and Technologies –CompSysTech.* Leden 2005.

[18] Sun, B., Song, S.-J. a Wu, C. A new algorithm of support vector machine based on weighted feature. In: *2009 International Conference on Machine Learning and Cybernetics.* 2009, sv. 3, s. 1616–1620. DOI: 10.1109/ICMLC.2009.5212256.

[19] Tan, C.-M., Wang, Y. a Lee, C.-D. The Use of BiGrams to Enhance Text Categorization. *Information Processing & Management.* Červenec 2002, sv. 38, s. 529–546. DOI: 10.1016/S0306-4573(01)00045-0.

[20] Wang, W. a Shirley, K. *Breaking Bad: Detecting malicious domains using word segmentation.* arXiv, 2015. DOI: 10.48550/ARXIV.1506.04111.

[21] Yumeng, C. a Yinglan, F. Research on PCA Data Dimension Reduction Algorithm Based on Entropy Weight Method. In: *2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI).* 2020, s. 392–396. DOI: 10.1109/MLBDBI51377.2020.00084.

[22] Zhang, T., Zhang, H. a Gao, F. A Malicious Advertising Detection Scheme Based on the Depth of URL Strategy. In: *2013 Sixth International Symposium on Computational Intelligence and Design.* 2013, sv. 2, s. 57–60. DOI: 10.1109/ISCID.2013.128.

Příloha A

Obsah přiloženého paměťového média

Přiložený disk obsahuje:

- tento dokument ve formátu PDF,
- zdrojové soubory tohoto dokumentu,
- zdrojové kódy vyvinutého klasifikačního systému i webové implementace,
- testovací data,

Zdrojový kód systému je dostupný na adrese `https://github.com/poli-cz/domain_evaluation`, součástí zdrojových kódu je i **podrobnější návod na instalaci softwaru**. Tato implementace je pak ve formě webové aplikace přístupná na adrese `https://urlcheck.eu/`

Příloha B

Manuál

Tento manuál podrobněji popisuje zprovoznění aplikace a odevzdané zdrojové kódy. Je také k dispozici v online podobě na adrese `https://github.com/poli-cz/domain_evaluation`. Repozitář obsahuje kromě implementace samotného systému vyhodnocení rizik doménových jmen také implementaci testovací webové aplikace. Dále jsou přiložena data, na kterých je možné systém testovat.

B.1 Adresáře

Základní popis adresářů s implementací v odevzdaném archivu je pak následující.

- **/src** - Samotná implementace systému.
- **/src_training** - Skripty určené k trénování modelů.
- **/web_app** - Zdrojové kódy demonstrační webové aplikace.
- **/web_app/sites** - Cache webové aplikace, ukládají se zde již vyhodnocené domény.
- **/mongo** - Testovací data ve formě zálohy databáze MongoDB.
- **/models** - Samotné vytrénované klasifikační modely.

B.2 Databáze

Testovací data jsou dostupná v adresáři **/mongo** ve formě zálohy databáze **MongoDB**. Záloha obsahuje databázi *domains* s následujícími kolekcemi:

- **allDomains** - Kolekce obsahující pouze jména domén, jak maligních tak benigních.
- **badDomains** - Data maligních domén, bez chybějících záznamů.
- **goodDomains** - Data benigních domén, opět vyfiltrováno.
- **badDataset** - Přeložená data maligních domén, ve formě vektoru.
- **goodDataset** - Přeložená data benigních domén, opět ve formě vektoru.

K vytvoření této zálohy byl využit nástroj **mongodump**, opětovné nahrání zálohy je možné nástrojem **mongorestore**.

B.3 Instalace

Systém je programován v jazyce python3, konkrétně je vyžadována verze **python 3.8.10** nebo vyšší. Webová implementace pak vyžaduje **nodejs** ve verzi **14.18** nebo vyšší. Konkrétní kroky, které jsou třeba provést před samotným spuštěním jsou následující:

1. Instalace balíčků pro jazyk python ze souboru **requirements.txt**.
2. Instalace balíčků pro jazyk javascript ze souboru **package.json**.
3. Nastavení hodnot proměnného prostředí (databáze) v souboru **.env**.
4. Inicializace webové aplikace pomocí skriptu **init_web_app.sh**

B.4 Použití

Systém může být spouštěn jako samostatná aplikace. K tomuto slouží soubor **init.py**, který nabízí jednoduchou abstrakci nad moduly celého systému. Tento soubor pak podporuje následující parametry:

- **–silent** - Bez výstupu na STDOUT, výstup probíhá do souboru ve <domain.tld.>json.
- **–stdout** - Výsledek je vytisknut na STDOUT.
- **–lexical** - Provede pouze lexikální analýzu domény.
- **–data** - Pouze datová analýza domény.
- **–svm** - Analýza pouze za použití SVM modelu.

Ukázka tohoto spuštění aplikace se nachází ve skriptu **backend_usage_example.sh**