
**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**
**BRNO UNIVERSITY OF TECHNOLOGY**

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**
**FACULTY OF INFORMATION TECHNOLOGY**

**ÚSTAV INFORMAČNÍCH SYSTÉMŮ**
**DEPARTMENT OF INFORMATION SYSTEMS**


# POROVNÁNÍ KLASIFIKAČNÍCH METOD PRO ÚČELY DETEKCE MALIGNÍCH DOMÉN

**COMPARISON OF CLASSIFICATION METHODS FOR MALICIOUS DOMAIN DETECTION**

**DIPLOMOVÁ PRÁCE**
**MASTER'S THESIS**


**AUTOR PRÁCE** — **BC. JAN POLIŠENSKÝ**
**AUTHOR**

**VEDOUCÍ PRÁCE** — **Ing. RADEK HRANICKÝ, Ph.D.**
**SUPERVISOR**



**BRNO 2024**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

# Zadání diplomové práce

163352

| | |
|---|---|
| Ústav: | Ústav informačních systémů (UIFS) |
| Student: | **Polišenský Jan, Bc.** |
| Program: | Informační technologie a umělá inteligence |
| Specializace: | Kybernetická bezpečnost |
| Název: | **Porovnání klasifikačních metod pro účely detekce maligních domén** |
| Kategorie: | Bezpečnost |
| Akademický rok: | 2024/25 |

Zadání:

1. Seznamte se s metodami strojového učení se zaměřením na oblast klasifikace.
2. Nastudujte možnosti detekce maligních doménových jmen pomocí metod strojového učení.
3. Z dostatečně velkého seznamu maligních a benigních domén vytvořte anotovanou datovou sadu obsahující data z dostupných zdrojů (informace z DNS, RDAP, TLS certifikátů apod.).
4. Na základě nastudovaných informací a získaných dat vytvořte sadu klasifikátorů s využitím různých klasifikačních metod (rozhodovací stromy, neuronové sítě, SVM aj.), případně jejich kombinací.
5. Experimentálně ověřte a porovnejte použitelnost vytvořených klasifikátorů pomocí standardních metrik.
6. Zhodnoťte dosažené výsledky a navrhněte možná rozšíření.

Literatura:


- Han, Jiawei, Jian Pei, and Hanghang Tong. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2022.
- Hajaj, Chen, Nitay Hason, and Amit Dvir. 2022. "Less Is More: Robust and Novel Features for Malicious Domain Detection" *Electronics* 11, č. 6: 969.
- Torroledo, Ivan, Luis David Camacho, and Alejandro Correa Bahnsen. "Hunting malicious TLS certificates with deep neural networks." In *Proceedings of the 11th ACM workshop on Artificial Intelligence and Security*, s. 64-73. 2018.
- Shi, Yong, Gong Chen, and Juntao Li. "Malicious domain name detection based on extreme machine learning." *Neural Processing Letters* 48.3, s. 1347-1357. 2018.


Při obhajobě semestrální části projektu je požadováno:
Body 1 až 3.

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

| | |
|---|---|
| Vedoucí práce: | **Hranický Radek, Ing., Ph.D.** |
| Vedoucí ústavu: | Kolář Dušan, doc. Dr. Ing. |
| Datum zadání: | 1.11.2024 |
| Termín pro odevzdání: | 21.5.2025 |
| Datum schválení: | 22.10.2024 |

# Abstrakt

Tato práce se zaměřuje na detekci škodlivých domén pomocí metod strojového učení a porovnává výkonnost různých klasifikátorů, včetně neuronových sítí, metody podůrných vektorů a stromových algoritmů. Hlavním přínosem je návrh vícestupňové klasifikační pipeline s rozhodovacím metamodulem, která dosáhla skóre macro-F1 0,984; konkrétně skóre F1 0,985 pro phishing a 0,980 pro malware.

Navržené řešení bylo úspěšně ověřeno na nezávislé testovací sadě a porovnáno s replikovanými přístupy z literatury. Ve všech sledovaných kategoriích dosahuje výrazně lepších výsledků než existující metody. Klíčovým faktorem úspěchu je využití rozsáhlého vektoru 176 příznaků kombinujících informace z více domén (TLS, DNS, RDAP, GeoIP a lexikální analýza), který umožňuje detailnější popis charakteristik domén. Přístup založený na kombinaci různých klasifikátorů dále přispívá k robustnosti a potvrzuje jeho vhodnost pro praktické nasazení v oblasti kybernetické bezpečnosti.



# Abstract

This thesis focuses on detecting malicious domains using machine learning methods and compares the performance of various classifiers, including neural networks, support vector machines, and tree-based algorithms. Its main contribution is the design of a multi-stage classification pipeline with a decision meta-model, which achieved an excellent macro-F1 score of 0.984; specifically, an F1 score of 0.985 for phishing and 0.980 for malware.

The proposed solution was successfully validated on an independent test set and compared with replicated approaches from prior research. It significantly outperforms existing methods across all categories. A key factor in this success is the use of a rich 176-dimensional feature vector combining information from TLS, DNS, RDAP, GeoIP, and lexical analysis, allowing for a more precise characterization of domain behavior. The ensemble strategy based on combining multiple classifiers further enhances the robustness of the system and confirms its applicability for real-world cybersecurity deployment.



# Klíčová slova

maligní domény, detekce, strojové učení, neuronové sítě, SVM, phishing, malware

# Keywords

malicious domains, detection, machine learning, neural networks, SVM, phishing, malware


# Citace

POLIŠENSKÝ, Bc. Jan. *Porovnání klasifikačních metod pro účely detekce maligních domén.* Brno, 2024. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Radek Hranický, Ph.D.

# Porovnání klasifikačních metod pro účely detekce maligních domén

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Radka Hranického, Ph.D a uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Bc. Jan Polišenský
15. května 2025

## Poděkování

Děkuji svému vedoucímu Ing. Radkovi Hranickému, Ph.D., který mě při psaní této práce zásoboval připomínkami a zasloužil se o to, že je tato práce delší, než bych si představoval. Bez něj by možná tato práce nebyla nikdy dopsaná. Díky patří také člověku, který mě v průběhu práce pravidelně vracel zpátky k podstatnému a staral se o pana Bazalku, který každým dnem jen sílil a vzkvétal. Speciální poděkování patří Jurajovi a Martinovi, kteří za mě v práci převzali agendu, zatímco jsem se utápěl v LaTeXu, grafových datech a kofeinu. Díky nim jsem měl prostor tuhle práci vůbec dopsat. A snad i obhájit. V neposlední řadě děkuji Petrovi – jeho úspěšné absolvování o rok dříve mi dalo naději, že i tohle se dá přežít.

# Obsah

# Kapitola 1

# Úvod

Internetové domény tvoří základní infrastrukturu digitálního světa. Slouží jako vstupní bod k online službám, komunikačním kanálům i infrastrukturním prvkům internetu. Tato klíčová role je však zneužívána. Domény jsou často zneužívány k šíření škodlivého softwaru, provozování phishingových kampaní nebo k řízení botnetových sítí. Zvláště phishingové útoky se staly natolik sofistikovanými, že dokáží přesvědčit i odborníky z oblasti kybernetické bezpečnosti. Malware distribuovaný přes web navíc představuje vážné riziko pro síťovou infrastrukturu i koncové systémy.

Detekce maligních domén je proto zásadním nástrojem pro obranu před těmito hrozbami. Aby bylo možné škodlivé domény odhalit včas, je třeba mít k dispozici nástroje, které dokáží na základě dostupných dat rozlišit legitimní aktivity od potenciálně nebezpečných. Tradiční metody detekce, jako jsou černé listiny, jsou však vůči novým typům útoků nedostatečné – zejména kvůli vysoké variabilitě domén, krátké životnosti útočných infrastruktur a stále sofistikovanějším technikám útočníků. Je tedy nezbytné hledat robustní a adaptabilní přístupy, které umožní efektivní rozpoznání maligních domén i v podmínkách vysoké nejistoty.

Moderní útoky využívají automatizaci, šifrovaný provoz a domény registrované anonymně nebo dynamicky generované . Tyto faktory ztěžují klasifikaci a vyžadují pokročilé metody, které dokáží pracovat s různými typy dat – např. DNS záznamy, TLS certifikáty nebo informace z WHOIS. Zároveň je nutné zohlednit nevyváženost dat a složitost optimalizace klasifikátorů pro reálné nasazení.

Cílem této diplomové práce je navrhnout, implementovat a experimentálně ověřit sadu klasifikátorů pro detekci maligních domén s využitím metod strojového učení. Práce se zaměřuje na návrh vektoru příznaků, vytvoření anotované datové sady a porovnání několika typů klasifikátorů – včetně neuronových sítí, stromových algoritmů a metody podpůrných vektorů (SVM).

Významnou součástí práce je návrh víceúrovňové pipeline, která kombinuje výstupy jednotlivých modelů, používá rozhodovací metamodul a obsahuje komponentu pro detekci falešně pozitivních výsledků. Důraz je kladen na přesnost klasifikace, robustnost vůči variabilitě dat a eliminaci falešně pozitivních výsledků.

## 1.1 Zasazení práce do kontextu

Tato diplomová práce je řešena v rámci výzkumného projektu Analýza šifrovaného provozu pomocí síťových toků, realizovaného na Fakultě informačních technologií Vysokého učení

technického v Brně. Navazuje na předchozí práce v oblasti detekce škodlivých domén (např. A. Horák [37], P. Pouč [72]) a rozšiřuje je o komplexní klasifikační framework schopný pracovat s bohatými datovými vstupy.

## 1.2 Přínosy práce

Hlavní přínosy práce lze shrnout do následujících bodů:

- Vznikla rozsáhlá anotovaná datová sada obsahující více než 1 milion domén s daty z DNS, RDAP, TLS a GeoIP.
- Byla navržena a vyhodnocena sada klasifikačních modelů včetně neuronových sítí, stromových algoritmů a SVM.
- Byl sestaven rozsáhlý vektor příznaků (až 176 prvků), proveden jejich výběr a analýza přínosu.
- Byla navržena a implementována klasifikační pipeline s rozhodovacím metamodelem a modulem pro detekci falešně pozitivních vzorků.
- Výsledky byly ověřeny na validační i nezávislé verifikační sadě, kde bylo dosaženo vysoké přesnosti 0,9875 a skóre F1 0,9837 na validační sadě, a rovněž dobré generalizace na verifikační sadě s přesností 0,9536 a skóre F1 0,9413.

## 1.3 Struktura práce

Tato diplomová práce je rozdělena do jedenácti kapitol, které logicky sledují jednotlivé fáze výzkumu, návrhu, implementace a vyhodnocení systému pro detekci maligních domén:

- **Kapitola 2** se zabývá problematikou detekce maligních domén a přehledem současných metod používaných k jejich identifikaci.
- **Kapitola 3** je věnována popisu datových zdrojů domén a jejich zpracování.
- **Kapitola 4** se soustředí na přehled metod strojového učení využívaných pro klasifikaci domén.
- **Kapitola 5** popisuje původ a strukturu konkrétních datových sad využitých v rámci této práce. Dále navrhuje úpravy a transformace těchto dat pro efektivní trénink modelů.
- **Kapitola 6** se zaměřuje na návrh vektoru příznaků a definuje jejich strukturu, kategorizaci a metodiku tvorby.
- **Kapitola 7** obsahuje předběžnou analýzu skupin příznaků a porovnání výkonnosti modelů pro jednotlivé podmnožiny.
- **Kapitola 8** představuje architekturu navržené klasifikační pipeline, včetně detailního popisu jednotlivých modelů (stromové algoritmy, neuronové sítě, metamodel, FPD) a jejich integrace do víceúrovňového systému.

- **Kapitola 9** prezentuje experimentální vyhodnocení navrženého systému na validační i verifikační sadě.
- **Kapitola 10** diskutuje dosažené výsledky, analyzuje jejich praktický dopad a omezení.
- **Kapitola 11** shrnuje hlavní přínosy práce a dosažené výsledky.

# Kapitola 2

# Detekce maligních domén

Tato kapitola se zabývá problematikou detekce maligních domén a přehledem současných přístupů k jejich identifikaci. Nejprve poskytne teoretický základ pro pochopení této problematiky, poté podrobně popíše existující metody detekce.

## 2.1 Úvod do problematiky detekce maligních domén

Webové domény jsou fundamentálním stavebním kamenem internetu, sloužícím jako centrální uzly pro většinu online komunikace. Tyto domény, esenciálně adresy, umožňují uživatelům lokalizovat a přistupovat k specifickým webovým serverům. Tento proces je usnadněn díky hierarchickému systému DNS (Domain Name System), který konvertuje doménová jména na odpovídající IP adresy, a to jak pro IPv4, tak IPv6 formáty [65].

Doménové jméno může být definováno jako abstraktní entita, která obsahuje různé parametry a informace. V kontextu detekce maligních domén, jsou doménová jména považována za abstraktní objekty, reprezentující entity komunikující a poskytující obsah přes internet [21].

Běžné využití doménových jmen zahrnuje identifikaci webových stránek, e-mailových serverů a dalších online služeb. V dnešní době, však tento systém čelí výzvám spojeným s maligními aktivitami. Například zločinci využívají doménová jména pro provádění škodlivých činností, jako je phishing nebo distribuce malware [75].

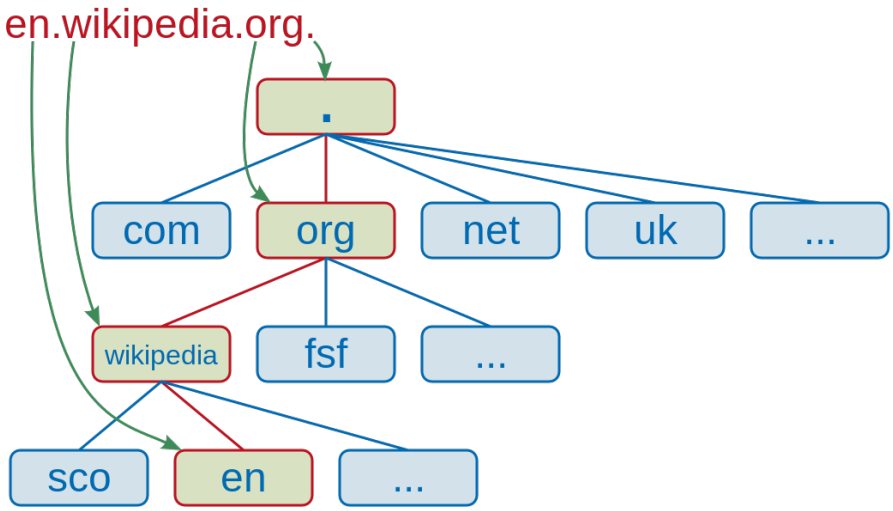
Různé typy maligních domén lze klasifikovat do kategorií na základě jejich charakteristik a zaměření. Mezi běžné kategorie patří phishingové domény, které se snaží imitovat legitimní webové stránky za účelem získání citlivých informací od uživatelů, domény spojené s distribucí malware, a domény propagující nežádoucí reklamu [66].

### 2.1.1 Systém DNS

DNS, neboli systém hierarchie doménových jmen, je zásadní internetový protokol určující strukturu a uspořádání doménových jmen. Tyto jména jsou uspořádána decentralizovaně a hierarchicky, což umožňuje překlad doménových jmen na IP adresy prostřednictvím vyhledávání v tomto systému, což je klíčové pro přístup k webovému obsahu [74].

### 2.1.2 Funkce internetových domén

Doménová jména hrají zásadní roli v uspořádání a fungování moderního internetu. Slouží jako lidsky čitelný prostředník mezi uživateli a IP adresami, což umožňuje snadnou navigaci




Obrázek 2.1: Hierarchická struktura systému DNS.

a přístup k online službám. Zároveň poskytují klíčovou infrastrukturu pro širokou škálu internetových funkcí, od webového hostingu přes e-mailové služby až po integraci aplikací a API. Tato flexibilita však přináší i rizika spojená se zneužitím [51].

Funkci domén bychom mohli shrnout do následujících bodů:

- **Identifikace zdrojů** Doménová jména umožňují jednoznačnou identifikaci zdrojů na internetu, což je zásadní pro směrování dat a poskytování služeb.
- **Značka a identita**Domény často nesou identitu organizací, produktů nebo jednotlivců, což jim dodává hodnotu nejen z technického, ale i z marketingového hlediska.
- **Hierarchické uspořádání**Struktura doménového jména (např. subdomény, hlavní doména, TLD) podporuje efektivní organizaci a škálování internetových zdrojů.
- **Zprostředkování bezpečnosti** Domény hrají klíčovou roli v implementaci bezpečnostních protokolů, jako je TLS/SSL, které zajišťují šifrovanou komunikaci.

### 2.1.3 Maligní domény

Maligní domény, definované ve širším kontextu, jsou webové domény vytvořené s účelem působit škodu nebo získávat finanční prospěch na úkor uživatelů internetu. Tyto domény jsou často zásadní součástí kybernetických hrozeb, jako jsou distribuce malware, phishingové útoky a další formy podvodné činnosti. Identifikace a monitorování maligních domén je nezbytné pro prevenci kybernetických útoků a zajištění bezpečnosti uživatelů internetu [8].

Typickým příkladem jsou phishingové domény, které imitují důvěryhodné služby, např. `secure-login-bank.com` nebo `paypal-verification.net`, přičemž cílem je oklamat uživatele k zadání citlivých údajů. Tyto názvy často využívají sociální inženýrství, kombinují důvěryhodně znějící slova a využívají domén nejvyšší úrovně, které jsou levně dostupné.

Maligní domény nejsou jasně definovanou kategorií, ale existují spíše na škále, které se liší účelem a metodami. Pro účely jednoduché klasifikace můžeme doménová jména rozdělit do následujících hlavních kategorií založených na jejich účelu a charakteristikách:

1. **Phishingové domény:** Tyto domény jsou navrženy tak, aby napodobovaly legitimní webové stránky s cílem získat citlivé informace, jako jsou přihlašovací údaje a finanční informace uživatelů [66].

2. **Domény distribuující malware:** Tento typ domén je používán k šíření škodlivého softwaru, jako jsou viry, červi nebo trojské koně, často prostřednictvím infikovaných souborů nebo skriptů [75].

3. **Domény propagující nežádoucí reklamu:** Tyto domény obvykle distribuují nevyžádanou nebo klamavou reklamu. Nicméně, je důležité si uvědomit, že hranice mezi maligními a benigními reklamními doménami je často nejasná. Ne všechny reklamní domény jsou nutně maligní, ačkoli některé mohou být použity pro škodlivé účely [88].

4. **DGA domény (Domain Generation Algorithms):** Tyto domény jsou generovány algoritmicky a často slouží k řízení botnetů. DGA domény jsou zásadní pro komunikaci mezi centrálním řídícím serverem a jednotlivými infikovanými zařízeními v botnetu. Typickým rysem je jejich zdánlivě náhodný vzhled, který ztěžuje jejich odhalení tradičními filtračními mechanismy [73].

Abstraktně lze u maligních domén vypozorovat několik charakteristik, včetně neobvyklých vzorců přístupu, podezřelého chování v síťovém provozu, a často i využití zaměňujících nebo podobných názvů k legitimním doménám, což umožňuje zastírat jejich pravou identitu a účel [71].

### 2.1.4 Phishing

Phishingové domény jsou typem maligních domén zaměřených na získávání citlivých informací od uživatelů předstíráním důvěryhodného zdroje. Tyto domény často imitují webové stránky bank, e-mailových služeb nebo jiných online platforem s cílem získat přihlašovací údaje, osobní informace nebo finanční data. Charakteristickým rysem těchto domén je vytváření vizuálně identických replik legitimních stránek. Detekce phishingových domén může být založena na analýze obsahu, porovnání s databázemi známých phishingových stránek a sledování neobvyklého chování návštěvníků [66].

### 2.1.5 Malware

Maligní domény spojené s distribucí malwaru slouží k šíření škodlivých programů a virů. Tyto domény mohou obsahovat infikovaný software, který je stažen nebo spuštěn návštěvníkem stránky. Malware může být použit pro odcizení dat, sledování chování uživatelů nebo pro vytvoření tzv. botnetů. Typické pro malwarové domény je často krátká životnost a rychlá změna názvu nebo obsahu k minimalizaci odhalení. Detekce malwarových domén vyžaduje analýzu souborů, síťového provozu a dynamického chování k identifikaci podezřelých aktivit [75].

### 2.1.6 Botnet

Domény spojené se sítěmi botnet představují centrální kontrolní body pro sítě infikovaných počítačů (botů). Tyto domény slouží k řízení botů a sběru dat z infikovaných zařízení. Sítě botnet jsou často využívány k masivnímu rozesílání spamu, útokům typu DDoS nebo k distribuci dalšího malwaru. Charakteristickým rysem botnetových domén je snaha o udržení

stability a dostupnosti pro správu botů. Detekce botnetových domén zahrnuje analýzu síťového provozu, sledování podezřelých komunikačních vzorů a identifikaci charakteristik botů [73].

### 2.1.7 DGA (Domain Generation Algorithms)

DGA domény jsou generovány algoritmicky a slouží primárně k řízení botnetů. Charakteristické pro DGA domény je jejich náhodná generace, krátká životnost a použití jako prostředků pro komunikaci s command-and-control servery. Tyto domény jsou klíčové pro skryté předávání pokynů a informací mezi sítěmi botnet a jejich řídícími servery, často slouží také jako proxy pro zabezpečení anonymity a komplikaci sledování [94].

#### Detekce a Charakteristiky Maligních Domén

Detekce maligních domén je klíčovým prvkem v kybernetické bezpečnosti. Analytici se zaměřují na specifické charakteristiky domén, které mohou indikovat škodlivost. Mezi tyto charakteristiky patří podezřelé znaky v doménovém jméně, neobvyklé kombinace písmen a čísel, nečekané změny v chování domény a další.

Různé metody detekce zahrnují analýzu DNS provozu, sledování historie domén, a využití strojového učení a umělé inteligence pro identifikaci vzorců a anomálií. Efektivní detekce vyžaduje komplexní přístup a neustálou aktualizaci technik pro odhalení stále se vyvíjejících metod kybernetických hrozeb [14].

Následující části této kapitoly podrobně rozebírají jednotlivé metody detekce a strategie pro účinnou identifikaci maligních domén v dynamickém prostředí internetu.

Klasifikace maligních domén vyžaduje systematický přístup k identifikaci indikátorů jejich škodlivé povahy. Kromě specifických ukazatelů pro phishing, malware a botnety existují obecné indikátory, které lze využít při analýze a klasifikaci těchto domén:

- **Krátká životnost:** Maligní domény často existují jen krátkou dobu, aby minimalizovaly riziko odhalení. Sledování délky existence domény může být klíčovým ukazatelem [96].
- **Neobvyklé chování:** Abnormální chování domény, jako například neobvyklý provoz, může být známkou škodlivé aktivity. Analýza chování může odhalit podezřelé vzory [14].
- **Častá změna obsahu nebo názvu:** Maligní domény se často snaží zůstat v krytu tím, že často mění svůj obsah nebo název. Sledování těchto změn může představovat klíčový indikátor [96].
- **Nízká reputace:** Využívání reputačních databází pro hodnocení domén může poskytnout informace o tom, zda byla doména dříve spojena s nekalými praktikami [9].
- **Podezřelé WHOIS informace:** Nepravdivé nebo anonymní údaje v registračních informacích (WHOIS) mohou být indikátorem snahy skrýt pravé záměry domény [14].
- **Špatně definovaná struktura:** Nedostatek jasné struktury nebo nekonzistence v obsahu domény může naznačovat, že byla vytvořena za účelem škodlivé činnosti [96].

## 2.2 Současné přístupy detekce maligních domén

Jak již bylo řečeno, internetové domény jsou často využívány k šíření malwaru, phishingovým útokům nebo jako nástroje pro řízení botnetů [75, 14]. V reakci na tento problém byly vyvinuty různé techniky, od tradičních černých listin až po pokročilé metody strojového učení a hlubokých neuronových sítí [58, 89]. Moderní metody, jako je analýza DNS dotazů nebo analýza algoritmů DGA, prokazují vysokou účinnost při identifikaci nově vznikajících hrozeb [14, 73].

### 2.2.1 Černé listiny

Černé listiny představují základní nástroj pro detekci maligních domén, sestavovaný a udržovaný různými organizacemi. Tento přístup využívá ruční hlášení uživateli, automatizované monitorování a integraci znalostí třetích stran [37]. Ačkoli jsou černé listiny efektivní a snadno implementovatelné, mohou být velmi omezené v případě nově vznikajících hrozeb.

Moore a Clayton ve své studii zkoumají účinnost odstranění phishingových webů, což je přístup úzce spojený s používáním černých listin. [66] Provos et al. se zaměřují na analýzu malwaru šířeného prostřednictvím webových stránek, což je klíčové pro pochopení, jak černé listiny pomáhají v identifikaci škodlivých domén. [76] Thomas et al. poskytují pohled na reklamní taktiky využívající maligní domény, což je oblast, kde černé listiny mohou hrát roli v detekci a prevenci [88].

### 2.2.2 Lexikální analýza

Lexikální analýza maligních domén se zaměřuje na analýzu a klasifikaci textu doménových jmen. Výzkumy v této oblasti představují lehký, ale efektivní přístup k proaktivní detekci a kategorizaci škodlivých URL. Algoritmy strojového učení, jako jsou Random Forest, K-Nearest Neighbour (KNN), Decision Tree a Extra Tree Classifier, byly použity k analýze více než 700 000 URL, přičemž byly klasifikovány do kategorií jako benigní, defacement, malware a phishing. Bylo zjištěno, že použití pouze 15 nejdůležitějších lexikálních vlastností je dostačující a nezbytné pro klasifikaci těchto kategorií [58, 70].

Důležitým aspektem této metody je rozdělení výsledků klasifikace do dvou částí: a) výsledky klasifikace různými algoritmy strojového učení založenými na lexikálních vlastnostech; a b) matice záměn a AUC-ROC křivka pro dva nejlepší algoritmy - Random Forest a Extra Tree. Tento přístup poskytuje spolehlivou a užitečnou metodu pro detekci nových typů útoků, zejména v případě, kdy je k dispozici velké množství URL [80, 86].

### 2.2.3 Využití kombinovaných CNN-GRU-Attention modelů

Tento přístup k detekci maligních domén spočívá v použití kombinace konvolučních neuronových sítí (CNN), rekurentních neuronových sítí s jednotkami Gated Recurrent Unit (GRU) a Attention mechanismů. Tento model je schopen efektivně zpracovávat jak prostorové vlastnosti (pomocí CNN), tak i časové závislosti v datech (pomocí GRU). Attention mechanismus pak umožňuje modelu lépe se zaměřit na relevantní části dat, což vede k větší přesnosti při klasifikaci domén. Tento přístup nabízí lepší výsledky než tradiční metody, protože dokáže zachytit složitější vzory a vztahy v datech [47].

### 2.2.4 Adopce strojového učení pro podporu detekce maligních domén

Tento přístup zahrnuje využití různých algoritmů strojového učení pro klasifikaci domén jako maligních nebo neškodných. Zahrnuje tvorbu a analýzu rozsáhléh datové sady domén, které jsou předem klasifikovány jako maligní nebo benigní. Výzkum zahrnuje aplikaci různých algoritmů strojového učení, jako jsou Naive Bayes, Support Vector Machines, Decision Trees, Random Forests, Logistic Regression a Neural Networks, na tuto datovou sadu. Výsledky ukazují, že některé algoritmy dosahují přesnosti klasifikace mezi 0.75 a 0.92, přičemž čas potřebný pro klasifikaci nové domény se pohybuje od několika sekund do více než hodiny [59].

### 2.2.5 Analýza DNS dotazů pomocí strojového učení

Tento přístup k detekci maligních domén spočívá v analýze DNS dotazů s využitím metod strojového učení. Metoda se zaměřuje na extrakci a analýzu charakteristik DNS dotazů, jako jsou počet dotazů, časové intervaly mezi dotazy, a typy požadovaných záznamů. Strojové učení je následně využito k detekci anomálií v těchto charakteristikách, které mohou indikovat maligní aktivity. Algoritmy jako Random Forest, Support Vector Machines nebo neurální sítě jsou aplikovány na data, což umožňuje identifikaci potenciálně škodlivých domén s vysokou přesností. Tento přístup je efektivní zejména v detekci nových a neznámých hrozeb, které nejsou zahrnuty v existujících černých listinách nebo databázích [19].

### 2.2.6 Detekce phishingových certifikátů pomocí hlubokých neuronových sítí

Vzhledem k nárůstu využití šifrované komunikace v rámci TLS certifikátů kybernetickými útočníky, kteří často zneužívají certifikáty k zakrývání phishingových aktivit, se objevily nové techniky zaměřené na detekci škodlivých certifikátů pomocí hlubokých neuronových sítí. Torroledo a kol. (2018) navrhli model, který analyzuje obsah TLS certifikátů a identifikuje škodlivé vzorce využívané útočníky. Systém dosahuje přesnosti 94.87 % při detekci malwarových certifikátů a 88.64% při detekci phishingových certifikátů. Tento model využívá kombinace klasifikačních prvků extrahovaných z certifikátů a schopnosti hlubokých neuronových sítí analyzovat textová data obsažená v certifikátech, čímž umožňuje odhalit i skryté vzorce v chování útočníků. Tento přístup ukazuje na potenciál hlubokého učení v oblasti kybernetické bezpečnosti a na jeho efektivitu v porovnání s tradičními metodami, jako je podpora vektorových strojů (SVM), kterou tento model převyšuje přesností o 7% [89].

### 2.2.7 Porovnání metod strojového učení pro detekci phishingu

Další výzkum v oblasti detekce phishingových útoků se zaměřuje na porovnání různých metod strojového učení, včetně logistické regrese, rozhodovacích stromů, bayesovských aditivních regresních stromů (BART), náhodných lesů a neuronových sítí. Studie Abu-Nimeha a kol. (2007) testuje predikční přesnost těchto metod na datové sadě obsahující phishingové a legitimní e-maily, přičemž jednotlivé metody vykazují různé úrovně přesnosti. Například náhodné lesy vykazují nejlepší výkonnost s přesností 90,24 % při vážení chyb na základě penalizace falešně pozitivních detekcí [2]. Tato studie zdůrazňuje význam pečlivého výběru klasifikátorů pro detekci phishingových aktivit, protože různé přístupy poskytují různé úrovně vyváženosti mezi falešně pozitivními a falešně negativními výsledky. Tento

přístup poskytuje základní vhled do toho, jak mohou různé modely přispět k účinné detekci phishingových útoků na základě specifických charakteristik phishingových zpráv a vzorců chování útočníků [2].

### 2.2.8 Detekce pomocí lexikální analýzy a síťových aktivit

Amit Kumar a Soumyadev Maity představili hybridní přístup k identifikaci a kategorizaci škodlivých URL pomocí kombinace lexikálních vlastností a síťových aktivit. Ve svém výzkumu použili rozsáhlou datovou sadu obsahující více než 700 000 URL, které byly klasifikovány do kategorií benigní, phishing, malware a defacement. Analyzované vlastnosti zahrnovaly délku URL, délku hostname, entropii a další metriky odvozené z textu URL.

Byly testovány klasifikační metody jako Random Forest, K-Nearest Neighbors, Decision Tree a Extra Tree Classifier. Nejlepších výsledků dosáhly algoritmy Random Forest a Extra Tree Classifier s přesností až 93 %. Tento přístup prokázal, že kombinace lexikálních vlastností a síťových aktivit umožňuje přesnou a rychlou detekci škodlivých URL. Autoři zdůraznili, že jejich metoda je obzvláště efektivní při analýze rozsáhlých datových sad a může být implementována jako klíčový nástroj pro boj proti phishingovým a malwarovým útokům [52].

## 2.3 Srhnutí současných přístupů a jejich přínosu

Na základě analýzy jednotlivých metod lze konstatovat, že žádný přístup není univerzální. Tradiční metody, jako jsou černé listiny, jsou rychlé a snadno implementovatelné, avšak jejich efektivita je omezena u nově vznikajících hrozeb. Na druhé straně pokročilé metody strojového učení nabízejí vyšší přesnost a adaptabilitu, avšak vyžadují větší výpočetní výkon a pečlivě připravené datové sady.

Integrace více přístupů do jednoho komplexního systému detekce maligních domén se jeví jako nejefektivnější strategie. Kombinace blacklistingu s analýzou DNS provozu a využitím modelů hlubokého učení může výrazně snížit míru falešně pozitivních i falešně negativních detekcí a zvýšit schopnost systému reagovat na nové typy hrozeb.

### 2.3.1 Využití rozsáhlých dat pro detekci maligních domén

S nárůstem počtu dostupných dat a pokročilých metod jejich analýzy se využití rozsáhlých datových sad a bohatých vektorů příznaků stalo klíčovým přístupem k efektivní detekci maligních domén. Moderní algoritmy dokáží těžit z kombinace různorodých zdrojů dat, jako jsou DNS dotazy, WHOIS informace, síťové aktivity, a vlastnosti získané lexikální analýzou [14, 58].

**Rozsáhlá data při klasifikaci** Rozšířené vektory příznaků zahrnují kombinaci tradičních metrik, jako je délka doménového jména, entropie nebo počet poddomén, spolu s pokročilými vlastnostmi získanými z dynamické analýzy, např. vzorce síťového provozu nebo doba existence domény. Tato rozmanitost umožňuje pokročilým modelům strojového učení, včetně hlubokých neuronových sítí, zachytit komplexní vzory a vztahy, které by tradiční metody mohly přehlédnout [47, 14].

Přístup spočívající v maximalizaci vektoru příznaků byl prokázán jako účinný zejména při využití pokročilých modelů, jako jsou konvoluční neuronové sítě (CNN) a modely využívající mechanismy pozornosti (Attention). Například modely kombinující CNN a rekurentní

neuronové sítě (GRU) ukázaly, že čím širší spektrum vlastností je analyzováno, tím přesnější je detekce [47].

**Výhody a optimalizace** Použití rozsáhlých datových sad nejen zlepšuje přesnost modelů, ale také umožňuje efektivní detekci nově vznikajících a neznámých hrozeb. Analýzy DNS provozu ukázaly, že data o časových intervalech mezi dotazy nebo geografické distribuci žádostí mohou poskytnout klíčové informace o anomálních vzorcích chování [19].

Optimalizace vektoru příznaků zahrnuje výběr relevantních vlastností a eliminaci redundantních nebo irelevantních informací. Metody jako PCA (Principal Component Analysis) nebo výběr na základě informačního zisku jsou často využívány pro zlepšení efektivity bez kompromisů na přesnosti [58, 14].

Tento přístup založený na rozsáhlých datových vektorech přináší významné výhody pro detekci maligních domén. Kombinace různorodých datových zdrojů s pokročilými analytickými metodami umožňuje vytvářet robustní modely schopné adaptace na dynamicky se měnící hrozby.

### 2.3.2 Vývoj v oblasti detekce maligních domén

Oblast detekce maligních domén zaznamenala v posledních dekádách významný vývoj, který odráží rostoucí sofistikovanost kybernetických hrozeb a pokrok v oblasti analýzy dat. Počáteční přístupy byly založeny na statických černých listinách, které se ukázaly jako účinné proti známým hrozbám [75, 66]. Tyto přístupy však byly postupně překonány dynamikou moderních útoků, jež využívají rychlé změny a maskování doménových jmen, což vedlo k rozvoji metod strojového učení.

Metody založené na strojovém učení, jako je lexikální analýza doménových jmen, přinesly významnou změnu díky schopnosti rychle a přesně klasifikovat velké množství URL [58, 70]. Další pokroky zahrnovaly analýzu DNS provozu, která umožnila identifikaci anomálních vzorců v síťovém provozu, a detekci pomocí TLS certifikátů, která odhalila skryté vzory chování útočníků [19, 89].

V posledních letech se do popředí dostávají hluboké neuronové sítě, jako jsou modely CNN a GRU s mechanizmy pozornosti (Attention), které dokáží efektivně zachytit prostorové i časové závislosti v datech. Tyto modely dosahují vysoké přesnosti při detekci nových typů hrozeb [47]. Například model Torroledo a kol. dosáhl přesnosti 94.87% při detekci malwarových certifikátů [89].

Dalším klíčovým trendem je integrace více zdrojů dat, jako jsou DNS záznamy, WHOIS informace, síťové aktivity a vlastnosti domén. Tyto multidimenzionální přístupy, často implementované prostřednictvím složených modelů, zvyšují robustnost detekčních systémů a umožňují lepší reakci na nově vznikající hrozby [14, 2].

Vývoj v této oblasti ukazuje, že budoucnost detekce maligních domén leží v kombinaci rychlosti, přesnosti a adaptivních schopností, což umožňuje efektivní boj proti stále sofistikovanějším kybernetickým hrozbám.

# Kapitola 3

# Zdroje dat a jejich zpracování

Tato kapitola bude věnována zdrojům dat internetových domén a jejich zpracování, které jsou nezbytné pro efektivní detekci a analýzu maligních domén. Pochopení a správná aplikace metod zpracování dat jsou klíčové pro identifikaci a klasifikaci škodlivých domén. Kapitola poskytne přehled o různých typech dat, které jsou k dispozici pro výzkum a praxi v oblasti kybernetické bezpečnosti.

První část kapitoly se zaměří na základní a univerzální zdroje dat, následně budou představeny zdroje pro škodlivé domény a na závěr bude rozebrána příprava dat.

## 3.1 Základní zdroje doménových dat

Zdroje dat poskytují klíčové informace potřebné pro analýzu a hodnocení domén z hlediska jejich důvěryhodnosti či potenciální škodlivosti. Dále budou rozebrány především protokoly DNS, WHOIS, RDAP, TLS certifikáty a geolokační data.

### 3.1.1 Domain Name System (DNS)

Jak bylo uvedeno v sekci 2.1.1, Domain Name System (DNS) je zásadní protokol internetové infrastruktury, jehož hlavním úkolem je převod doménových jmen na odpovídající IP adresy. DNS využívá hierarchickou a decentralizovanou strukturu, která umožňuje efektivní směrování uživatelských požadavků na cílové servery. Tento systém je nezbytný pro správné fungování internetu a tvoří základní vrstvu propojení mezi uživateli a zdroji na webu [74].

DNS uchovává informace o doménách ve formě tzv. *Resource Records* (RR), které poskytují různé typy údajů o konfiguraci a funkcích jednotlivých domén. Mezi nejvýznamnější patří:

- **A a AAAA záznamy**: Slouží k mapování domén na IPv4 nebo IPv6 adresy, což je základní funkce DNS pro zajištění dostupnosti online zdrojů.
- **MX záznamy**: Uvádějí e-mailové servery odpovědné za zpracování pošty pro konkrétní doménu. Tyto záznamy jsou klíčové pro ochranu proti nevyžádané poště a phishingovým kampaním, protože lze analyzovat neobvyklé nebo podezřelé konfigurace.
- **TXT záznamy**: Umožňují ukládání volitelných textových dat, která se často využívají k ověřování bezpečnostních mechanismů, jako jsou SPF, DKIM nebo DMARC, jež zajišťují ochranu proti spoofingu e-mailů.

- **CNAME záznamy**: Poskytují aliasování domén, což umožňuje mapovat více názvů na stejný server. Tato vlastnost je často využívána v kombinaci se službami CDN (Content Delivery Network).
- **NS záznamy (Name Server)**: Definují autoritativní DNS servery, které jsou odpovědné za konkrétní doménu. Tyto záznamy určují, které servery mají být dotazovány pro získání informací o doméně.
- **SOA záznamy (Start of Authority)**: Obsahují základní informace o doméně, například primární DNS server, e-mail administrátora a intervaly aktualizací záznamů. Jsou nezbytné pro správnou správu zón.
- **PTR záznamy (Pointer)**: Používají se při reverzním DNS, kde je IP adresa mapována zpět na doménové jméno. Tato funkce je užitečná například při konfiguraci e-mailových serverů nebo detekci zneužití IP adres.
- **SRV záznamy (Service)**: Určují umístění specifických služeb, například VoIP nebo služby instant messagingu, a umožňují vyhledávání podle protokolu a priority.
- **CAA záznamy (Certificate Authority Authorization)**: Definují, které certifikační autority mohou vydávat TLS/SSL certifikáty pro danou doménu, čímž přispívají k zabezpečení komunikace.
- **NAPTR záznamy (Naming Authority Pointer)**: Poskytují pokročilé přesměrování a jsou často využívány v telekomunikačních aplikacích, například při překladu telefonních čísel na URI.
- **DNSKEY záznamy**: Obsahují veřejné klíče používané v rámci DNSSEC (Domain Name System Security Extensions), které poskytují autentizaci DNS záznamů a chrání před podvržením.
- **DS záznamy (Delegation Signer)**: Používají se v DNSSEC a obsahují hash klíče podepisujícího zónu, čímž umožňují propojení bezpečnostních klíčů mezi rodičovskou a podřízenou doménou.

### Pasivní DNS (pDNS)

Pasivní DNS (*Passive DNS*) je metoda umožňující monitorování DNS provozu, přičemž zachycuje a ukládá informace o dotazech a odpovědích mezi servery. Tento přístup přináší možnost analýzy historických dat, což je klíčové pro identifikaci vzorců zneužívání domén. Pasivní DNS poskytuje globální pohled na chování domén a odhaluje podezřelé aktivity, jako je využívání krátkodobých domén pro phishingové kampaně nebo sítě botnet [84].

### Využití DNS při analýze domén

Kromě své primární funkce lze DNS využít jako cenný zdroj dat pro detekci podezřelých aktivit spojených s maligními doménami:

- *NXDOMAIN Responses*: Vysoký počet odpovědí označujících neexistující domény může naznačovat aktivitu malwaru nebo botnetů, které generují velké množství dotazů na domény vytvořené algoritmem DGA (*Domain Generation Algorithm*) [97].

- *TTL analýza*: Časté změny hodnot *Time to Live* mohou indikovat použití metody *Fast Flux*, kde útočníci pravidelně mění IP adresy domén za účelem zakrytí jejich skutečného umístění [84].

### 3.1.2 WHOIS a RDAP

WHOIS a RDAP (Registration Data Access Protocol) jsou dva zásadní protokoly poskytující informace o vlastnictví, správě a stavu domén. Tyto zdroje dat hrají klíčovou roli při analýze domén, umožňují identifikaci jejich legitimity a pomáhají odhalovat potenciálně maligní aktivity [82, 17].

#### WHOIS

WHOIS je tradiční protokol používaný již od počátků internetu. Zajišťuje přístup k údajům o doménách, jako jsou informace o registrátorovi a vlastníkovi domény, data registrace a expirace či aktuální stav domény (aktivní, pozastavená, expirovaná apod.) [65]. Typicky poskytuje tyto informace:

- **Identifikace registrátora a vlastníka**: Zahrnuje jméno, adresu a kontaktní informace. Tyto údaje mohou být u některých registrů anonymizovány prostřednictvím služeb na ochranu soukromí (*WHOIS Privacy*).
- **Časové informace**: Datum registrace, expirace a poslední aktualizace záznamu. Nově registrované domény mohou být varovným signálem pro phishingové kampaně nebo jiné škodlivé aktivity [57].
- **Stav domény**: Informace o tom, zda je doména aktivní, pozastavená, nebo ve fázi přenosu, což může být důležité pro odhalování zneužití domény.

Ačkoli WHOIS poskytuje cenná data, jeho nevýhodou je často nejednotný formát a chybějící standardizace napříč registry [17]. Navíc ochrana soukromí u některých registrátorů může omezit dostupnost informací. V důsledku těchto omezení vznikl modernější protokol RDAP.

#### RDAP (Registration Data Access Protocol)

Protokol RDAP byl navržen jako standardizovaná a bezpečnější alternativa k WHOIS. Tento protokol poskytuje strukturovaná data ve formátu JSON, což usnadňuje jejich zpracování a analýzu. RDAP zahrnuje podporu pro řízení přístupu, což znamená, že různí uživatelé mohou mít přístup k různým úrovním detailů na základě jejich oprávnění [82]. Klíčové vlastnosti RDAP zahrnují:

- **Strukturovaný formát**: Data jsou organizována ve formátu JSON, což umožňuje snadnou integraci s analytickými nástroji a automatizované zpracování.
- **Podpora pro zabezpečení**: RDAP implementuje autentizaci a autorizaci přístupu, což chrání data před neoprávněným využitím [57].
- **Dodatečné informace**: Kromě základních údajů může RDAP poskytovat metadata, jako jsou geolokační informace, informace o subjektech spravujících doménu či technické údaje o DNS serverech [17].

RDAP umožňuje analýzu dat v kontextu většího počtu atributů než WHOIS, což jej činí výhodným při zkoumání potenciálně škodlivých domén. Například detekce anomálních vzorců v registracích (např. masové registrace anonymními subjekty) může být usnadněna díky transparentnějšímu přístupu ke struktuře dat.

#### Využití WHOIS a RDAP při analýze domén

WHOIS a RDAP se často využívají při analýze důvěryhodnosti domén a identifikaci potenciálně maligních aktivit. Mohou odhalit:

- **Podezřelé registrátory**: Některé registrátory jsou známé tím, že umožňují anonymní registrace nebo mají slabší bezpečnostní politiku [82].
- **Rychlé registrace a expirace**: Maligní domény bývají často registrovány na krátkou dobu a jejich platnost se pravidelně obnovuje nebo expirováno během krátkého období [57].
- **Geografické vzory**: Analýza země původu registrátora může odhalit oblasti spojené s častým hostingem škodlivých domén [17].

WHOIS a RDAP jsou tak základními nástroji při shromažďování dat pro systémy detekce škodlivých domén, přičemž RDAP se díky své moderní infrastruktuře stává preferovanou volbou [17].

### 3.1.3 TLS Certifikáty

TLS (Transport Layer Security) certifikáty jsou klíčovou součástí infrastruktury veřejných klíčů (PKI) a zajišťují bezpečnou komunikaci mezi klienty a servery na internetu. Certifikáty založené na standardu X.509 obsahují informace o identitě serveru, veřejném klíči a dalších parametrech nezbytných pro šifrování a autentizaci. [79, 24]

#### Struktura TLS certifikátu

TLS certifikát obsahuje několik klíčových komponent, které umožňují jeho použití v kryptografickém procesu [24, 16].

- **Subjekt certifikátu**: Identifikuje entitu, pro kterou byl certifikát vydán, například konkrétní doménu (*Common Name, CN*) nebo organizaci. Může zahrnovat také alternativní názvy subjektu (*Subject Alternative Name, SAN*) pro podporu více domén či subdomén.
- **Vydavatel certifikátu**: Identifikuje certifikační autoritu (CA), která certifikát vydala. Tato informace obsahuje název CA, její digitální podpis a případná rozšíření.
- **Veřejný klíč**: Kryptografický klíč, který je použitý při inicializaci šifrované komunikace. Veřejný klíč slouží ke šifrování dat, která může dešifrovat pouze odpovídající privátní klíč.
- **Sériové číslo certifikátu**: Jedinečný identifikátor certifikátu, který je přiřazen vydavatelem.
- **Platnost certifikátu**: Definuje časové období, během kterého je certifikát platný. Obsahuje datum a čas vydání a expirace.

- **Rozšíření certifikátu**: Tato část zahrnuje další informace, například omezení na specifické aplikace nebo protokoly (*Key Usage*), podporu šifrovacích algoritmů či identifikaci hierarchie certifikačních autorit (*Certificate Policies*).
- **Digitální podpis**: Podpis vytvořený certifikační autoritou, který ověřuje autenticitu certifikátu. Používá hash původního obsahu certifikátu šifrovaný privátním klíčem CA.

### Funkce TLS certifikátů v PKI

TLS certifikáty fungují jako součást infrastruktury veřejných klíčů, která zajišťuje důvěryhodnost a bezpečnost v rámci internetové komunikace. Tento proces zahrnuje několik klíčových kroků [24, 87].

1. **Vydání certifikátu**: Certifikační autorita (CA) po ověření identity subjektu vystaví certifikát, který obsahuje veřejný klíč a další informace o subjektu.
2. **Distribuce certifikátu**: Certifikát je distribuován klientům prostřednictvím protokolu TLS během procesu navazování spojení (*TLS handshake*).
3. **Validace certifikátu**: Klient při navazování spojení ověřuje podpis certifikátu proti seznamu důvěryhodných certifikačních autorit (uložených v tzv. *trust store*) a kontroluje jeho platnost (např. datum expirace a zrušení).
4. **Použití veřejného klíče**: Veřejný klíč obsažený v certifikátu je využit k šifrování přenosu nebo k ověření digitálních podpisů serveru.

### TLS Handshake a role certifikátů

TLS handshake je proces navazování šifrovaného spojení mezi klientem a serverem. Certifikáty zde hrají klíčovou roli v autentizaci a zabezpečení komunikace: [79, 87]

- **Serverová autentizace**: Server poskytuje svůj certifikát klientovi, který ověřuje jeho pravost proti důvěryhodným CA.
- **Sdílení šifrovacích parametrů**: Na základě informací z certifikátu a zvolených algoritmů je mezi klientem a serverem vygenerován šifrovací klíč pro zabezpečení další komunikace.
- **Volitelné ověření klienta**: V některých scénářích může být použit i klientský certifikát pro oboustrannou autentizaci.

### Typy TLS certifikátů

TLS certifikáty mohou být kategorizovány podle úrovně validace a použití [16, 11].

- **Domain Validation (DV)**: Ověřuje vlastnictví domény, ale neposkytuje informace o organizaci. DV certifikáty jsou rychle vydávány a jsou nejčastěji využívány menšími weby [11].
- **Organization Validation (OV)**: Ověřuje vlastnictví domény i identitu organizace. OV certifikáty zajišťují větší důvěryhodnost a jsou vhodné pro organizace poskytující citlivé služby, například finanční instituce [16].

- **Extended Validation (EV)**: Poskytuje nejvyšší úroveň důvěry, zahrnuje přísné ověření organizace a je vizuálně zvýrazněn v prohlížečích (např. zelený adresní řádek u starších prohlížečů) [16].
- **Wildcard certifikáty**: Platí pro doménu a všechny její subdomény. Jsou vhodné pro organizace, které potřebují zabezpečit větší množství subdomén [11].
- **Multi-Domain (SAN) certifikáty**: Umožňují zabezpečení více domén v jednom certifikátu. Tento typ je využíván například v prostředích s virtualizací nebo pro služby využívající více aliasů [16].

### 3.1.4 Geolokační informace

Geolokační data představují klíčový zdroj informací pro analýzu síťových aktivit a bezpečnostních hrozeb. Tato data lze odvodit z IP adres serverů a klientů, přičemž poskytují informace o jejich geografické poloze, jako je země, město, region nebo dokonce přibližné GPS souřadnice. Geolokační informace lze získat z následujících zdrojů:

- **Databáze geolokací IP adres**: Služby, jako je MaxMind GeoIP, IP2Location nebo RIPE NCC, nabízejí pravidelně aktualizované databáze, které mapují IP adresy na geografické lokace [61, 44].
- **WHOIS záznamy**: Informace o registraci IP adres, poskytované regionálními internetovými registry (např. ARIN, RIPE, APNIC), mohou obsahovat údaje o geografické oblasti registrace [7].
- **Pasivní DNS**: Kombinace dat o doménách a jejich IP adresách umožňuje mapování vztahů mezi doménami a jejich geografickou distribucí [84].

#### Přesnost geolokačních dat

Přesnost geolokačních dat závisí na několika faktorech:

- **Úroveň podrobnosti**: Přesnost se může pohybovat od zjištění země nebo regionu (většinou přesné na více než 95 %) až po lokalizaci na úroveň města, kde přesnost může klesnout na 50–70 % [43].
- **Typ IP adresy**: Přesnost je obecně vyšší u statických IP adres než u dynamických, které mohou být přidělovány do různých geografických oblastí.
- **Proxy a VPN**: Použití proxy serverů, VPN nebo cloudových služeb může zkreslit geolokační data a naznačovat falešné umístění [63].
- **Aktualizace databází**: Četnost aktualizací geolokačních databází ovlivňuje aktuálnost a přesnost informací.

I přes široké využití mají geolokační data své limity, které ovlivňují jejich přesnost a spolehlivost. Jedním z hlavních problémů je nejasné umístění cloudových služeb. Poskytovatelé, jako jsou AWS nebo Google Cloud, často využívají geograficky distribuované IP adresy, což výrazně komplikuje přesnou lokalizaci serverů a dalších zdrojů. Další výzvu představují mobilní sítě, kde operátoři obvykle přidělují IP adresy na základě logiky své sítě, nikoli podle fyzického umístění uživatele. To může vést k významným rozdílům mezi skutečnou

polohou uživatele a místem, které geolokační databáze indikuje. Kromě toho mohou geolokační data zkreslovat útočníci, kteří využívají technologie, jako jsou proxy servery nebo VPN, k zakrytí své skutečné polohy. Tyto metody umožňují vytvořit falešnou geografickou stopu, která může zmást detekční systémy a ztížit analýzu [63].

## 3.2 Zdroje dat pro maligní domény

Veškeré výše zmíněné zdroje dat, jako jsou DNS, WHOIS, RDAP nebo TLS certifikáty, poskytují cenné informace pro analýzu domén. Tyto zdroje umožňují získat široké spektrum atributů, které lze využít při klasifikaci domén. Nicméně, aby bylo možné doménu označit jako maligní, je klíčové mít k dispozici informace o její škodlivosti. Tyto informace, označované často jako *ground truth*, slouží jako základ pro trénování a vyhodnocování modelů. Následující sekce se věnují zdrojům, které poskytují informace o škodlivých doménách, jako jsou zdrojová data platformy MISP a černé listiny.

### 3.2.1 MISP

MISP (*Malware Information Sharing Platform*) je platforma pro sdílení informací o kybernetických hrozbách. Tato platforma umožňuje bezpečnostním týmům a organizacím sdílet indikátory kompromitace (*Indicators of Compromise, IoC*), mezi které patří informace o škodlivých doménách, IP adresách, hashe souborů a dalších entitách. Zdrojová data platformy MISP jsou sestavována na základě údajů od různých organizací, zahrnujících bezpečnostní společnosti, výzkumné týmy a národní kybernetické agentury [91].
Zdrojová data platformy MISP poskytují následující klíčové informace:

- **Detailní metadata o škodlivých doménách**: Obsahují informace o typu hrozby, čase detekce, původu a dalších relevantních atributech.
- **Kontext hrozeb**: Například informace o tom, zda je doména spojena s phishingovou kampaní, šířením malwaru nebo provozem botnetů.
- **Automatizovaná integrace**: Díky podpoře standardizovaných formátů, jako STIX nebo OpenIOC, lze data snadno integrovat do systémů detekce a reakce na incidenty (SIEM).

Zdrojová data platformy MISP jsou užitečná zejména při průběžné aktualizaci datových sad a obohacení modelů pro detekci maligních domén o aktuální informace o hrozbách [5].

#### Využití bezpečnostních feedů

Využití MISP zahrnuje integraci s různými bezpečnostními nástroji a systémy, jako jsou SIEM systémy (Security Information and Event Management), síťové senzory, firewally a další nástroje pro prevenci a detekci hrozeb. Díky MISP lze tyto nástroje konfigurovat tak, aby automaticky reagovaly na aktuální informace, což umožňuje rychlou reakci na nově identifikované hrozby [91].

#### Software a Technologie

MISP poskytuje API (Application Programming Interface), které umožňuje snadnou integraci s různými bezpečnostními řešeními. Díky této funkcionalitě je MISP často integrován

s dalšími nástroji a platformami, jako jsou Threat Intelligence Platforms (TIP), což umožňuje komplexní správu a analýzu hrozeb. Kromě toho MISP podporuje různé formáty pro sdílení informací, včetně STIX/TAXII, což umožňuje širší kompatibilitu s různými systémy a standardy [64].

### 3.2.2 Černé listiny

Černé listiny (*blacklists*) představují další zásadní zdroj informací o škodlivých doménách. Jak je popsáno v sekci 2.2.1, jedná se o seznamy udržované různými organizacemi a automatizovanými systémy, které identifikují známé škodlivé domény. Tyto seznamy jsou často sestavovány na základě ručního hlášení uživatelů, automatizovaného monitorování a analýzy škodlivých aktivit. Mezi nejznámější černé listiny patří například Google Safe Browsing, PhishTank nebo Spamhaus [76, 66, 88].

Černé listiny jsou užitečné nejen při identifikaci známých hrozeb, ale také jako základ pro *ground-truth verifikaci* (viz kapitola 5.4). Slouží k validaci domén označených jako maligní a pomáhají zajistit spolehlivost datových sad. Nicméně jejich omezení spočívají v neschopnosti detekovat nově vznikající hrozby nebo hrozby, které se vyhýbají běžným detekčním metodám.

#### VirusTotal API

VirusTotal je populární online služba, která kombinuje více než 70 antivirových enginů a bezpečnostních nástrojů pro analýzu souborů, URL a domén. Poskytuje podrobné informace o škodlivých aktivitách spojených s analyzovanými objekty. VirusTotal API umožňuje přístup k této službě prostřednictvím automatizovaných skriptů a je užitečné jak pro jednotlivé analýzy, tak pro hromadné zpracování datových sad. [90]

**Funkce VirusTotal API** VirusTotal API nabízí následující klíčové funkce: [90]

- **Hodnocení domén**: VirusTotal poskytuje skóre založené na počtu detekcí jednotlivých antivirových enginů, což pomáhá rychle určit, zda je doména maligní.
- **Metadata o doménách**: Služba poskytuje informace o IP adresách, DNS záznamech, TLS certifikátech a dalších atributech spojených s analyzovanou doménou.
- **Historické údaje**: Umožňuje analyzovat historické chování domén, což je užitečné při sledování trendů a dlouhodobých aktivit.

**Využití VirusTotal API při verifikaci** VirusTotal API hraje klíčovou roli při *ground-truth verifikaci* (viz kapitola 5.4). V rámci tohoto procesu byly všechny analyzované domény porovnány s databázemi VirusTotal, což umožnilo víceúrovňové hodnocení. Pomocí tohoto API byly získány informace, které byly následně využity pro označení domén jako benigních nebo maligních. [72]

## 3.3 Transformace dat

Vzhledem k povaze a komplexnosti dat internetových domén je klíčové zvolit správné transformace dat, které umožní efektivní klasifikaci. Následuje několik navrhovaných transformací a metod, které by mohly být použity:

- **Normalizace:** Mnoho atributů, jako jsou počty DNS záznamů nebo délky IP prefixů, může mít různé rozsahy hodnot. Normalizace těchto hodnot pomůže zlepšit výkon klasifikačních algoritmů.
- **Kódování kategoriálních proměnných:** Některé atributy, jako jsou typy DNS záznamů, jsou kategoriální. Použití technik jako one-hot encoding nebo label encoding může převést tyto atributy na číselný formát vhodný pro strojové učení.
- **Redukce dimenzionality:** Vzhledem k vysokému počtu atributů může být užitečné aplikovat techniky jako hlavní komponentní analýza (PCA) pro redukci dimenzionality a zvýraznění nejvýznamnějších vlastností.
- **Vytvoření nových atributů:** Můžeme vytvářet nové atributy (feature engineering) z existujících dat, například výpočet entropie názvů domén, což může být indikátorem phishingové aktivity.

## 3.4 Zpracování dat v projektu FETA

Transformace dat hraje klíčovou roli při přípravě vstupů pro strojové učení, zejména v oblasti detekce maligních domén. Tato část práce vznikla v rámci řešení výzkumného projektu **FETA – Flow-based Encrypted Traffic Analysis** (kód **VJ02010024**), který je financován Ministerstvem vnitra ČR. Projekt se zaměřuje na vývoj metod detekce hrozeb ve šifrovaném síťovém provozu pomocí strojového učení a analýzy síťových toků.

V této sekci jsou představeny tři významné přístupy které byly využity v rámci řešení projektu: systém A. Horáka, P. Pouče a O. Ondryáše. Všechny přístupy se zaměřují na optimalizaci sběru, přípravy a transformace dat, přičemž využívají pokročilé techniky pro zvýšení přesnosti klasifikačních modelů.

### 3.4.1 Systém A. Horáka

V rámci svého výzkumu A. Horák [37] navrhl sofistikovaný systém zpracování a transformace dat pro klasifikaci maligních domén. Jeho přístup zdůrazňuje důležitost detailního výběru a transformace dat v kontextu strojového učení, což je klíčové pro účinnou detekci a analýzu škodlivých aktivit. Hlavními charakteristikami Horákova systému zpracování dat jsou:

1. **Sběr dat:** Horák se zaměřuje na shromažďování široké škály datových atributů, včetně DNS záznamů, RDAP dotazů a technických metadata spojených s doménami, čímž vytváří komplexní základ pro následnou analýzu.
2. **Příprava dat:** Data jsou pečlivě připravena a zpracována, což zahrnuje normalizaci hodnot, kódování kategoriálních proměnných a redukci dimenzionality, což zajistí, že jsou data správně formátována pro strojové učení.
3. **Vytvoření atributů:** Horák používá techniky pro vytváření nových atributů (feature engineering), například výpočet entropie názvů domén, což umožňuje identifikovat potenciálně škodlivé vzorce.

Výhody tohoto přístupu:

- **Komplexnost:** Systém umožňuje komplexní analýzu díky shromažďování různorodých dat, což vede k detailnějšímu pohledu na charakteristiky domén.

- **Přesnost:** Pečlivá příprava a transformace dat zlepšují přesnost klasifikačních modelů.
- **Adaptabilita:** Navržený systém je flexibilní a může se přizpůsobit různým výzvám v oblasti detekce maligních domén.

Horákův přístup představuje důležitý základ pro další výzkum a vývoj metod v oblasti kybernetické bezpečnosti. Následující části kapitoly budou rozvíjet Horákův systém a zkoumat další pokročilé metody zpracování dat, které by mohly dále vylepšit efektivitu a přesnost v detekci maligních domén [37]

### 3.4.2 Systém P. Pouče

Petr Pouč ve své diplomové práci [72] představil pokročilý systém optimalizace klasifikačních modelů zaměřený na detekci maligních domén. Jeho přístup se zaměřuje na řešení problémů spojených s nerovnováhou datových sad, výběrem příznaků a optimalizací hyperparametrů, čímž zajišťuje vysokou přesnost a spolehlivost klasifikace.

Hlavními charakteristikami Poučova systému jsou:

1. **Tvorba ground-truth datových sad:** Proces zahrnoval vytvoření vysoce kvalitních datových sad pomocí specializované verifikační pipeline. Tato pipeline kombinovala metody automatizované detekce a ručního přezkumu, aby zajistila, že data přesně reprezentují škodlivé a benigní domény.
2. **Předzpracování dat:** Systém využívá pokročilé techniky pro čištění dat, včetně detekce a odstraňování anomálií, normalizace hodnot a řešení problémů s chybějícími hodnotami.
3. **Optimalizace klasifikace:** Důraz byl kladen na výběr nejdůležitějších příznaků pomocí metod, jako je analýza SHAP, na přizpůsobení modelů prostřednictvím optimalizace hyperparametrů.

#### Výhody Poučova přístupu

Poučův systém nabízí několik klíčových výhod:

- **Robustnost:** Díky optimalizaci dat a modelů dosahuje vysoké odolnosti vůči falešně pozitivním i falešně negativním klasifikacím.
- **Přizpůsobivost:** Systém lze snadno upravit pro různé typy dat a specifické požadavky v oblasti kybernetické bezpečnosti.
- **Praktická použitelnost:** Experimenty ukázaly, že optimalizovaný systém dosahuje skvělých výsledků, například přesnosti skóre F1 až 0,9926 a snížení míry falešně pozitivních výsledků.

Poučova práce staví na metodách zavedených v projektu FETA, což mu umožnilo rozšířit stávající strategie a navrhnout nové postupy, které mohou být přínosem pro detekci škodlivých domén v reálném prostředí.

### 3.4.3 Systém O. Ondryáše

O. Ondryáš [67] se zaměřil na škálovatelný návrh komponent pro **sběr a extrakci příznaků** ve vysokozátěžových sítích. Jeho přístup přináší architektonická vylepšení předešlých systémů v několika směrech:

- **Distribuovaný sběr dat:** Sběr doménových údajů z více zdrojů (DNS, RDAP, TLS, reputační systémy) s využitím systému `Apache Kafka`.
- **Sběrač a kolektory:** Modulární komponenty pro automatizovaný sběr a obohacování dat v reálném čase.
- **Extrakce příznaků:** Transformace nasbíraných dat do vektoru příznaků o 176 prvcích použitelných pro klasifikaci.

Navržený systém prokázal svou efektivitu jak při přípravě trénovacích sad, tak při reálném nasazení v akademické síti CESNET, kde dosáhl propustnosti až 28 doménových jmen za sekundu při paralelním běhu několika instancí.

# Kapitola 4

# Klasifikace internetových domén metodami strojového učení

Kybernetická bezpečnost čelí neustále se vyvíjejícím hrozbám, které zahrnují phishingové útoky, šíření malwaru a řízení botnetů pomocí maligních domén. V tomto dynamickém prostředí je nutné neustále zdokonalovat přístupy k detekci škodlivých aktivit. Strojové učení se v posledních letech stalo klíčovým nástrojem v této oblasti, protože umožňuje analýzu rozsáhlých dat a identifikaci komplexních vzorců, které tradiční metody nemohou efektivně zpracovat.

Tato kapitola se zaměřuje na klasifikaci domén pomocí metod strojového učení, počínaje obecnou diskusí o technikách strojového učení a jejich přínosu v oblasti detekce maligních domén. Následně budou podrobně popsány jak základní, tak pokročilé metody klasifikace, včetně přístupů využívajících hluboké neuronové sítě, Support Vector Machines (SVM) a boostingové algoritmy jako XGBoost. Kapitola také rozebere problematiku dimenzionality dat, která je klíčová při práci s rozsáhlými datovými sadami.

Cílem této kapitoly je poskytnout přehled metod používaných k klasifikaci domén a zdůraznit jejich výhody i omezení v kontextu rychle se měnícího prostředí kybernetických hrozeb. Tato analýza má za cíl připravit základ pro další kapitoly, které se budou zabývat implementací konkrétních metod a jejich aplikacemi na reálná data.

## 4.1 Naivní Metody Klasifikace Maligních Domén

Přestože v oblasti kybernetické bezpečnosti je stále více upřednostňováno použití metod strojového učení, existují také tradiční "naivní"metody pro klasifikaci maligních domén. Tyto metody se spoléhají na jednodušší statistické nebo pravidlově založené přístupy. Mezi tyto metody patří například bayesovský klasifikátor a různé statistické analýzy. Tyto metody mohou být v některých případech efektivní, ale obecně trpí určitými omezeními, zejména v kontextu neustále se měnících vzorců maligních domén.

### 4.1.1 Bayesovský Klasifikátor

Bayesovský klasifikátor je založen na aplikaci Bayesovy teorie pravděpodobnosti. Umožňuje klasifikovat domény na základě pravděpodobnosti jejich příslušnosti k určité kategorii, například maligní versus benigní. Přestože bayesovské klasifikátory byly úspěšně použity v mnoha aplikacích, jejich efektivita může být omezená v případě, že distribuce charakteristik domén se rychle mění [62].

### 4.1.2 Statistické Analýzy

Statistické analýzy, včetně metody detekce outlierů a analýzy četnosti určitých charakteristik, jsou dalšími běžně používanými technikami. Tyto metody mohou být efektivní při identifikaci domén s neobvyklými vlastnostmi, které se vymykají obecným trendům. Nicméně, jejich schopnost adaptace na nové a sofistikované typy útoků může být omezená [9].

### 4.1.3 Omezení Naivních Metod

Hlavní nevýhodou těchto naivních metod je jejich omezená schopnost přizpůsobit se proměnlivé povaze kybernetických hrozeb. Maligní domény a techniky používané útočníky se neustále vyvíjejí, což může způsobit, že statické modely založené na předem definovaných pravidlech nebo historických datech rychle zastarají. Například, bayesovský klasifikátor může mít potíže při rozpoznávání nových vzorců chování, které nebyly zahrnuty v původní trénovací datové sadě [28].

Kromě toho, vysoká míra variability v charakteristikách maligních domén může vést k chybným klasifikacím a vysokému počtu falešně pozitivních nebo falešně negativních výsledků. Toto je zvláště problematické v kontextu dynamického prostředí internetových hrozeb, kde se mohou objevit zcela nové typy útoků nebo sofistikované obcházení existujících detekčních metod [14].

Zatímco naivní metody mohou nabídnout určitý úvodní vhled do detekce maligních domén, jejich omezení v adaptabilitě a přesnosti činí je méně vhodnými pro použití v dynamickém a neustále se vyvíjejícím prostředí kybernetické bezpečnosti. V důsledku toho je důležité směřovat k pokročilejším metodám, jako jsou techniky strojového učení, které nabízejí lepší schopnost učení se z nových dat a adaptace na proměnlivé vzorce útoků.

## 4.2 Dimenzionalita dat

V současné době se kybernetická bezpečnost stále více spoléhá na metody strojového učení pro klasifikaci a identifikaci maligních domén. S rostoucím objemem a složitostí internetových dat se stává nezbytným využití metod schopných zpracovávat data vysoké dimenze. Tyto metody jsou klíčové pro efektivní analýzu a rozpoznání vzorců, které mohou být ukryty v rozsáhlých a komplexních datových sadách [27].

Data vysoké dimenze představují významnou výzvu pro tradiční analytické techniky. "Prokletí dimenzionality"popisuje problémy, které vznikají, když se modely snaží interpretovat data s velkým počtem proměnných. Tyto problémy zahrnují nadměrné učení, kde modely příliš specificky reagují na trénovací data a ztrácí schopnost generalizace [12].

Pro řešení těchto problémů je třeba vybrat algoritmy strojového učení, které jsou robustní vůči vysoké dimenzionalitě dat. Tyto algoritmy musí být schopné efektivně redukovat dimenze, zachovat relevantní informace a zároveň minimalizovat riziko nadměrného učení [34].

### 4.2.1 Vhodné Metody Strojového Učení

V rámci strojového učení jsou některé algoritmy obzvláště vhodné pro zpracování dat vysoké dimenze, což je klíčové pro detekci maligních domén [33]. Mezi tyto metody patří neuronové sítě a konvoluční neuronové sítě (CNN), Support Vector Machines (SVM), rozhodovací

stromy, XGBoost a AdaBoost. Každý z těchto algoritmů má specifické vlastnosti, které je činí vhodnými pro zpracování složitých dat.

### Neuronové sítě

Neuronové sítě, zejména jejich hluboké architektury jako konvoluční neuronové sítě (CNN), hrají významnou roli v moderním strojovém učení. Jsou schopny modelovat složité nelineární vztahy a zachytit komplexní vzory i ve vysoce dimenzionálních datech, což z nich činí vhodné kandidáty pro úlohy detekce anomálií nebo škodlivého chování [53].

Jejich předností je vysoká míra expresivity a schopnost automatické extrakce relevantních charakteristik z dat. Nevýhodou je však náročnost na trénovací data, výpočetní prostředky a náchylnost k přeučení v malých nebo nevyvážených datech [32].

### Support Vector Machines (SVM)

Metoda podpůrných vektorů je klasický a stále velmi robustní přístup k binární klasifikaci, známý svou schopností nalézat optimální rozhodovací hranici i v datech s vysokou dimenzionalitou [25].

Hlavní výhodou SVM je jeho matematická čistota, stabilita a schopnost generalizace i při menším množství trénovacích vzorků. Mezi nevýhody patří citlivost na výběr jádra a potřeba pečlivého ladění hyperparametrů, zejména při práci s většími datovými sadami [94].

### Rozhodovací stromy

Rozhodovací stromy patří mezi nejlépe interpretovatelné modely strojového učení. Díky své stromové struktuře umožňují snadnou vizualizaci rozhodovací logiky a práci s heterogenními daty [77].

Mezi jejich výhody patří jednoduchost, rychlost a přirozená schopnost pracovat s chybějícími hodnotami. Jejich slabinou je však sklon k přeučení, zejména pokud nejsou správně ořezány, a omezená schopnost modelovat složité vzory v datech [57].

### Boostingové algoritmy (XGBoost, AdaBoost)

Boostingové metody, jako jsou AdaBoost nebo XGBoost, představují výkonný přístup ke skládání slabších klasifikátorů (např. rozhodovacích stromů) do silného modelu.

Výhodou těchto metod je vysoká přesnost, schopnost zvládat různé typy dat a relativní odolnost vůči přeučení. XGBoost je navíc optimalizován pro rychlý paralelní běh a efektivní správu paměti. Nevýhodou je vyšší složitost ladění, náchylnost ke zpracování šumu a nižší interpretovatelnost výsledného modelu [29, 18].

Výběr vhodné metody strojového učení pro detekci maligních domén závisí na specifických požadavcích a charakteristikách dat. Neuronové sítě a CNN jsou vhodné pro složité vzory a velké množství dat, ale vyžadují značný výpočetní výkon. SVM jsou užitečné pro efektivní zpracování dat s vysokou dimenzionalitou, ale vyžadují pečlivé nastavení. Rozhodovací stromy jsou uživatelsky přívětivé a snadno interpretovatelné, ale mohou mít problémy s nadměrným učením.

Každá z těchto metod nabízí různé přístupy k řešení problému detekce maligních domén, což umožňuje adaptaci na specifické potřeby a omezení dané situace. Neuronové sítě a CNN jsou ideální pro složité úlohy, kde je potřeba modelovat vysokou úroveň abstrakce,

zatímco SVM a rozhodovací stromy nabízejí více přímou a interpretovatelnou cestu. XGBoost a AdaBoost jsou efektivní v situacích, kde je potřeba zvýšit přesnost slabších modelů prostřednictvím iterativního vylepšování. Volba správné metody je tedy klíčová pro úspěch v detekci a klasifikaci maligních domén v rámci rychle se měnícího kybernetického prostředí [32, 25, 18].

## 4.3 Neuronové sítě

Neuronové sítě excelují v abstrahování nad velkým množstvím dat, hledáním souvislostí a vzorců, které nejsou přímo zřejmé [53].

Tato sekce se zaměřuje na vysvětlení základních principů neuronových sítí, jejich struktury a funkčnosti. Dále jsou zde popsány různé typy neuronových sítí, které jsou relevantní pro účely detekce maligních domén, včetně konvolučních neuronových sítí (CNN) [54], rekurentních neuronových sítí (RNN) [32], a LSTM sítí [36]. Speciální pozornost je věnována způsobům, jakými neuronové sítě zpracovávají a klasifikují data, což je klíčové pro pochopení jejich efektivity v identifikaci a rozlišování maligních domén od legitimních [89].

### 4.3.1 Komponenty a architektura neuronových sítí

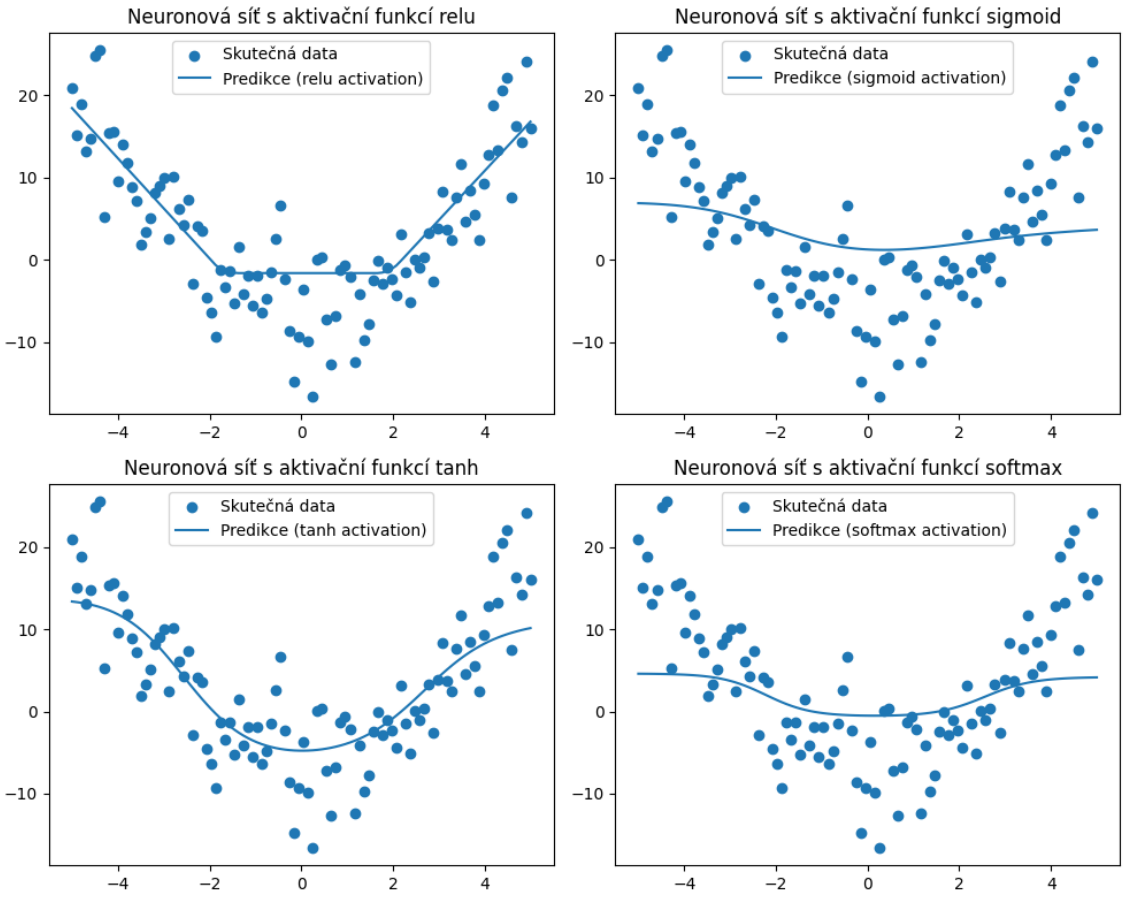
Neuronové sítě jsou složeny z mnoha různých komponent, které společně definují jejich schopnost učení a klasifikace. Klíčové komponenty zahrnují:

- **Neurony:** Základní jednotky neuronových sítí, které přijímají a zpracovávají vstupní data [32].
- **Vrstvy:** Skupiny neuronů organizované do vrstev, včetně vstupní, skryté a výstupní vrstvy [53].
- **Aktivační funkce:** Funkce používané k určení výstupu neuronu na základě jeho vstupů, běžně ReLU, sigmoid nebo softmax [32].
- **Zpětná propagace a učení:** Proces, při kterém se síť učí a upravuje své váhy pro zlepšení předpovědí, založený na gradientním sestupu [32].

### 4.3.2 Aktivační funkce

Aktivační funkce hrají klíčovou roli v neuronových sítích, neboť určují výstupní hodnoty neuronů a tím ovlivňují celkovou schopnost sítě učit se a provádět klasifikaci. Různé aktivační funkce mohou mít dramatický vliv na výkon a konvergenci tréninkového procesu, jak je ilustrováno na Obrázku: 4.1 [6].

- **ReLU (Rectified Linear Unit)**: Nejčastěji používaná aktivační funkce ve skrytých vrstvách neuronových sítí kvůli své efektivitě a jednoduchosti. Její matematický popis je následující:
$$f(x) = \max(0, x) \tag{4.1}$$
Tato funkce umožňuje modelu rychle konvergovat a snižuje pravděpodobnost gradientní degradace při dlouhém tréninku.




Obrázek 4.1: Vliv aktivačních funkcí na klasifikaci

- **Sigmoid**: Funkce sigmoid je tradiční volba pro výstupní vrstvy binární klasifikace, protože její výstupy jsou omezeny na interval (0,1). Matematický popis funkce je:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{4.2}$$

  Výstupy této funkce lze interpretovat jako pravděpodobnost.

- **Tanh (Hyperbolický Tangens)**: Funkce tanh je podobná sigmoidální funkci, ale její výstupy jsou normalizovány na rozsah (-1,1), což může vést k rychlejší konvergenci během tréninku. Matematický popis:

$$f(x) = \tanh(x) \tag{4.3}$$

  Tato normalizace může vést k lepšímu škálování gradientů.

- **Softmax**: Funkce softmax je rozšíření sigmoidní funkce pro více tříd a je často používána v poslední vrstvě neuronových sítí pro více-třídní klasifikaci. Matematický popis funkce je:

$$\text{Softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{4.4}$$

kde $\mathbf{z}$ představuje vstupní vektor a $K$ je počet tříd. Tato funkce vypočítává pravděpodobnosti tříd tak, že exponentně normalizuje výstupy vrstvy.

Výběr správné aktivační funkce závisí na specifickém úkolu a distribuci dat. Jak je vidět na obrázku 4.1, různé aktivační funkce mohou vést k odlišným výsledkům i pro stejná vstupní data, což zdůrazňuje jejich význam při návrhu architektury neuronové sítě.

### 4.3.3 Architektura neuronových sítí

Architektura neuronové sítě odkazuje na způsob, jakým jsou její komponenty uspořádány a propojeny. Tato část se zaměřuje na:

- **Struktura a design:** Způsoby, jakými lze neurony a vrstvy organizovat pro různé typy úloh.
- **Výběr a nastavení vrstev:** Rozhodování o počtu a typech vrstev pro specifické aplikace.
- **Konvoluční neuronové sítě (CNN):** Specifika a aplikace CNN, zvláště v oblasti zpracování obrazu a vizuálního rozpoznávání.
- **Rekurentní neuronové sítě (RNN):** Význam a využití RNN v kontextech, kde jsou důležité časové sekvence a vzory.

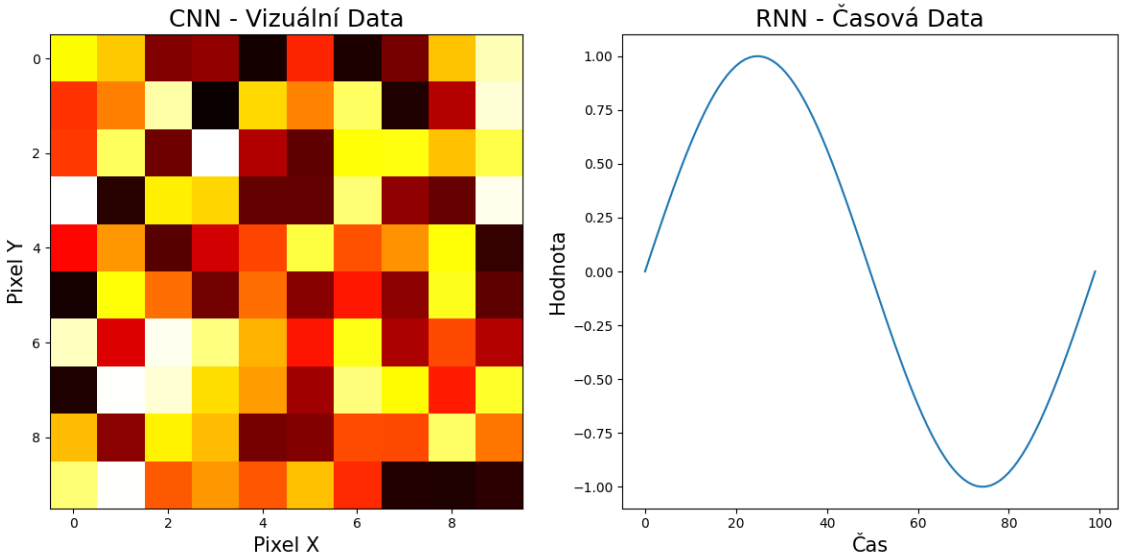
### 4.3.4 Použití a vhodnost různých architektur neuronových sítí

Výběr vhodné architektury neuronové sítě je zásadní pro úspěšnou detekci maligních domén. Konvoluční neuronové sítě (CNN) jsou obzvláště užitečné v případech, kdy je potřeba extrahovat a identifikovat vzory z vizuálních dat, díky své schopnosti efektivně zpracovávat a klasifikovat obrazy [6]. Na druhou stranu, rekurentní neuronové sítě (RNN) a jejich varianty, jako je LSTM (Long Short-Term Memory), jsou vhodnější pro úlohy, kde je důležitá časová posloupnost dat, jako je analýza časových řad nebo přirozeného jazyka [92]. V kontextu detekce maligních domén může být kombinace obou těchto typů architektur efektivní, například použití CNN pro identifikaci vizuálních vzorců v rámci webových stránek a RNN pro analýzu sekvenčního chování uživatelů nebo komunikace serverů. Další variantou rekurentních neuronových sítí jsou Gated Recurrent Units (GRU), které poskytují efektivní a méně výpočetně náročnou alternativu k LSTM. GRU dokáží zachytit dlouhodobé závislosti v sekvenčních datech, což je klíčové pro detekci změn a anomálií v doménových vzorcích. Díky své jednodušší struktuře a schopnosti minimalizovat problém degratujícího gradientu jsou GRU často preferovány v případech, kdy je nutné rychle a efektivně zpracovávat velké objemy dat[20].

Níže 4.2 se nachází demonstrace rozdílu vstupních dat pro rekurentní a konvoluční neuronové sítě

### 4.3.5 CNN

Konvoluční neuronové sítě (CNN) představují významný pokrok v oblasti strojového učení, zejména pro úkoly zpracování obrazových dat. Jejich schopnost efektivně identifikovat a klasifikovat vzory v datech je zvláště užitečná při analýze dat domén, která často obsahují složité a bohaté informace s obrazovými charakteristikami. Podobně jako v obrazové




Obrázek 4.2: Srovnání CNN a RNN

analýze, kde CNN efektivně detekují a klasifikují různé vizuální prvky, mohou být tyto sítě aplikovány na data domén, aby rozpoznávaly specifické vzory a charakteristiky, které jsou indikativní pro určité kategorie domén. Díky schopnosti zpracovávat velké množství dat a extrahovat z nich klíčové vlastnosti, jsou CNN ideálním nástrojem pro překlenutí komplexity a rozmanitosti v datech domén [1].

Gated Recurrent Units (GRU) jsou efektivní variantou rekurentních neuronových sítí, která kombinuje jednoduchost a schopnost zachytit dlouhodobé závislosti v sekvenčních datech. Oproti LSTM sítím využívají méně parametrů, což snižuje výpočetní náročnost, a přitom zachovávají vysokou schopnost učení. GRU byly představeny jako zjednodušená alternativa k LSTM a ukázaly srovnatelný výkon při nižší složitosti modelu [20].

Tato architektura se ukazuje jako vhodná pro analýzu sekvenčních dat doménových jmen, kde může efektivně identifikovat vzorce, jako jsou neobvyklé kombinace znaků nebo chování generované automaticky (např. DGA domény) [47].

Struktura GRU sítě typicky zahrnuje:

- **Aktualizační a resetovací brány:** Umožňují efektivně řídit tok informací a eliminovat problém zmizelého gradientu.
- **Flexibilita:** Schopnost zachytit jak krátkodobé, tak dlouhodobé závislosti v datech.
- **Nižší výpočetní náročnost:** Menší počet parametrů oproti LSTM umožňuje rychlejší trénink na velkých datových sadách.

### 4.3.6 LSTM

Long Short-Term Memory (LSTM) sítě, speciální typ rekurentních neuronových sítí, jsou zvláště vhodné pro lexikální analýzu doménových jmen z několika důvodů. Prvním klíčovým aspektem je jejich schopnost zachytit dlouhodobé závislosti v sekvenčních datech.

V kontextu doménových jmen může LSTM efektivně identifikovat a učit se ze vzorců v sekvencích znaků, což je zásadní pro rozpoznávání anomálií nebo neobvyklých struktur, které často indikují škodlivé aktivity [36].

Dále LSTM minimalizuje problém zmizelého gradientu, který je běžný u tradičních RNN, díky své unikátní struktuře s bránami pro zapomínání, vstup a výstup. Tato struktura umožňuje efektivnější trénink a lepší generalizaci při analýze složitých nebo zakódovaných jmen domén, kde je potřeba rozlišit legitimní domény od těch maligních. LSTM tak nabízí robustní rámec pro analýzu lexikálních charakteristik doménových jmen, což je klíčové pro včasnou a přesnou detekci potenciálně škodlivých webových aktivit [47].

## 4.4 Metoda podpůrných vektorů (SVM)

Metoda podpůrných vektorů, neboli *Support Vector Machines* (SVM), je klasifikační algoritmus z třídy algoritmů učení s učitelem. SVM bylo původně vyvinuto jako pomocná metoda pro trénování neuronových sítí, ale později se ukázalo jako velmi efektivní samostatná klasifikační metoda [25]. Algoritmus je obzvlášť výhodný při práci s daty o vysoké dimenzi, a to zejména díky schopnosti najít optimální rozhodovací hranici mezi třídami [34].

Základním principem SVM je identifikace hyperroviny, která maximálně separuje klasifikované třídy a zároveň mezi nimi udržuje co největší možnou marginu. Pokud nejsou data lineárně separovatelná v dané dimenzi, SVM využívá tzv. kernelovou transformaci – mapuje vstupní prostor do vyšší dimenze, kde separace možná je [31]. Z matematického hlediska lze téměř vždy najít vyšší dimenzi, ve které budou data separovatelná.

Klasifikátor SVM se sestává z několika klíčových parametrů, které výrazně ovlivňují jeho chování i výsledky klasifikace:

- Regularizační parametr **C**
- Jádrová funkce **kernel** a stupeň **degree**
- Koeficient **gamma**

### Regularizační parametr C

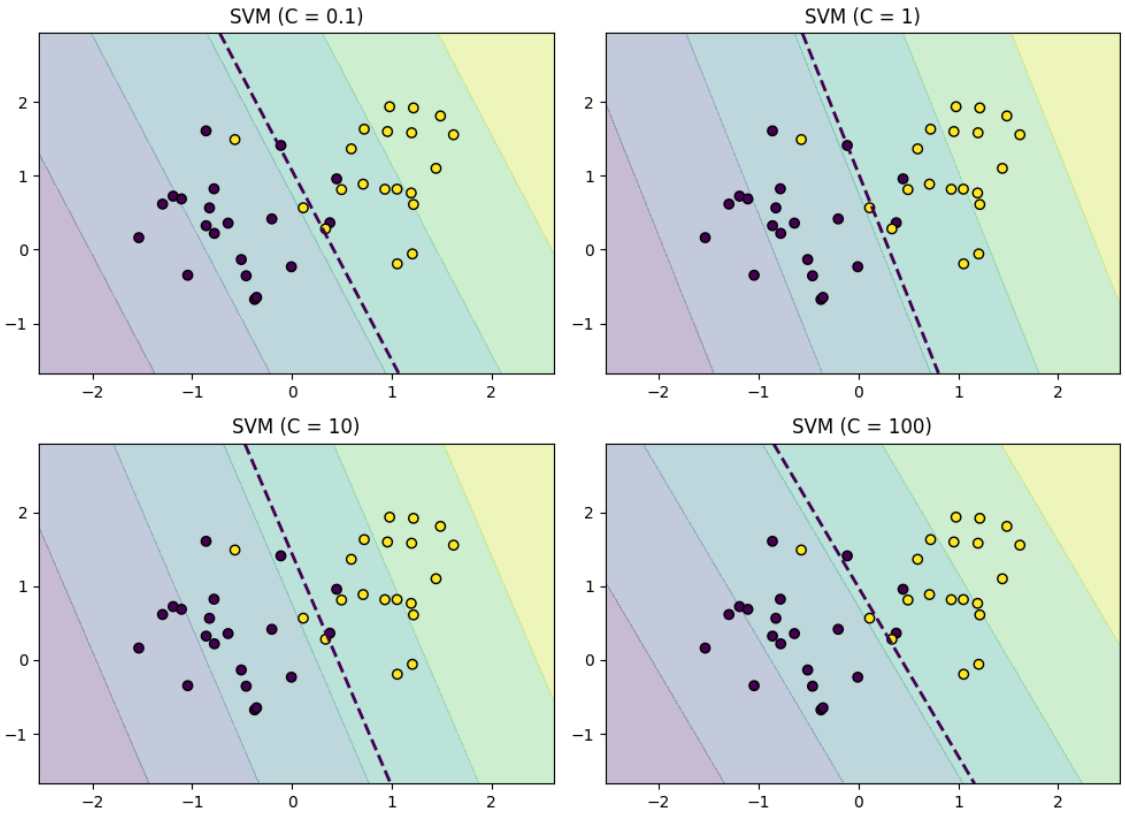
Cílem algoritmu SVM je nalezení co nejrobustnější separace tříd. Parametr **C** představuje kompromis mezi maximalizací okraje a minimalizací chyb klasifikace. Nízké hodnoty **C** vedou k větší toleranci pro chybně klasifikované vzory – model více generalizuje a je robustnější vůči šumu. Naopak vysoké hodnoty **C** kladou důraz na správnou klasifikaci trénovacích dat, což může vést k přeučení (overfitting) [25, 31].

V grafu 4.3 je demonstrován vliv změny tohoto parametru. V případě vysokého koeficientu se metoda snaží zahrnout všechny body a tedy zvýšit přesnost klasifikace, ovšem na úkor schopnosti generalizovat. Opačný případ nastává pro malé hodnoty C.

### Separující hyperrovina - kernel

Pomocí tohoto parametru je možné volit matematický charakter hyperroviny, pomocí které jsou data separována a tedy klasifikována. Tato funkce může být lineárního i nelineárního charakteru. Vizualizace vlivu jádrové funkce je ilustrována na obrázku: 4.4 [31].

Nejpoužívanější funkce jsou pak:



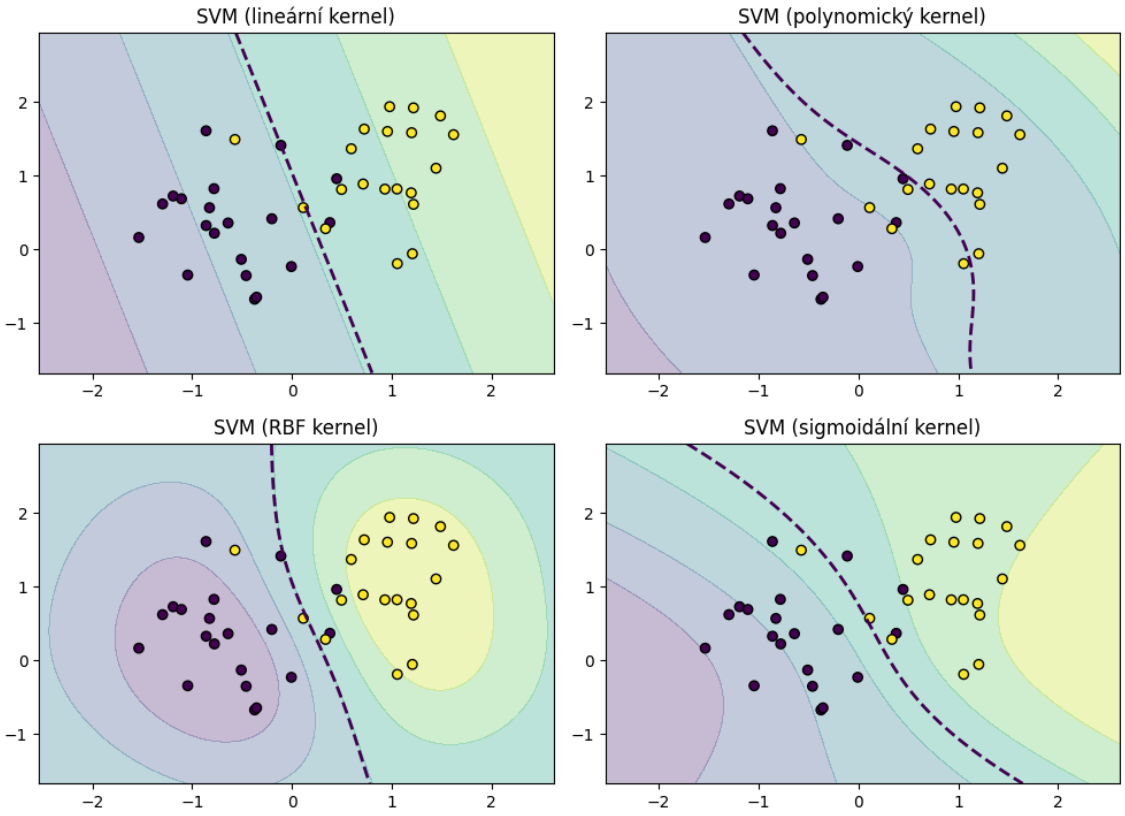

Obrázek 4.3: Vliv koeficientu **C** na klasifikaci SVM

- lineární
- polynomiální
- rbf
- sigmoid

### Koeficient *gamma*

Koeficient gamma ($\gamma$) hraje klíčovou roli v rámci klasifikace pomocí Support Vector Machines (SVM), zejména při použití Gaussiánského (RBF) jádra. Tento parametr určuje míru vlivu jednotlivých trénovacích vzorů na rozhodovací hranici. Čím vyšší je hodnota $\gamma$, tím užší je oblast vlivu jednotlivých vzorů, což vede k tvorbě složitějších, zakřivených hranic [31].

Zvýšením hodnoty $\gamma$ dochází k tomu, že model lépe zachytí lokální vlastnosti dat, což může zlepšit klasifikaci složitých struktur, ale zároveň zvyšuje riziko přetrénování (overfitting). Přetrénovaný model má tendenci přizpůsobit se šumu v datech a selhává při generalizaci na neviděná data. Naopak nižší hodnoty $\gamma$ vedou k hladší a jednodušší rozhodovací

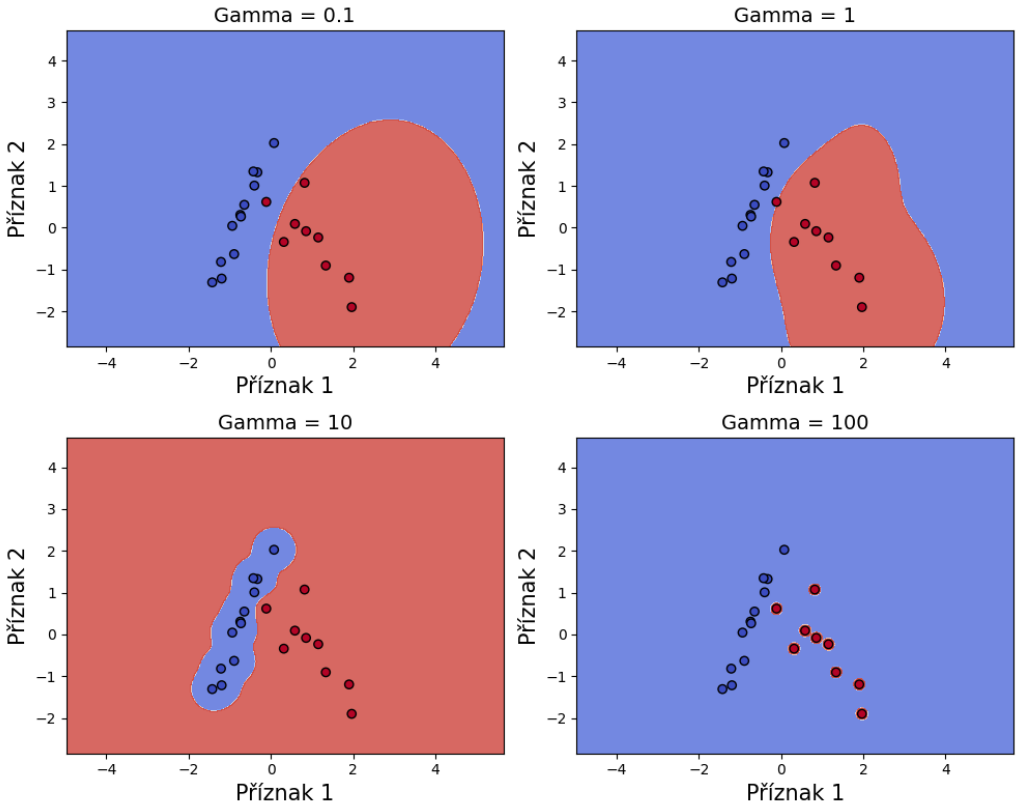



Obrázek 4.4: Vliv výběru jádrové funkce na klasifikaci SVM

hranici, což zvyšuje schopnost modelu generalizovat, avšak může vést k nedostatečné přesnosti [13].

Optimální hodnota parametru $\gamma$ se zpravidla hledá pomocí technik, jako je křížová validace nebo bayesovská optimalizace hyperparametrů. Výběr správné hodnoty závisí na distribuci dat a složitosti klasifikační úlohy [31, 13].

- **Gamma = 0.1:** Na obrázku 4.5 je patrný větší vliv okolních bodů na tvar rozhodovací plochy. Model je méně přizpůsoben trénovacím datům a má tendenci k jednoduššímu modelu.
- **Gamma = 1:** S hodnotou gamma rovno 1 (obrázek 4.5) máme vyvážený přístup, kde vliv trénovacích bodů není příliš vysoký ani příliš nízký. Model je schopen efektivně generalizovat.
- **Gamma = 10:** Zvýšení hodnoty gamma na 10 (obrázek 4.5) vede k tomu, že rozhodovací plocha je silně ovlivněna trénovacími body v jejich bezprostředním okolí. Model se tím stává citlivějším na detaily dat.




Obrázek 4.5: Vliv koeficientu $\gamma$ na klasifikaci SVM

- **Gamma = 100:** Pro velmi vysokou hodnotu gamma (obrázek 4.5) má rozhodovací plocha tendenci procházet jednotlivými trénovacími body, což může vést k přeučení modelu na trénovací data.

V obecnosti platí, že volba optimální hodnoty gamma závisí na konkrétních datech a cílech modelu. Je vhodné provádět ladění hyperparametrů (tj. volbu hodnoty gamma) pomocí validačních dat a sledovat výslednou výkonnost modelu na testovacích datech.

### 4.4.1 Klasifikace pomocí SVM

Algoritmus SVM je široce využíván pro klasifikaci maligních domén, a to zejména díky své schopnosti efektivně pracovat s daty vysoké dimenze [25]. Je obzvlášť vhodný pro úlohy, kde je potřeba nalézt nelineární rozhodovací hranice v komplexních datových strukturách. Překážkou však může být neuniformita vstupních dat – tedy různorodost typů příznaků (např. kombinace číselných, kategoriálních či binárních atributů).

#### GridSearch

Pro nalezení optimálních parametrů metody SVM se běžně používá technika zvaná GridSearch, která systematicky prochází předdefinované kombinace parametrů modelu [13]. Typicky zahrnuje volbu typu jádra (kernel), regularizačního faktoru $C$ a parametru $\gamma$ u RBF

jádra. Pro každou kombinaci parametrů se pomocí křížové validace (cross-validation) vyhodnocuje výkon modelu, často podle metrik jako jsou přesnost, skóre F1 nebo AUC [13, 6]. Přestože je GridSearch výpočetně náročný, jeho použití je zásadní pro dosažení maximální výkonnosti modelu v reálném prostředí.

## 4.5 Stromové algoritmy

Stromové algoritmy se vyznačují schopností reprezentovat a modelovat rozhodovací procesy pomocí hierarchické struktury stromu, která reflektuje logiku rozhodování. Tyto algoritmy mají výhodu v interpretovatelnosti a snadné vizualizaci, což je zásadní pro pochopení mechanismů, jež vedou k jejich klasifikačním rozhodnutím [77]. V kontextu detekce maligních domén, kde rychlá a spolehlivá klasifikace může znamenat rozdíl mezi úspěšnou prevencí a bezpečnostním rizikem, je důležité zhodnotit, jak efektivně tyto algoritmy identifikují potenciálně nebezpečné online entity [84, 97].

Dále budou detailněji představeny nejpoužívanější stromové algoritmy, jako jsou například rozhodovací stromy (Decision Trees), náhodné lesy (Random Forests) či gradient boosting algoritmy. Zaměření bude následovat na porovnání těchto algoritmů v kontextu ostatních metod klasifikace a objektivní srovnání jejich účinnosti v rámci konkrétního bezpečnostního kontextu [15, 37, 49].

### Rozhodovací stromy - Decision Trees

Rozhodovací stromy patří mezi základní a široce používané klasifikační algoritmy, které pracují na principu rekurzivního dělení datového prostoru na základě hodnot atributů. Výsledkem je struktura ve tvaru stromu, kde každý uzel reprezentuje rozhodovací pravidlo a listy pak odpovídají výsledným třídám [77].

Mezi hlavní přednosti patří jejich vysoká interpretovatelnost a schopnost pracovat s kategoriálními i číselnými atributy. Dokáží modelovat nelineární vztahy a jejich rozhodovací pravidla jsou snadno sledovatelná. Na druhou stranu jsou náchylné k přeučení, zejména pokud nejsou ořezávány nebo je množina trénovacích dat omezená [46].

### Náhodné lesy - Random Forests

Náhodné lesy představují kombinovanou metodu (ensemble), která kombinuje více rozhodovacích stromů za účelem zvýšení přesnosti a robustnosti klasifikace. Každý strom je trénován na náhodném vzorku dat a náhodné podmnožině atributů, což podporuje diverzitu mezi jednotlivými stromy a snižuje přeučení [15].

Hlavní výhodou je vyšší odolnost vůči šumu a schopnost pracovat i s rozsáhlými a nečistými daty. Algoritmus také poskytuje měření důležitosti atributů. Nevýhodou je nižší interpretovatelnost výsledků ve srovnání s jednotlivými stromy a vyšší výpočetní náročnost [55].

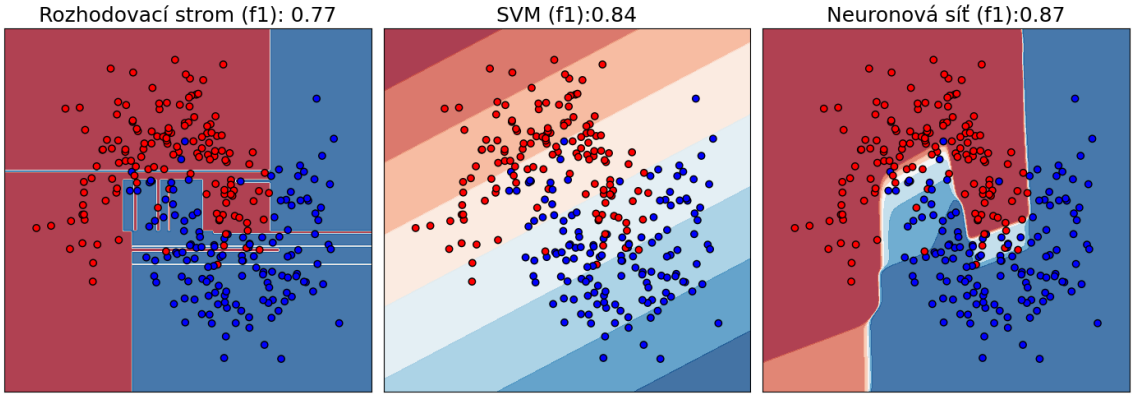
### Algoritmy na bázi Gradient Boosting

Gradient Boosting je výkonná kombinovaná (ensemble) metoda, která buduje finální model iterativním přidáváním stromů, přičemž každý nový strom se snaží minimalizovat chyby předchozího modelu. Patří sem známé implementace jako XGBoost, LightGBM nebo CatBoost [30, 18, 49].

Mezi výhody patří vysoká přesnost, schopnost pracovat s nevyváženými třídami a efektivní zpracování i velkých datasetů. Modely založené na gradient boosting však obvykle vyžadují důkladné ladění hyperparametrů a jejich interpretace bývá složitější ve srovnání s jednoduššími stromovými metodami.
Další část práce bude zaměřena na konkrétní příklady a aplikace těchto stromových algoritmů v oblasti detekce maligních domén, bude poskytnut přehled o jejich úspěšnosti a efektivitě v reálných podmínkách. Budou také diskutovány jejich limitace a možné další využití [84, 97].

## 4.6 Složené klasifikátory

Kombinování různých metod strojového učení, známé také jako *ensemble learning*, poskytuje robustní řešení pro zvýšení přesnosti a spolehlivosti klasifikačních modelů. Složené klasifikátory využívají výhod různorodých prediktivních modelů tím, že integrují jejich předpovědi a přispívají k vytvoření silnějšího a více generalizovaného konečného modelu [26, 93]. Tento přístup může být obzvláště účinný v případech, kde jednotlivé modely jsou silné na různých typech dat nebo aspektech úlohy klasifikace. V oblasti kybernetické bezpečnosti jsou kombinované (ensemble) metody často využívány ke kombinaci výstupů různých klasifikátorů, jako jsou SVM, rozhodovací stromy nebo neuronové sítě, s cílem zvýšit robustnost proti chybné detekci [10].




Obrázek 4.6: Rozdíly klasifikačních metod

Na obrázku 4.6 jsou zobrazeny rozhodovací hranice a skóre F1 pro tři různé klasifikační metody: rozhodovací strom, SVM a neuronovou síť. Rozhodovací strom, s skóre F1 0.77, je schopen zachytit nelineární vzory v datech, což je patrné z komplexní struktury jeho rozhodovací hranice. Tato metoda je výhodná, pokud jsou data nelineárně separovatelná a obsahují rozhodovací pravidla, která mohou být reprezentována stromovými strukturami. Na druhé straně, SVM dosahuje skóre F1 0.84, což ukazuje na jeho schopnost efektivně najít optimální rozdělující hranici, zejména v případě, že data jsou lineárně nebo téměř lineárně separovatelná. Nakonec neuronová síť, s nejvyšším skóre F1 0.87, demonstruje svou schopnost vytvořit složité nelineární rozhodovací hranice díky použití aktivačních funkcí, což ji činí vhodnou pro složitější vzorce v datech, které mohou být pro ostatní metody obtížně rozpoznatelné.

### 4.6.1 Metody kombinování klasifikátorů

Kombinování klasifikátorů je klíčovou strategií pro zvýšení přesnosti a spolehlivosti modelů strojového učení. Tato sekce detailně rozebírá tři hlavní přístupy: Bagging, Boosting a Stacking.

#### Bagging

Bagging (bootstrap aggregating) zvyšuje stabilitu a přesnost modelů tím, že kombinuje predikce více modelů, které byly trénovány na různých náhodných podmnožinách dat. Tento přístup snižuje variabilitu a zamezuje přeučení, což je zvláště užitečné pro modely citlivé na změny v trénovacích datech, jako jsou rozhodovací stromy. Nejznámějším příkladem baggingu je Random Forest, který trénuje velké množství rozhodovacích stromů na různých vzorcích a kombinuje jejich výstupy prostřednictvím hlasování.

**Použití:** Bagging je ideální pro situace, kdy je cílem zvýšit robustnost modelů na šumových nebo nerovnoměrně rozložených datech. Například v oblastech, jako je predikce zdravotních rizik nebo detekce anomálií, kde je stabilita modelu klíčová.

**Literatura:** Breiman, L. (2001). Random Forests. *Machine Learning, 45*(1), 5–32. [15] Tato studie představuje Random Forest jako efektivní aplikaci baggingu, která přináší vysokou přesnost při zachování jednoduchosti.

#### Boosting

Boosting iterativně kombinuje slabé klasifikátory, jako jsou malé rozhodovací stromy, aby vytvořil silný model. Každý nový klasifikátor se zaměřuje na chyby předchozích modelů tím, že přiděluje vyšší váhy těm vzorkům, které byly klasifikovány nesprávně. Tím se postupně snižuje chyba modelu. AdaBoost a Gradient Boosting patří mezi nejčastěji používané algoritmy boostingu.

**Použití:** Boosting se často používá tam, kde je vyžadována vysoká přesnost a nízká míra chyb, například v lékařské diagnostice nebo v ekonomických predikcích. Je také oblíbený v soutěžích v oblasti strojového učení.

**Literatura:** Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences, 55*(1), 119–139. [29] Tato práce představuje algoritmus AdaBoost jako základní metodu boostingu, která iterativně zlepšuje výkon modelu.

#### Stacking

Stacking kombinuje výstupy různých modelů prostřednictvím meta-modelu, který se učí optimálně kombinovat predikce jednotlivých základních modelů. Tento přístup umožňuje využít různé silné stránky jednotlivých modelů a zlepšit celkovou generalizaci.

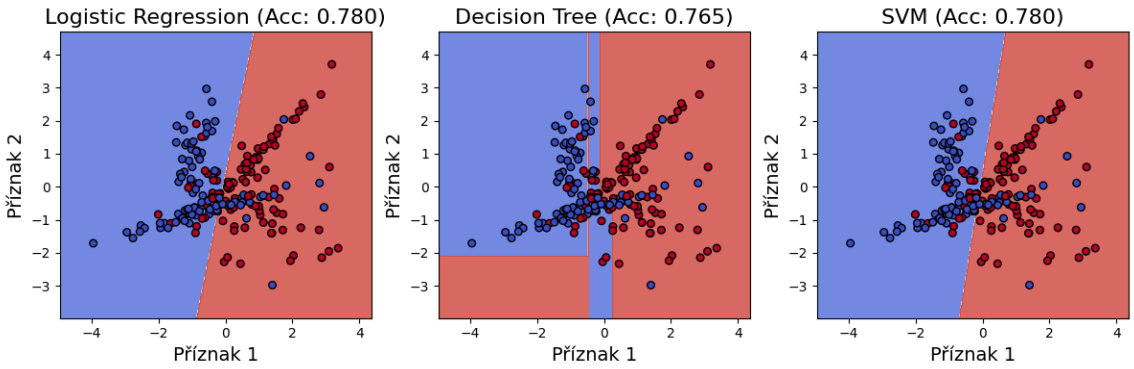
**Použití:** Stacking je oblíbený při analýze komplexních dat, kde různé modely poskytují různé pohledy. Typickými oblastmi použití jsou predikce v biologii, analýza trhu nebo zpracování přirozeného jazyka.

**Literatura:** Wolpert, D. H. (1992). Stacked generalization. *Neural Networks, 5*(2), 241–259. [93] Tento článek popisuje stacking jako metodu, která kombinuje různé klasifikátory pomocí meta-modelu, což výrazně zlepšuje přesnost a generalizaci.

### 4.6.2 Vizualizace metod a efekt skládání

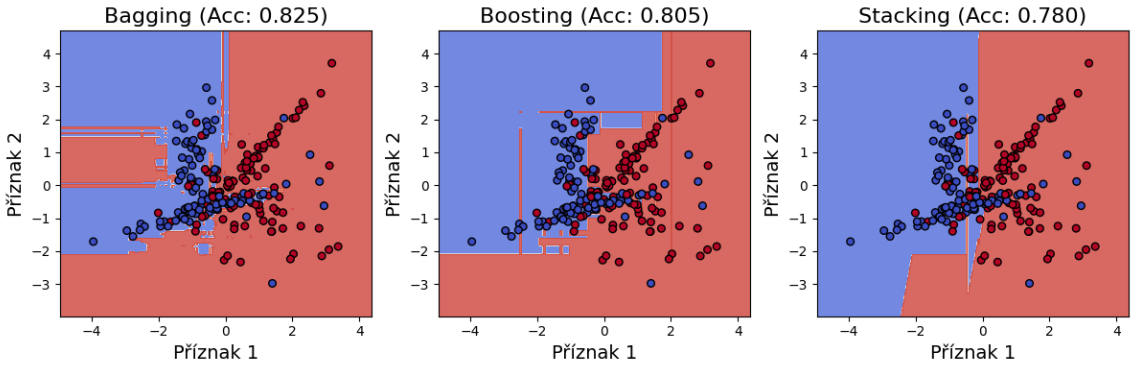
Následující demonstrace zobrazuje rozdíl v úspěšnosti klasifikace pomocí referenčních klasifikátorů a klasifikátorů vytvořených skládáním.

Pro tuto demonstraci byla vygenerována datová sada 1000 2D bodů s použitím python knihovny *sklearn*. Jednotlivé příznaky jsou dále referencovány jako příznak 1 a 2.




Obrázek 4.7: Klasifikace pomocí individuálních metod (LR, SVM, DT).

Obrázek 4.7 ukazuje rozhodovací hranice a přesnost základních modelů, které mohou být omezené při pokrývání komplexních nelineárních vzorů.




Obrázek 4.8: Klasifikace po aplikaci složených metod (Bagging, Boosting, Stacking).

Obrázek 4.8 zobrazuje zlepšení dosažené kombinací metod. Složené metody kompenzují chyby jednotlivých klasifikátorů a vytvářejí robustnější rozhodovací hranice, což vede ke zvýšení přesnosti.

Na základě analýzy metod kombinování klasifikátorů lze zobecnit jejich hlavní výhody a oblasti použití:

- **Zvýšení přesnosti**: Kombinace predikcí různých modelů vede k vyšší přesnosti, než kterou je schopný dosáhnout jednotlivý model samostatně. Tento efekt je způsoben tím, že různé modely zachytí různé aspekty dat, což je klíčové pro dosažení robustních výsledků [4].

- **Zlepšení generalizace**: Složené modely jsou schopny redukovat přeučení díky tomu, že chyby jednotlivých modelů se vzájemně kompenzují. Tato vlastnost zvyšuje schopnost modelů generalizovat na neviděných datech, což je zásadní zejména u složitých úloh [81].
- **Adaptabilita k různým typům dat**: Díky kombinaci různorodých algoritmů dokáží složené klasifikátory lépe zvládat různé charakteristiky dat, například data s šumem, komplexní nelineární vzory nebo nerovnoměrné rozdělení tříd [48].

Složené metody kombinování klasifikátorů tak představují efektivní nástroj pro řešení široké škály úloh strojového učení, od klasifikace po predikci a detekci anomálií.

# Kapitola 5

# Datová sada a sběr

Tato kapitola se věnuje původu datových sad a jejich následnému zpracování v kontextu této diplomové práce. Je zde navázáno na výzkum projektu FETA. V rámci této diplomové práce jsou navrženy dodatečné transformace a úprava formátu datových sad, pro efektivnější použití při trénování modelů strojového učení. První část se zaměřuje na popis charakteristiky a původu datových sad a ve druhé části jsou navrženy dodatečné metody zpracování a je zde demonstrován jejich přínos.

## 5.1 Datová sada

V rámci této diplomové práce byla využita datová sada, kterou jsem vytvořil společně se spoluautory článku připravovaného k publikaci v časopise *Data in Brief* pod názvem *A Multi-Dimensional DNS Domain Intelligence Dataset for Cybersecurity Research*. Sada je ve formátu JSON a představuje export databáze z MongoDB, kde jsou uchovávána data nasbíraná nástrojem `DomainRadar` [1] [42]. Podrobnosti o struktuře a obsahu dat jsou uvedeny v detailu publikovaného datasetu na platformě *Zenodo*: [2].

Datová sada obsahuje více než jeden milion domén, které jsou detailně anotovány jako benigní, phishingové nebo obsahující malware. Klíčovou vlastností této sady je její **multi-dimenzionální struktura**, která zahrnuje data z následujících zdrojů:

- **DNS záznamy:** Informace o doménových jménech, jako jsou A, AAAA, MX nebo NS záznamy.
- **TLS certifikáty:** Detailní informace z TLS handshake a certifikátů.
- **RDAP a WHOIS:** Metadata o registraci domén.
- **Geolokační data:** Informace o poloze IP adres získané z databáze GeoLite2.
- **Reputační data:** Údaje o doménách a IP adresách z platformy VirusTotal a dalších zdrojů.

Datová sada je formátována ve formátu JSON, což usnadňuje její integraci do analytických nástrojů a využití ve strojovém učení.

[1] https://github.com/nesfit/domainradar-dib
[2] https://zenodo.org/records/13330073

### 5.1.1 Deskriptivní statistiky datové sady

Tabulka 5.1 poskytuje přehled o základní struktuře datové sady, včetně počtu domén v jednotlivých kategoriích a zdrojů dat.

| Kategorie | Počet domén | Zdroj |
|---|---|---|
| Benigní (Cisco Umbrella) | 368,956 | Cisco Umbrella Top 1 Million |
| Benigní (CESNET) | 461,338 | akademická síť CESNET |
| Phishingové | 164,425 | PhishTank, OpenPhish |
| Malware | 100,809 | ThreatFox, URLHaus, další černé listiny |

Tabulka 5.1: Přehled jednotlivých kategorií a jejich zdrojů v datové sadě.

#### Datová sada phishing domén

Datová sada phishingových domén, sbíraná v průběhu roku 2023 a 2024, představuje komplexní sadu dat určenou k identifikaci phishingových útoků. Tato datová sada obsahuje následující charakteristiky:

Phishingové domény byly získány z OpenPhish [68], automatizované platformy pro phishingovou inteligenci, a PhishTank [23], kolaborativní databáze pro data a informace o phishingu. Obě platformy důkladně ověřují hlášené domény, čímž snižují počet falešně pozitivních výsledků. Domény byly sbírány z jejich MISP kanálů, jakmile byly publikovány. Sběr vyústil v 164,901 potenciálních phishingových domén. Další filtrování bylo provedeno prostřednictvím služby VirusTotal [90], kterou provozuje Chronicle Security a která detekuje škodlivý obsah v souborech a URL, včetně detekce podvodných phishingových stránek. Pomocí API VirusTotal bylo identifikováno a odstraněno 476 chybně klasifikovaných domén. Toto filtrování vyústilo ve vysoce kvalitní datovou sadu 164,425 ověřených phishingových domén.

#### Datová sada benigních domén

K získání sady benigních domén byl použit veřejně dostupný seznam Top One Million od platformy Cisco Umbrella [22].

Tento seznam byl dále zpracován a filtrován, aby bylo zajištěno, že obsahuje pouze benigní domény. Byl implementován proces filtrování inspirovaný metodologií Rahbarinia a kol. [78]. Tento přístup zahrnuje extrakci měsíčních dat, pro tento milion nejvyhledávanějších domén a provedení selekce takových domén, které se zde nacházejí pravidelně. Dodatečné benigní domény byly získány z provozu v akademických sítích.

#### Datová sada malware domén

Domény obsahující malware byly získány ze služby ThreatFox, což je online platforma poskytující komplexní přehled a analýzy ohledně hrozeb spojených s malware. ThreatFox se specializuje na shromažďování a poskytování podrobných informací o doménách, IP adresách a hash hodnotách souborů spojených s nejrůznějšími typy malware. Díky této službě je možné identifikovat aktuální a vznikající hrozby v kyberprostoru [3].

Domény byly průběžně sbírány v průběhu roku 2023 a 2024, čímž bylo zajištěno, že datová sada reflektuje aktuální stav a dynamiku v oblasti malware domén. Vzhledem k tomu, že životnost malware domén není dlouhá a jejich charakteristiky se mohou rychle měnit,

bylo nezbytné zajistit co nejrychlejší sběr a analýzu dat. Po sběru dat byly k doménám ihned doplněny informace, které umožňují jejich efektivní klasifikaci a analýzu.

Pro validaci a odstranění chybně nahlášených domén bylo využito služby VirusTotal, která poskytuje rozsáhlou detekci škodlivého obsahu v souborech a URL, včetně detekce phishingových a malware domén. Tento proces dalšího ověřování zvyšuje kvalitu a spolehlivost datové sady [90].

### 5.1.2 Verifikační datová sada

Pro účely nezávislého ověření klasifikační výkonnosti modelů byla vytvořena separátní verifikační datová sada, která nebyla nijak využita při trénování ani ladění modelů. Tato sada slouží k testování schopnosti modelu generalizovat na dosud neznámá data, pocházející z odlišného časového období i zdrojů.

Sběr dat probíhal ve dnech 23. až 25. května 2024. Benigní domény byly zachyceny z běžného provozu na síti CESNET, což odpovídá autentickému provozu v akademickém prostředí. Naopak škodlivé domény byly získány z veřejně dostupných databází *OpenPhish* [68], *PhishTank* [23] a z komunitně spravovaného blacklistu *StevenBlack*, dostupného na platformě Github[3].

Celkový rozsah verifikační sady činil 5 048 domén:

- 4 276 benigních domén
- 480 phishingových domén
- 292 malware domén

Stejně jako u hlavní datové sady byly všechny škodlivé domény podrobeny kontrole prostřednictvím služby VirusTotal [90], aby byla zajištěna jejich aktuálnost a validita. Podrobný popis této verifikační metodiky je uveden v sekci 5.5.

Verifikační sada tak představuje nezávislý a realistický vzorek domén, umožňující objektivní měření robustnosti a reálné použitelnosti navržených klasifikačních přístupů.

## 5.2 Zdroje dat

Data domén z výše uvedených datových sad byla získána z různých zdrojů, zahrnující jak veřejně dostupné bezpečnostní feedy, tak i specializované databáze. Mezi hlavní zdroje dat patří: [72]

- **VirusTotal**: Online platforma poskytující nástroje pro analýzu URL a souborů pomocí více antivirových nástrojů.
- **Bezpečnostní feedy a černé listiny**: Databáze obsahující seznamy známých maligních domén, jako je MISP a PhishTank.
- **RDAP a WHOIS databáze**: Informace o registračních údajích domén.

## 5.3 Metodologie a proces sběru domén

Filtrování dat bylo prováděno na základě několika klíčových kritérií, aby byla zajištěna kvalita a spolehlivost datových sad. Hlavní kroky zahrnovaly: [72]

[3] https://github.com/StevenBlack/hosts

- **Validace doménových jmen**: Využití API VirusTotal k ověření aktuálního stavu domén (maligní/benigní).
- **Čištění dat**: Odstranění duplicitních záznamů, neúplných dat a domén s nespecifikovanými atributy.

Domény byly získány z různorodých zdrojů a následně obohaceny o doplňující informace prostřednictvím nástroje *DomainRadar*, který vznikl v rámci řešení projektu FETA. Na jeho vývoji jsem se podílel jako spoluautor. Tento nástroj využívá moduly implementované v jazyce Python pro sběr dat z externích služeb, jako jsou WHOIS/RDAP dotazy nebo extrakce TLS certifikátů.

### 5.3.1 DomainRadar

V rámci této diplomové práce byl zásadním nástrojem pro sběr a zpracování dat systém **DomainRadar** [42], který vznikl na Fakultě informačních technologií VUT v Brně jako součást projektu *Metody AI pro zabezpečení kybernetického prostoru a řídicí systémy* (FIT-S-20-6293, 2020–2023). Jedná se o komplexní, modulárně navržený systém pro aktivní i pasivní sběr, ukládání a analýzu doménových dat. Jeho cílem je efektivní příprava datových sad pro analýzu hrozeb, detekci škodlivých domén a vývoj modelů strojového učení. Systém je dostupný jako výzkumný prototyp [4].

DomainRadar je tvořen sadou specializovaných kolektorů, které získávají informace o doménách z různých zdrojů — například DNS záznamy, RDAP metadata, TLS certifikáty, reputační informace, geolokaci IP adres nebo informace o ASN. Jednotlivé datové proudy jsou následně zpracovávány pomocí komponent jako **Data Merger**, které sjednocují data do jednotného JSON formátu a ukládají je do databáze **MongoDB**. Architektura systému je založena na **Apache Kafka**, která slouží k orchestraci sběru a oddělení jednotlivých fází pipeline. Jednotlivé komponenty systému jsou znázorněny na obrázku 5.1 v sekci pojednávající o procesu sběru.

Získaná data jsou následně zpracovávána pomocí nástroje **Feature Extractor**, který převádí JSON reprezentaci domén na numerické vektory příznaků. Ty jsou ukládány ve formátu **Apache Parquet**[5], a to ve dvou verzích: (1) jako surové cache výstupy z databáze a (2) jako finální vektory příznaků připravené pro strojové učení.

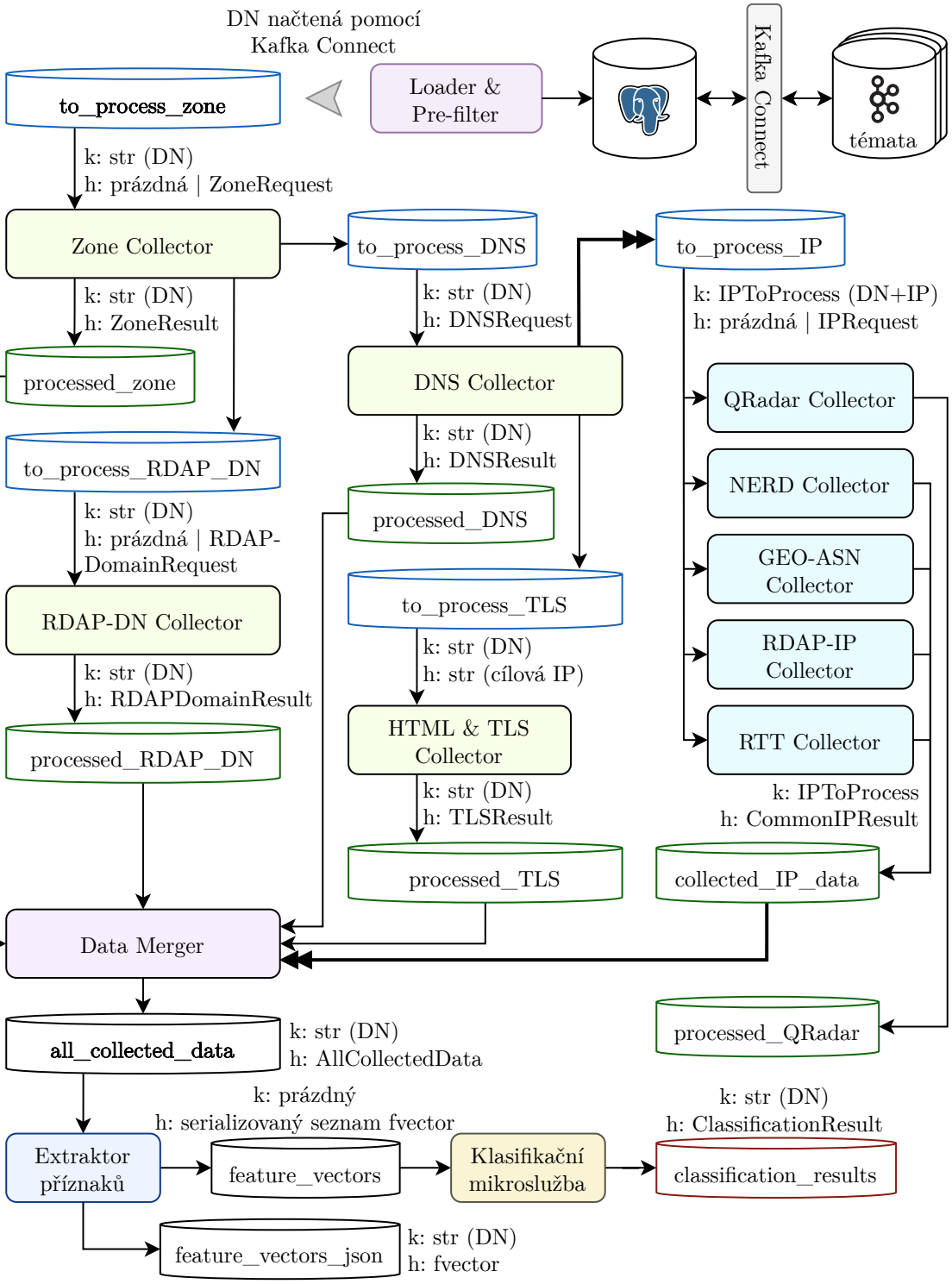
#### Domain Collector

Softwarem ze kterého vychází systém DomainRadar je systém **Domain Collector** [38], který umožňuje automatizovaný sběr, agregaci a ukládání informací o doménách. Tento nástroj využívá data ze zdrojů jako **Cisco Umbrella**, **PhishTank**, **OpenPhish**, **ThreatFox** a dalších threat intelligence platforem. Získaná data jsou obohacena pomocí aktivního sběru (např. ICMP testy, TLS handshaky) a následně ukládána do MongoDB pro další zpracování.

### 5.3.2 Segmentovaný proces sběru

Schéma na obrázku 5.1 znázorňuje komponenty systému **DomainRadar**, které zajišťují plně automatizovaný a hierarchicky uspořádaný sběr dat o doménách. Celý proces je navržen

[4] https://www.fit.vut.cz/research/product/c179852/.en
[5] https://parquet.apache.org/




Obrázek 5.1: Schéma segmentovaného procesu sběru dat v systému DomainRadar [42], převzato z [67].

jako sekvence na sebe navazujících kroků, kde každý z kolektorů doplňuje doménový záznam

o specifický typ informací — od základních syntaktických rysů až po reputační skóre IP adres a metainformace z TLS certifikátů.

Tato hierarchická struktura sběru má přímý dopad na návrh skupin příznaků a definici jejich logického rozšiřování. Strategie konstruování vektorů příznaků ve třech úrovních — od čistě lexikálních až po plně obohacené záznamy s RDAP a HTML — přímo vychází z architektury tohoto systému. Tato souvislost je blíže popsána v sekci 7.2.3.

**Načtení a filtrování domén.** Na vstupu systému figuruje komponenta **Loader & Prefilter**, která načítá seznamy domén (např. z PostgreSQL databáze) a prostřednictvím Kafka Connect je vkládá do tematických front. Tato komponenta zároveň filtruje již dříve zpracované domény, čímž zamezuje duplicitnímu sběru.

**Sběr dat pomocí kolektorů.** Domény postupně procházejí skrze sadu asynchronně volaných sběračů:

- **Zone Collector** – získává základní informace z TLD zón.
- **DNS Collector** – zajišťuje DNS záznamy (A, AAAA, MX, NS...).
- **RDAP-DN Collector** – provádí RDAP dotaz na registrátora domény.
- **HTML & TLS Collector** – navazuje TLS spojení a získává informace o certifikátu.
- **IP kolektory** – navazují na IP adresy získané z DNS a doplňují:
  - **QRadar Collector** – reputace IP z bezpečnostních databází,
  - **NERD Collector** – stav IP z NERD databáze,
  - **GEO-ASN Collector** – geografii a číslo AS,
  - **RDAP-IP Collector** – RDAP údaje k IP,
  - **RTT Collector** – měření odezvy IP.

Každý kolektor odesílá výstup do specifického tématu v Kafce (např. `processed_TLS`), odkud jsou data odeslána k agregaci.

**Agregace a extrakce příznaků.** Všechny informace jsou sloučeny pomocí komponenty **Data Merger** do jednotné struktury `all_collected_data`. Tento výstup je předán do **extraktoru příznaků**, který data převede na číselné vektory. Ty jsou následně serializovány a uloženy ve formátu **Apache Parquet**.

Nástroj pro extrakci[6] podporuje jak export pro dávkové trénování, tak i klasifikaci v reálném čase pomocí klasifikační mikroslužby.

**Neúplnost záznamů.** Vzhledem k povaze sběru dat není garantováno, že pro každou doménu budou k dispozici všechny typy dat. Například:

- doména nemusí mít validní TLS endpoint,
- RDAP servery mohou být nedostupné,

[6] https://github.com/nesfit/domainradar-training/tree/main/feature-extraction

- IP může chybět v reputační databázi.

Tato neúplnost ovlivňuje dostupnost příznaků, a tím i konstrukci datových subsetů. Práce proto zohledňuje rozdílné úrovně datové úplnosti v návrhu experimentů i hodnocení výsledků (viz sekce 7.2.2).

Pipeline systému DomainRadar je navržena s důrazem na modularitu a možnost připojení dalších zdrojů dat, včetně paralelního zpracování rozsáhlých datových sad.

## 5.4 Verifikace Ground-truth

Ground-truth (česky "referenční pravda") představuje v kontextu strojového učení a datové analýzy označení pro data, jejichž správná klasifikace je známa a ověřena. V rámci této práce ground-truth kolekce reprezentují doménová jména, u kterých bylo na základě víceúrovňové verifikace potvrzeno, zda se jedná o domény benigní, phishingové nebo sloužící k šíření malwaru. Tato referenční data jsou zásadní pro správné trénování a hodnocení klasifikačních modelů, protože určují správné cílové hodnoty, vůči kterým jsou měřeny chyby modelu. K ověření pravdivosti datových sad byla implementována následující strategie: [72]

- Všechny domény byly analyzovány pomocí akademické verze VirusTotal API, která poskytuje víceúrovňové hodnocení pomocí různých antivirových enginů.
- Výsledky analýz byly zpracovány pomocí skriptu pro hromadné označování domén jako maligních nebo benigních.
- Finální verifikace byla provedena porovnáním s veřejně dostupnými seznamy černých listin a ručním přezkoumáním podezřelých vzorků.

Tento přístup zajistil, že vytvořená datová sada obsahuje kvalitní a spolehlivá data, která slouží jako základ pro trénování a testování klasifikačních modelů [72].

## 5.5 Filtrování datových sad

Aby bylo možné zaručit vysokou kvalitu a spolehlivost vytvořených datových sad, bylo nezbytné aplikovat systematický proces filtrování doménových záznamů. Tento proces přímo navazuje na segmentovanou architekturu systému **DomainRadar**, jejíž výstupem je kolekce domén uložená v databázi **MongoDB** a následně exportovaná ve formátu **Apache Parquet**[7].

Důvodem volby právě tohoto formátu je jeho přímá návaznost na výstupní fázi sběrové pipeline (viz sekce 5.3.2). V systému DomainRadar vznikají typicky dva druhy Parquet souborů:

- **Raw Parquet** – dočasný výstup generovaný z MongoDB projekcí, obsahující původní JSON data získaná kolektory (DNS, RDAP, TLS, atd.),
- **Parquet feature vektory** – zpracovaná reprezentace vytvořená nástrojem *Feature Extractor*, kde jsou jednotlivé domény reprezentovány jako číselné vektory připravené pro klasifikaci.

[7] https://parquet.apache.org/

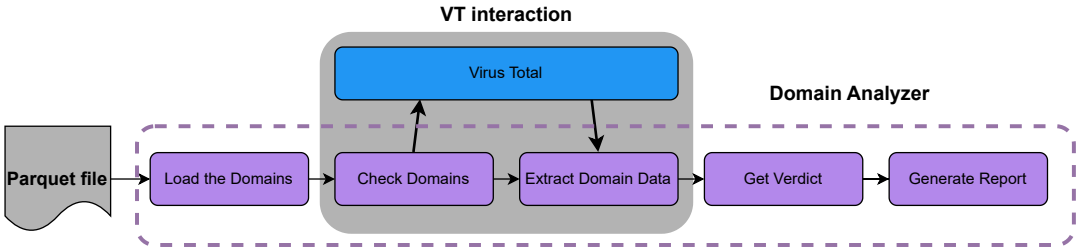
Použitý extraktor je veřejně dostupný v repozitáři [8] systému DomainRadar.

Filtrování bylo aplikováno buď přímo nad těmito Parquet soubory, nebo ve specifických případech nad extrahovanými seznamy domén. Další sekce pak popisují konkrétní mechanismy filtrace.

### 5.5.1 Ověřování domén pomocí VirusTotal

Pro externí ověření důvěryhodnosti domén byla použita služba **VirusTotal**, která kombinuje detekční výsledky desítek antivirových enginů a reputačních databází. Systém načetl domény ze vstupního Parquet souboru, odeslal je pomocí oficiálního API ke kontrole a následně extrahoval metriky jako *detection count*, *malicious flag* a seznam identifikovaných detekcí.

Jak ukazuje obrázek 5.2, celý proces ověřování byl automatizován a navázán na další filtrovací mechanismy. Na základě výstupů z VirusTotal byly domény rozděleny do kategorií (např. spolehlivě benigní, podezřelé, potvrzené škodlivé) a pouze validované domény byly zařazeny do finální datové sady.




Obrázek 5.2: Interakce se službou VirusTotal v rámci ověřování domén. Převzato z dokumentace systému DomainRadar [42].

### 5.5.2 Filtrace domén z dat ze sítě CESNET

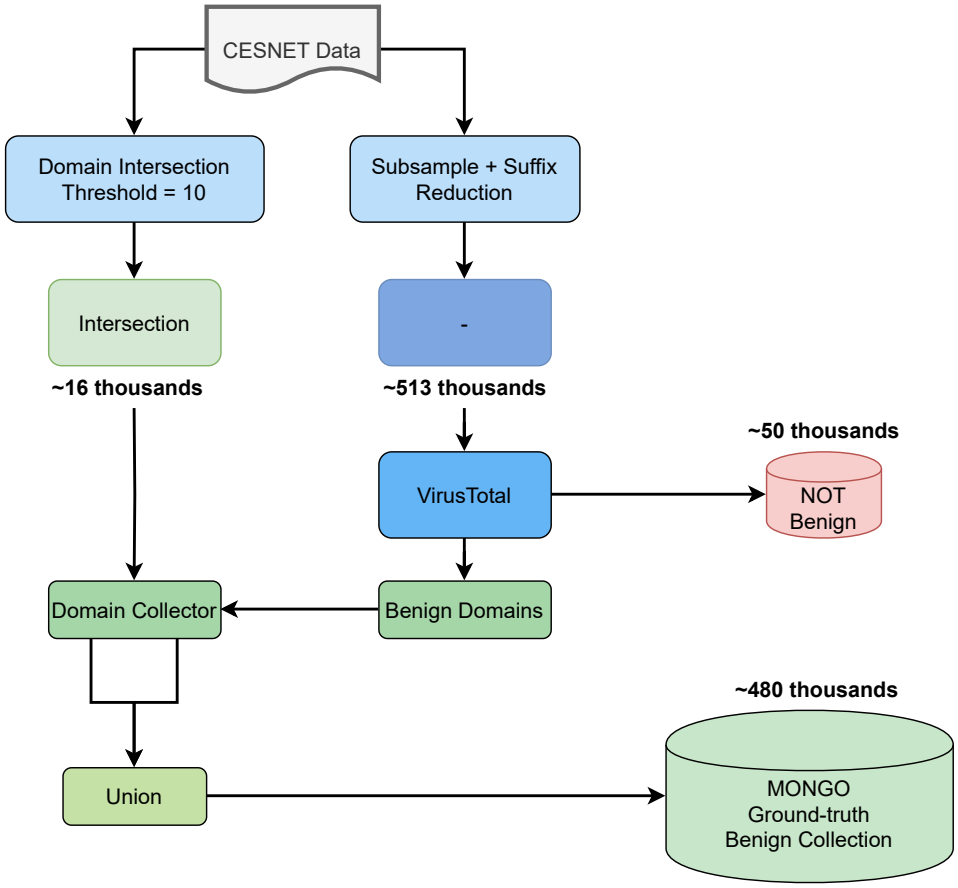
Specifickým případem byl sběr benigních domén z akademické sítě CESNET. Tyto domény byly považovány za potenciálně důvěryhodné, ale bylo nezbytné odstranit technické domény, překlepy a domény s příliš nízkým výskytem.

Aplikován byl dvoustupňový filtrační proces, který kombinuje:

1. **Intersekci na základě výskytu** – ponechány byly pouze domény, které se vyskytovaly v provozu pravidelně a opakovaně.
2. **Suffix redukci** – odstraněny byly domény, které měly společný kořen nebo generické TLD bez informativní hodnoty (např. `cdn`, `akamaiedge`).
3. **Podvzorkování** – pro zajištění vyváženosti mezi třídami byly některé domény náhodně vybrány, aby odpovídaly velikostně ostatním kategoriím.

Celý proces je znázorněn na obrázku 5.3, který je převzat z práce P. Pouče [72].

[8] https://github.com/nesfit/domainradar-training/tree/main/feature-extraction




Obrázek 5.3: Proces filtrace domén získaných z CESNET dat. Převzato z práce P. Pouče [72].

### 5.5.3 Shrnutí

Použitím kombinace vícestupňového filtrování, validace třetí stranou (VirusTotal) a heuristik vyvinutých pro akademický provoz byly vytvořeny vysoce kvalitní datové sady s minimem chybné klasifikace. Tento proces navazuje na architekturu sběru dat systému DomainRadar a jeho výstupní Parquet formát, čímž je zajištěna konzistence celého pipeline od sběru až po klasifikaci.

## 5.6 Transformace datových příznaků

Pro účely efektivního trénování klasifikátorů bylo nezbytné navrhnout systematický postup předzpracování doménových dat. Vstupní data byla dodána ve formě `parquet` souborů obsahujících nezpracované příznaky domén (např. počty záznamů, délku názvu, entropii, geolokační informace apod.). Aby bylo možné tyto příznaky použít pro strojové učení, bylo nutné provést několik kroků transformace, které lze rozdělit na dvě hlavní fáze: **obecné předzpracování**, které je shodné pro všechny modely a **modelově specifické transformace**, odrážející potřeby individuálních modelů.

### 5.6.1 Obecné předzpracování

Tato fáze byla společná pro všechny klasifikační modely a aplikovala se na celou datovou množinu.

1. **Čištění a typová normalizace:** Chybějící hodnoty byly nahrazeny nulami, binární příznaky převedeny na celočíselné reprezentace (0/1), a redundantní nebo nekonzistentní sloupce byly odstraněny.

2. **Min-max škálování:** Všechny příznaky byly převedeny do jednotného rozsahu $[0, 1]$ pomocí standardního min-max škálování:

$$x_{ij}^{\text{scaled}} = \frac{x_{ij} - \min_j}{\max_j - \min_j} \tag{5.1}$$

   kde $x_{ij}$ je původní hodnota příznaku $j$ pro vzorek $i$.

3. **Odstranění odlehlých hodnot:** Pro zvýšení robustnosti modelů byly odstraněny extrémní hodnoty na základě jejich vzdálenosti od průměru. Hodnota byla považována za odlehlou, pokud:

$$x < \mu - 2\sigma \quad \text{nebo} \quad x > \mu + 2\sigma \tag{5.2}$$

   kde $\mu$ je průměr a $\sigma$ směrodatná odchylka daného příznaku.

Tento obecný postup zajišťuje, že všechny vstupní atributy jsou srovnatelné a vhodné pro většinu algoritmů strojového učení, zejména těch, které jsou citlivé na měřítko (např. SVM nebo kNN).

### 5.6.2 Modelově specifické transformace

Některé modely vyžadují dodatečné transformace vstupních dat, které reflektují jejich interní fungování. Typickým příkladem jsou neuronové sítě, které těží z hladce rozložených vstupů v omezeném rozsahu.

#### Sigmoidní transformace

U neuronových sítí byla po základním škálování aplikována navíc sigmoidní funkce, která převede hodnoty do intervalu $(0, 1)$ a zároveň zvýrazní rozdíly v okolí nulové hodnoty:

$$x_{ij}^{\text{sigmoid}} = \frac{1}{1 + e^{-(a_{ij} - \mu_j)/\sigma_j}} \tag{5.3}$$

Zde $a_{ij}$ představuje škálovanou hodnotu příznaku, $\mu_j$ a $\sigma_j$ odpovídají průměru a směrodatné odchylce daného příznaku. Tato transformace pomáhá stabilizovat zpětnou propagaci chyb a zmírňuje výskyt problémů jako *exploding* nebo *vanishing gradients*.

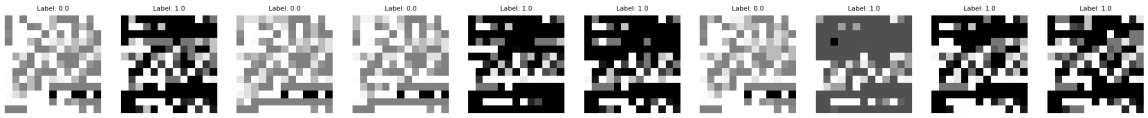
#### 2D transformace dat domén

Pro účely konvoluční neuronové sítě (CNN) je nezbytné transformovat doménová data do dvourozměrného (2D) formátu. Tato transformace umožňuje využití technik z oblasti zpracování obrazu, jako je detekce vzorů prostřednictvím konvolučních vrstev. V našem případě jsou jednotlivé atributy dat škálovány do intervalu [0, 1], což umožňuje interpretovat je jako intenzity pixelů.

Vizualizace na obrázku 5.4 demonstruje, jak je každá doména reprezentována jako 2D snímek, kde každý pixel odpovídá jednomu atributu. Proces transformace dat je založen na převodu jednorozměrného vektoru atributů na dvourozměrný obraz. Nechť $n$ je počet atributů (prvků) v každém vzorku, pak strana čtverce $s$ pro reprezentaci dat ve 2D prostoru je dána výrazem:

$$s = \lceil \sqrt{n} \rceil,$$

kde $\lceil \cdot \rceil$ označuje zaokrouhlení nahoru. Tímto způsobem je každý vektor atributů o velikosti $n$ transformován na matici o rozměrech $s \times s$. Tento postup je aplikován na všechny vzorky v trénovací i testovací sadě dat. Takto přetransformovaná data jsou vhodná pro zpracování pomocí CNN, což umožňuje efektivní detekci a klasifikaci vzorů a charakteristik v datech.

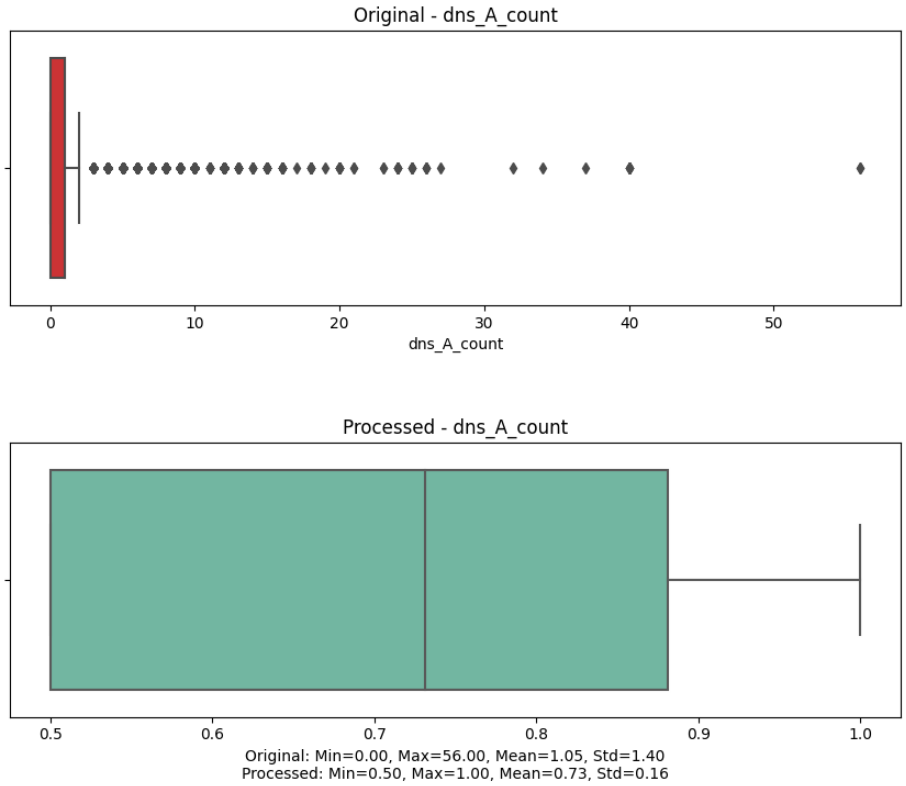



Obrázek 5.4: Vizualizace domén předpřipravených pro CNN

### 5.6.3 Vizualizace a dopad transformací

Na obrázku 5.5 je ukázáno srovnání mezi nezpracovanou a zpracovanou verzí příznaku `dns_NS_count`. Je patrné, že odstraněním odlehlých hodnot a následným škálováním došlo k výraznému zlepšení rozložení hodnot, což je žádoucí pro zajištění konzistentního trénování modelů.

Transformace dat tímto způsobem zajišťuje, že výsledné atributy jsou numericky stabilní, vhodné pro strojové učení a zároveň robustní vůči šumu a extrémům v datech. Na jednotlivé modely se v dalších kapitolách odkazujeme vždy s poznámkou, jaké konkrétní části tohoto postupu byly využity.




Obrázek 5.5: Porovnání zpracovaných a nezpracovaných hodnot příznaku `dns_NS_count`.

# Kapitola 6

# Tvorba a segmentace příznaků

Úspěšná detekce phishingových a malwarových domén je podmíněna vhodnou reprezentací vstupních dat, tedy takovou sadou příznaků, která co nejlépe zachycuje rozdíly mezi benigními a škodlivými doménami.

Tato kapitola proto nejprve vymezuje **terminologii** a zpřesňuje kategorie příznaků, z nichž je sestaven vektor použitý ve všech následných experimentech. Následně jsou prezentována **vlastní měření a srovnání současných přístupů** popsaných v literatuře a je diskutován jejich praktický dopad na klasifikační úspěšnost. Třetí část formuluje **metodologii tvorby příznaků** vycházející z nástroje `DomainRadar` a definuje subsety různé náročnosti sběru. [39] Kapitola je zakončena cíleným **feature engineeringem pro TLS certifikáty**, jenž rozšiřuje původní vektor o atributy s vysokým diskriminačním potenciálem.

Veškeré analýzy a výsledné metriky vycházejí z datové sady popsané v kapitole 5.

## 6.1 Terminologie a kategorie příznaků

Základem této práce je vektor příznaků převzatý z nástroje `DomainRadar`, který obsahuje celkem **263** atributů. Tabulkový přehled všech použitých příznaků je uveden v příloze D [42]. Příznaky byly vytvořeny v rámci publikace [40] a vývojového produktu [39], na nichž jsem se podílel jako spoluautor.
Všechny atributy byly rozděleny podle původu do následujících kategorických skupin:

- **LEX** – lexikální příznaky odvozené přímo z doménového jména; zahrnují délku druhé úrovně, Shannonovu entropii, četnosti znaků či $n$-gramové shody.
- **DNS** – parametry DNS odpovědí, například počty jednotlivých RR typů, průměrné hodnoty TTL nebo skóre DNSSEC.
- **IP** – charakteristiky IP adres navázaných na doménu (počet IPv4/IPv6, diverzita autonomních systémů, rozptyl RTT).
- **TLS** – vlastnosti TLS handshake a certifikátů, např. verze protokolu, použitá ciphersuite, délka a validita certifikačního řetězce.
- **GEO** – geolokační údaje IP adres získané z databází MaxMind GeoLite2 (země, region, vzdálenost k měřícímu uzlu).

- **RDAP/WHOIS** – informace o registraci a vlastnictví domény, stáří, registrátor či využití anonymizačních služeb.
- **HTML** – atributy odvozené z obsahu webové stránky, jako je počet rámců či přítomnost formulářů.
- **MISC** – doplňkové příznaky, jež nelze jednoznačně zařadit (např. TLD abuse score).

Kromě uvedených základních kategorií byly zkoumány také různé *agregace* těchto skupin, přičemž byl brán ohled na obtížnost a časovou náročnost sběru jednotlivých atributů vycházející z technické implementace nástroje DomainRadar [42].

## 6.2 Srovnání existujících příznaků

V rámci tohoto výzkumu byly replikovány přístupy vybraných studií uvedených v kapitole 2. Z původních článků byla vždy převzata definice příznaků a implementován odpovídající transformační skript, který vytvořil vektorové reprezentace potřebné pro trénink klasifikačních modelů. Pro každý zdrojový článek byl použit klasifikátor doporučený autory; v případech neúplné specifikace hyperparametrů byly tyto nalezeny metodou *grid-search*. Pokud studie nabízela více algoritmů, byl do tabulky zařazen ten s nejvyšší dosaženou hodnotou F1.

Tabulka 6.1 shrnuje nejlepší replikované výsledky na vlastní datové sadě z kapitoly 5. Hodnoty tedy neodpovídají číslům převzatým z originálních publikací, nýbrž skóre naměřenému v jednotném prostředí tohoto výzkumu. Tyto výsledky následně sloužily jako východisko pro návrh vlastních rozšíření popsaných v navazujících kapitolách.

| První autor | Rok | Typ | Nejlepší F1 | # f. | Kat. přízn. | Model |
|---|---|---|---|---|---|---|
| Torroledo | 2018 | Malw | 0.966 | 30 | TLS | LightGBM |
| Shi | 2017 | Mix | 0.915 | 9 | MIX | LightGBM |
| Magalhães | 2020 | Mix | 0.969 | 17 | MIX | LightGBM |
| Zhu | 2019 | Mix | 0.910 | 11 | MIX | AdaBoost |
| Kumar | 2022 | Malw | 0.932 | 15 | LEX | AdaBoost |
| Silveira | 2021 | Mix | 0.921 | 19 | DNS | SVM |
| Iwahana | 2021 | Mix | 0.968 | 25 | MIX | LightGBM |
| Gopinath | 2020 | C&C | 0.937 | 17 | WHOIS+DNS | LightGBM |
| Hason | 2020 | Mix | 0.971 | 9 | MIX | LightGBM |
| Chatterjee | 2019 | Phish | 0.924 | 14 | MIX | XGBoost |
| Sadique | 2020 | Phish | 0.924 | 20 | MIX | XGBoost |

Tabulka 6.1: Replikace vybraných studií

### Nejpoužívanější příznaky

Na základě analýzy tabulky 6.1 můžeme identifikovat nejčastěji používané typy příznaků a jejich variabilitu mezi jednotlivými přístupy. Nejčastěji se objevují kombinované příznaky (MIX), které zahrnují různé aspekty doménových charakteristik, jako jsou DNS záznamy, TLS certifikáty nebo lexikální rysy doménových jmen. Tento přístup je patrný zejména ve studiích autorů Shi kol. [83], Iwahana kol. [45] a Hason kol. [35], kteří dosahují průměrných

skóre F1 mezi 0.90 a 0.95. Kombinované příznaky poskytují vyváženou reprezentaci dat a umožňují klasifikátorům efektivněji generalizovat.

Dalším často používaným typem příznaků jsou příznaky založené na TLS certifikátech, jak ukazuje práce Torroleda a kol. [89], která dosahuje velmi vysokého skóre F1 (0,9245). TLS příznaky jsou klíčové zejména v kontextu bezpečnostních atributů a spolehlivosti domén. Tento přístup těží z hluboké analýzy certifikátů a jejich atributů, jako jsou délka platnosti, certifikační autorita nebo typ validace. V příloze E této práce je uveden experimentální klasifikátor založený výhradně na TLS příznacích, který byl navržen s cílem ověřit jejich samostatnou výpovědní hodnotu. Ačkoliv jeho výkon nedosahuje úrovně kombinovaných modelů, výsledky naznačují, že TLS certifikáty mohou poskytovat doplňkový signál užitečný zejména v případech, kdy jiné datové zdroje nejsou dostupné.

Naopak méně časté jsou příznaky založené výhradně na lexikální analýze (LEX), jak ukazuje studie Kumar [52], která i přes menší množství příznaků (15) dosahuje skóre F1 0.87. Lexikální příznaky se zaměřují na strukturu a vzorce v doménových jménech, což může být užitečné zejména pro detekci domén generovaných algoritmy (DGA). Jejich omezené rozšíření může souviset s tím, že samy o sobě neposkytují kontextuální ani síťové informace.

Příznaky založené na DNS a WHOIS záznamech se objevují méně často, ale stále hrají důležitou roli. Například studie Gopinatha [69] ukazuje, že kombinace těchto příznaků může vést k solidním výsledkům s skóre F1 0.87. Tyto příznaky zahrnují informace o registraci domény, geografii serverů a metriky DNS dotazů, které pomáhají identifikovat nově registrované nebo podezřele krátkodobě aktivní domény.

## 6.3 Metodologie tvorby příznaků

V návaznosti na výsledky replikovaných studií a vlastní analýzy byl navržen systematický postup tvorby a rozšiřování příznaků, s cílem maximalizovat klasifikační úspěšnost při detekci phishingových a škodlivých domén.

Výchozí inspirací pro metodologii byl nástroj `DomainRadar`, jenž byl rozšířen na základě poznatků získaných v rámci výzkumu publikovaného v [41]. Tento přístup kombinuje více zdrojů dat (DNS, RDAP, TLS, IP, GEO) a vytváří komplexní vektor popisující doménu.

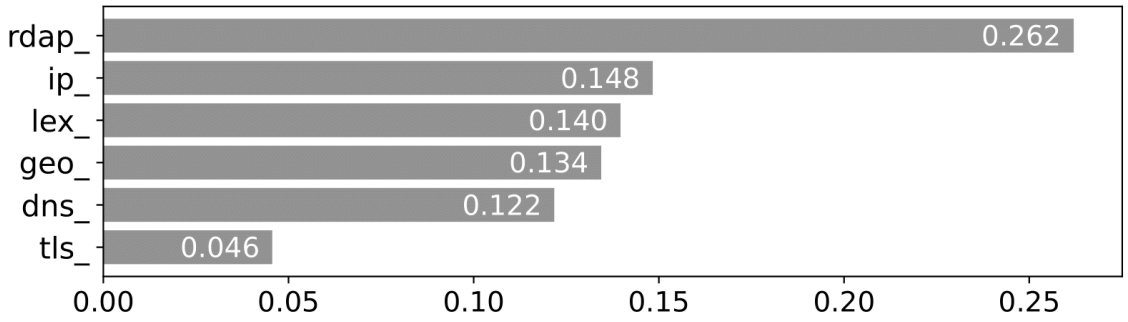
Na základě analýzy Shapleyho hodnot (SHAP) bylo ověřeno, že všechny hlavní kategorie příznaků přispívají k rozhodovacímu procesu, avšak jejich přínos není rovnoměrný. Největší význam vykazovaly atributy odvozené z **RDAP** (registrace domén a IP), následované **DNS** a **lexikálními** vlastnostmi. Oproti očekávání se ukázalo, že **TLS** příznaky mají výrazně menší dopad na rozhodování klasifikátorů, s výjimkou několika specifických parametrů, jako je vyjednaná šifra nebo délka certifikačního řetězce.

### 6.3.1 Agregovaná analýza přínosu kategorií

Na základě výsledků dřívější analýzy publikované v článku *Spotting the Hook* [41] byla provedena agregovaná interpretace významu jednotlivých kategorií příznaků pomocí Shapleyho hodnot vypočtených nad modelem LightGBM. Výpočty vycházejí z předchozí podmnožiny datové sady, která neobsahovala data z CESNETu a obsahovala menší zastoupení phishingových a malware domén.

Jak ukazuje obrázek 6.1, nejvyšší přínos k rozhodování vykazuje skupina RDAP příznaků, následovaná IP, LEX a GEO atributy. Naopak DNS a TLS příznaky měly ve srovnání s ostatními kategoriemi menší význam.

- **RDAP** – Nejvýznamnější skupina, zejména atributy jako věk domény a vlastnosti registrátora.
- **IP** – Entropie IP prefixů, počet autonomních systémů a další příznaky odvozené z dat IP adres přispívaly k odhalení phishingu.
- **LEX** – Vysoký význam měly zejména skóre zneužití TLD a $n$-gramové shody.
- **GEO** – Střední přínos; geografické rozložení serverů pomáhalo diferencovat legitimní a podvodné domény.
- **DNS** – Nízké TTL hodnoty a charakteristiky rekordů se ukázaly jako užitečné.
- **TLS** – Nejmenší agregovaný vliv, s výjimkou vybraných atributů (vyjednaná šifra, délka řetězce certifikátů).




Obrázek 6.1: Agregovaný přínos kategorií příznaků dle hodnot analýzy SHAP vypočtených pro LightGBM model. Výsledky vycházejí z podmnožiny dat použité v publikaci Spotting the Hook [41].

Tento rozbor potvrzuje, že efektivní detekce phishingu nelze postavit pouze na jedné kategorii dat, ale vyžaduje kombinaci více zdrojů. Zároveň zdůrazňuje potřebu cíleného rozšiřování klíčových kategorií a optimalizace méně přínosných oblastí.

### 6.3.2 Strategie dalšího rozšiřování

Na základě těchto poznatků byla metodologie rozšíření příznaků formulována následovně:

1. **Prioritizace expanze RDAP a IP příznaků:** Bylo navrženo doplnění nových atributů zaměřených na registrátory, DNSSEC podporu, entropii IP prefixů a geografickou diverzitu.
2. **Detailní analýza LEX příznaků:** Probíhá rozšiřování $n$-gramových shod a výpočty specifických skóre rizikovosti na základě doménového jména.
3. **Doplňková optimalizace DNS příznaků:** Sběr specifických parametrů TTL a analýza variability v čase.
4. **Kritická revize TLS příznaků:** Vzhledem k nízkému přínosu bylo rozhodnuto zaměřit se pouze na selektivní rozšíření těch atributů, které prokazatelně korelují s podvodnými doménami (např. délka certifikátu, typ cipheru).

Tento postup je dále podložen experimentálními výsledky z práce Hranického a kol. [41], kde bylo prokázáno, že kombinace více zdrojů informací výrazně snižuje míru falešných pozitivních detekcí a zvyšuje robustnost klasifikátorů.

### 6.3.3 Shrnutí a hlavní přínosy

Metodologie tvorby příznaků navazuje na předchozí vývoj a vychází z poznatků publikovaných v rámci práce Hranického et al. [41], která se zaměřuje na využití doménových dat pro pokročilou detekci phishingu.

- Využití pěti nezávislých datových zdrojů (DNS, RDAP, IP, GEO, TLS).
- Detailní analýzu přínosu jednotlivých kategorií pomocí metody SHAP.
- Cílenou expanzi klíčových příznaků a optimalizaci slabších oblastí.
- Experimentální ověření dopadu příznaků na skóre F1 a míru falešně pozitivních výsledků.

Výsledky ukazují, že pečlivý feature engineering zásadním způsobem ovlivňuje úspěšnost detekce phishingových domén a představuje klíčovou komponentu každého moderního systému kybernetické bezpečnosti.

# Kapitola 7

# Předběžná analýza podmnožin příznaků

Předběžná analýza byla provedena s cílem identifikovat nejpřínosnější skupiny příznaků pro detekci škodlivých doménových jmen. Bylo zkoumáno, podle kterých charakteristik je vhodné provádět klasifikaci a které modely poskytují nejpřesnější výsledky. Na základě těchto měření byly následně definovány výsledné skupiny příznaků a zvoleny nejvhodnější klasifikátory.

Pro automatizaci procesu trénování a porovnání modelů byla využita knihovna `PyCaret`, která umožňuje rychlé testování různých klasifikačních algoritmů na daných sadách příznaků. Pro každé měření jsou provedeny následující kroky:

1. Nastavení prostředí pro trénování modelů v knihovně `PyCaret`.
2. Pro každou skupinu příznaků se spustí trénovací proces s cílem maximalizovat *skóre F1*.
3. Jsou vybrány tři nejlepší modely pro každou skupinu a uloží se jejich výsledky.
4. Výsledky jsou následně agregovány do jedné tabulky pro lepší přehlednost.

## 7.1 Skupiny příznaků

Pro každou skupinu příznaků uvedenou výše 6.1 byla provedena série experimentů, v rámci nichž bylo testováno deset různých klasifikačních algoritmů. Dále budou použity zkratky, jejichž plný výčet se nachází níže:

- `RF` – Random Forest,
- `ET` – Extra Trees,
- `KNN` – K-Neighbors Classifier,
- `XGB` – XGBoost,
- `LGBM` – LightGBM,
- `DT` – Decision Tree,

- `GB` – Gradient Boosting,
- `ADA` – AdaBoost,
- `LDA` – Linear Discriminant Analysis,
- `Ridge` – Ridge Classifier,
- `LR` – Logistic Regression,
- `Dummy` – Dummy Classifier,
- `SVM` – Support Vector Machine (Linear),
- `QDA` – Quadratic Discriminant Analysis,
- `NB` – Naive Bayes

### 7.1.1 Seskupování

Jak bylo popsáno v sekci 5.3.2, systém DomainRadar implementuje segmentovaný proces sběru dat, v rámci kterého jsou doménová jména postupně obohacována o různé typy informací (DNS odpovědi, RDAP záznamy, TLS certifikáty apod.). Tento architektonický přístup znamená, že **ne vždy jsou k dispozici všechny druhy dat pro každou doménu**. V praxi proto vzniká potřeba pracovat s více úrovněmi kompletnosti dat – tzv. **podmnožiny**.

Na základě charakteru sběru a dostupnosti dat byly definovány tři hlavní podmnožiny příznaků:

1. **Pouze lexikální příznaky (`["lex_"]`)** – Tato podmnožina zahrnuje pouze informace vycházející z názvu domény samotné. Jelikož každé doménové jméno je alespoň známé, představují lexikální příznaky nejzákladnější a vždy dostupnou variantu.
2. **Lexikální + DNS + IP + geolokační příznaky (`["lex_", "dns_", "ip_", "geo_"]`)** – Tato podmnožina je dostupná, pokud se podaří provést úspěšnou DNS rezoluci doménového jména na IP adresu. Na základě IP adresy lze následně obohatit data o geolokační informace. Tyto příznaky jsou sbírány zároveň (viz schéma sběru DomainRadar na Obrázku 5.1) a jejich získání je zpravidla rychlé a nenáročné.
3. **Plná data (`["lex_", "dns_", "ip_", "tls_", "geo_", "rdap_"]`)** – Nejkomplexnější varianta, obsahující kromě předchozích příznaků také informace z TLS certifikátů a RDAP protokolu. Tato podmnožina umožňuje dosáhnout nejvyšší přesnosti klasifikace, protože kombinuje široké spektrum technických a registračních dat. Je však dostupný pouze pro domény, pro které byl sběr těchto pokročilých dat úspěšný.

Výběr konkrétní podmnožiny příznaků pro klasifikaci závisí na úspěšnosti sběru dat. **V případě neúplných dat je provedena klasifikace pomocí nejvyšší kompletní podmnožiny.** Například pokud není dostupný RDAP záznam, ale je známá IP adresa a geolokační údaje, použije se podmnožina `["lex_", "dns_", "ip_", "geo_"]`. Pokud selže i DNS rezoluce, využijí se pouze lexikální příznaky.

Tento přístup umožňuje maximalizovat využitelnost nasbíraných dat a zajistit, že každé doménové jméno bude klasifikováno s nejvyšší možnou mírou detailu, kterou dostupná data umožňují.

## 7.2 Výsledky měření

Celkem bylo provedeno 420 měření, přičemž každá metoda byla aplikována na 42 různých datových sad, zahrnujících jednotlivé skupiny příznaků a jejich případné agregace.

Kvůli přehlednosti vizualizací byla zavedena následující tabulka pro překlad jednotlivých agregací:

| Zkratka | Klíč | # příznaků | # vzorků |
|---|---|---|---|
| Lex | `lex` | 63 | 649037 |
| Lex+dns+ip | `lex_+dns_+ip` | 111 | 649037 |
| Lex+...+geo | `lex_+dns_+ip_+geo` | 129 | 649037 |
| Lex+...+tls | `lex_+dns_+ip_+tls_+geo` | 153 | 649037 |
| Lex+...+rdap | `lex_+dns_+ip_+tls_+geo_+rdap` | 177 | 649037 |
| Lex+...+html | `lex_+dns_+ip_+tls_+geo_+rdap_+html` | 264 | 49037 |

Tabulka 7.1: Překlad popisů agregací skupin příznaků

Výsledky měření můžeme rozdělit do tří částí:

- Samostatné skupiny příznaků - Pouze na základě izolovaných skupin příznaků dle jejich původu.
- Agregované skupiny příznaků - Postupné agregování příznaků.
- Logické stupňování příznaků - Dle pořadí sběru.

Definici jednotlivých segmentů příznaků nalezneme výše v sekci Segmentace. Pro ilustraci struktury měření je uveden příklad výsledků dosažených na podmnožině `dns` pro malwar domény:

| **Model** | **Acc** | **AUC** | **Rec** | **Prec.** | **F1** | **Kappa** | **MCC** | **TT (s)** |
|---|---|---|---|---|---|---|---|---|
| `RF` | 0.9165 | 0.9120 | 0.9165 | 0.9135 | 0.9143 | 0.6861 | 0.6886 | 1.80 |
| `ET` | 0.9155 | 0.8910 | 0.9155 | 0.9125 | 0.9133 | 0.6829 | 0.6851 | 0.45 |
| `KNN` | 0.9142 | 0.8860 | 0.9142 | 0.9109 | 0.9116 | 0.6757 | 0.6787 | 0.24 |
| `XGB` | 0.9166 | 0.9175 | 0.9166 | 0.9129 | 0.9115 | 0.6681 | 0.6785 | 0.44 |
| `LGBM` | 0.9174 | 0.9176 | 0.9174 | 0.9144 | 0.9113 | 0.6647 | 0.6794 | 0.61 |
| `DT` | 0.8958 | 0.8503 | 0.8958 | 0.8957 | 0.8957 | 0.6268 | 0.6271 | 0.19 |
| `GB` | 0.8911 | 0.8799 | 0.8911 | 0.8921 | 0.8728 | 0.4991 | 0.5538 | 2.77 |
| `ADA` | 0.8632 | 0.8417 | 0.8632 | 0.8536 | 0.8335 | 0.3352 | 0.3997 | 0.71 |
| `LDA` | 0.8491 | 0.7843 | 0.8491 | 0.8264 | 0.8136 | 0.2532 | 0.3100 | 0.17 |
| `Ridge` | 0.8346 | 0.7842 | 0.8346 | 0.7962 | 0.7724 | 0.0722 | 0.1401 | 0.12 |
| `LR` | 0.8242 | 0.6988 | 0.8242 | 0.7307 | 0.7582 | 0.0143 | 0.0262 | 1.31 |
| `Dummy` | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 | 0.06 |
| `SVM` | 0.6466 | 0.6454 | 0.6466 | 0.7720 | 0.6756 | 0.1180 | 0.1475 | 0.14 |
| `QDA` | 0.5847 | 0.7880 | 0.5847 | 0.8468 | 0.6332 | 0.2208 | 0.3161 | 0.14 |
| `NB` | 0.3432 | 0.6313 | 0.3432 | 0.8202 | 0.3546 | 0.0640 | 0.1513 | 0.07 |

Tabulka 7.2: Výsledky modelů pro podmnožinu DNS (malware); modely označeny zkratkami

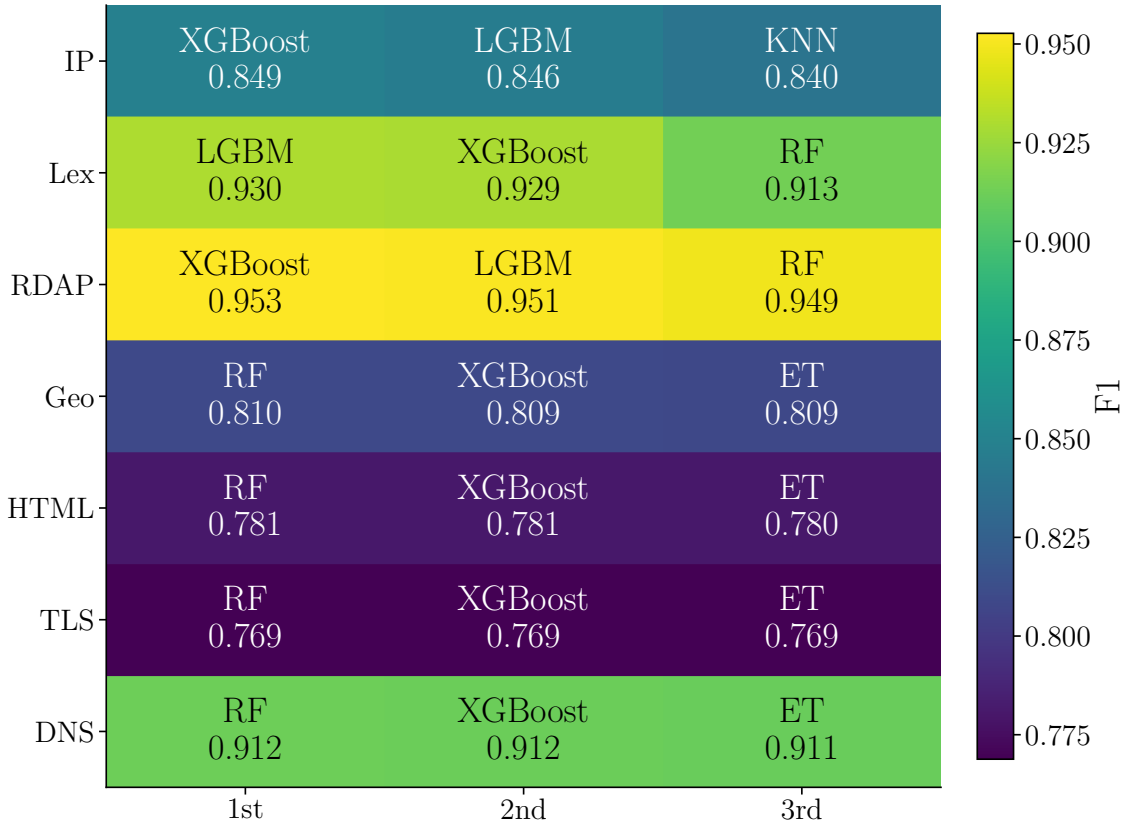
Kompletní výsledky všech měření a porovnání modelů pro ostatní podmnožiny jsou uvedeny v příloze G.

### 7.2.1 Samostatné skupiny příznaků

V první sérii experimentů byla každá kategorie příznaků testována samostatně. Cílem bylo zjistit, které jednotlivé typy informací (např. pouze TLS, pouze DNS, pouze LEX) nesou největší diskriminační sílu při klasifikaci škodlivých domén. Výsledky měření ukázaly, že nejvyšší úspěšnosti v samostatných skupinách dosahovaly:

- **RDAP příznaky** – zejména díky atributům souvisejícím s věkem a registrací domén,
- **DNS příznaky** – hlavně v oblasti TTL hodnot a četnosti typů záznamů,
- **IP příznaky** – entropie IP prefixů a diverzita autonomních systémů.

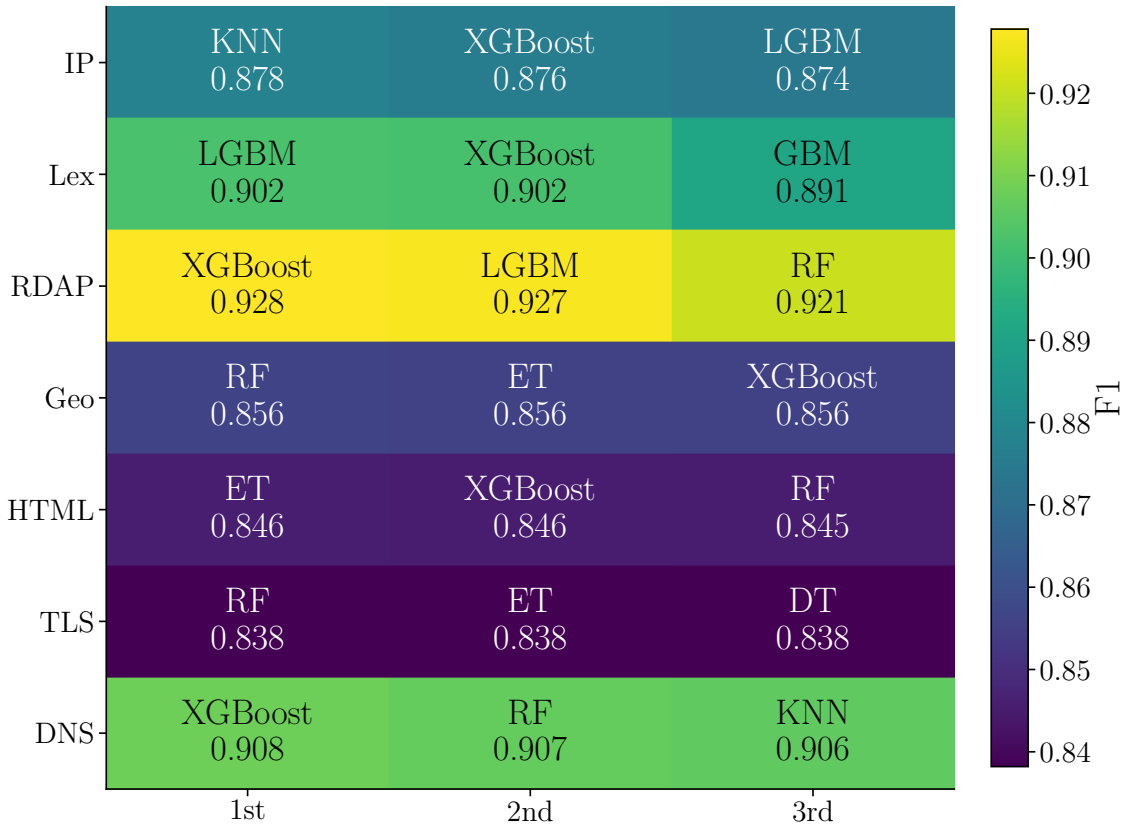
Podrobné výsledky pro malware a phishing domény nalezneme na ilustracích 7.1 a 7.2.




Obrázek 7.1: Výsledky měření samostatných podmnožin – phishing.

### 7.2.2 Agregované skupiny příznaků

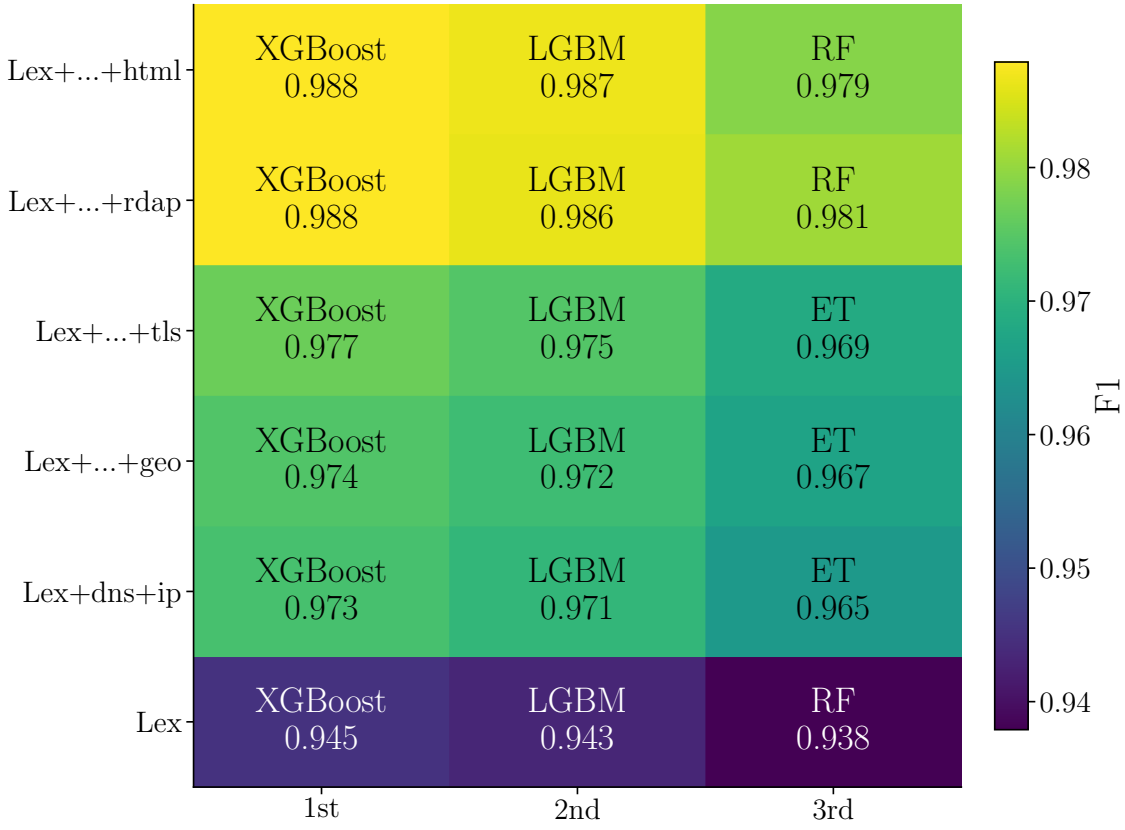
Ve druhé fázi experimentů byly jednotlivé skupiny příznaků agregovány dohromady. Cílem bylo ověřit, jak kombinace více zdrojů informací ovlivňuje klasifikační úspěšnost.




Obrázek 7.2: Výsledky měření samostatných podmnožin – malware.

Agregované výsledky ukazují, že kombinace příznaků napříč kategoriemi výrazně zvyšuje skóre F1 oproti použití samostatných skupin. Nejvyšší dosažené hodnoty *skóre F1* dosáhly 0.9880 pro phishing a 0.9850 pro malware. Zajímavým zjištěním je také to, že přidání poslední skupiny příznaků, tedy HTML, nezvýší přesnost klasifikace.

Pro každou kombinaci příznakových skupin (řádky Lex až Lex+...+html) jsou zobrazeny tři nejlepší klasifikátory (sloupce 1st–3rd). Barevná paleta („heatmap“) indikuje hodnotu skóre F1 – tmavší buňka znamená vyšší skóre. Na buňkách jsou vytištěny zkratky modelů (RF, ADA, DT, ..., ET) a přesné F1 hodnoty na tři desetinná místa. Výsledky agregovaných podmnožin jsou znázorněny na obrázcích 7.3 a 7.4.




Obrázek 7.3: Výsledky měření agregovaných subsetů – phishing.

### 7.2.3 Logické stupňování příznaků

Ve třetí sadě experimentů byla testována strategie postupného rozšiřování vektoru příznaků, vycházející z architektury nástroje `DomainRadar`. Příznaky byly přidávány v následujících fázích:

1. Pouze `lex` příznaky (minimální sběr dat).
2. Kombinace `lex + dns + ip + tls + geo`. Vynechán náročný sběr RDAP a HTML.
3. Plná kombinace všech skupin: `lex + dns + ip + tls + geo + rdap`.

Tento stupňovitý přístup reflektuje implementaci systému Domainradar [42], stejně tak i náročnost sběru jednotlivých skupin příznaků.

Výsledky rozšiřování vstupního vektoru dat hraje velmi významnou roli v přesnosti klasifikace, avšak jen do určité meze. Přidáním HTML příznaků se dle měření přesnost nijak nezvýší.

Výsledky pro tento přístup jsou uvedeny v předchozí sekci, neboť se jedná o podmnožinu všech postupných podmnožin.

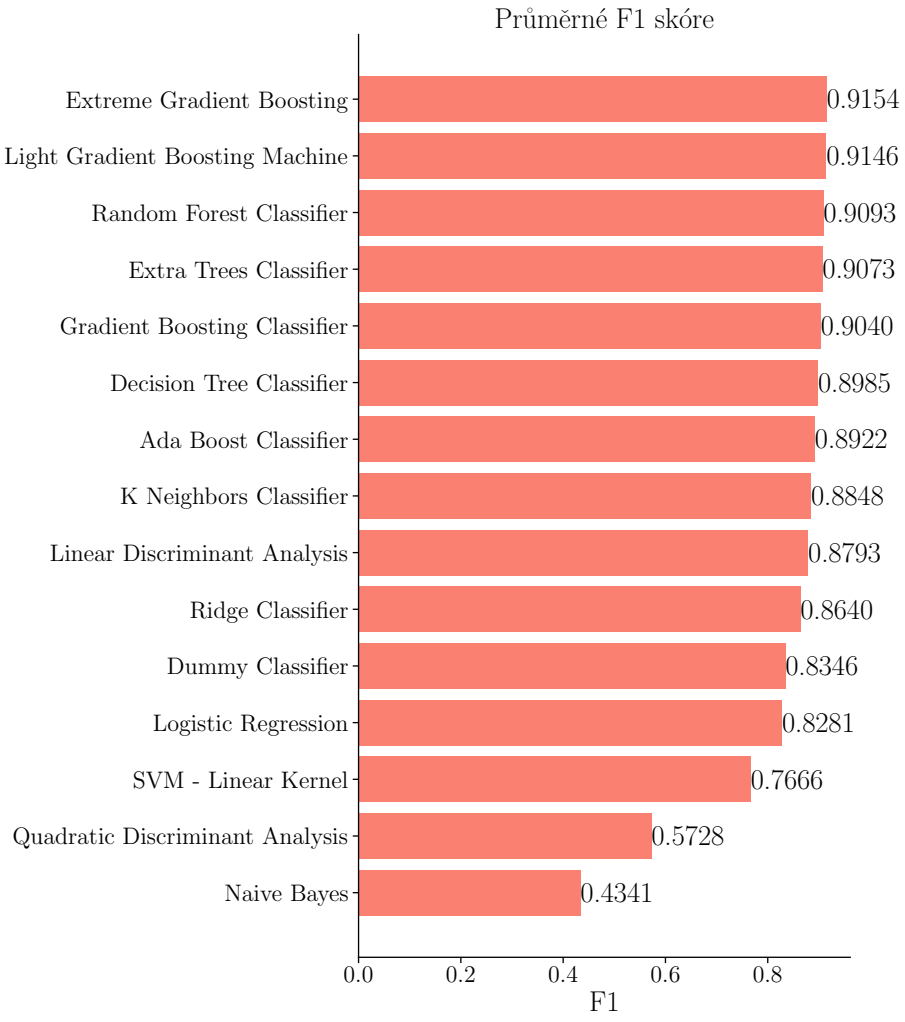

Obrázek 7.4: Výsledky měření agregovaných subsetů – malware.

## 7.3 Shrnutí

Závěry měření jsou tedy následující:

- Samostatné podmnožiny poskytují užitečné informace, ale jejich kombinace výrazně zvyšuje úspěšnost detekce.
- Nejvyššího *skóre F1* bylo dosaženo při použití agregovaných příznaků ze všech dostupných zdrojů.
- Postupné rozšiřování vektoru příznaků podle dostupnosti dat přináší stabilní zlepšování výsledků.
- Nejlepšími klasifikátory napříč všemi experimenty byly **Extreme Gradient Boosting (XGBoost)**, **LightGBM** a **Random Forest**.
- HTML příznaky mají pouze minimální přínos.

Z Obrázku 7.5 je zřejmé, že nejvyšší průměrné skóre F1 dosahují klasifikátory **LightGBM** a **XGBoost**. Tyto dva modely byly proto zvoleny pro další fázi vývoje finálního klasifikačního systému.




Obrázek 7.5: Průměrné skóre F1 jednotlivých klasifikátorů napříč podmnožinami.

# Kapitola 8

# Návrh a implementace klasifikátorů

Účinná detekce maligních domén vyžaduje promyšlenou kombinaci různých klasifikačních přístupů, které dokáží pružně reagovat na variabilitu dostupných dat a rychle se měnící charakter kybernetických hrozeb. V této kapitole je podrobně popsán návrh modulární klasifikační pipeline, která zahrnuje více úrovní rozhodování a využívá kombinovaný přístup ke zvýšení přesnosti a robustnosti detekce.

Cílem kapitoly je představit jednotlivé modely, které tvoří stavební bloky této pipeline – včetně stromových algoritmů (XGBoost, LightGBM), neuronových sítí (FFNN, CNN) i specializovaných modulů (TLS klasifikátor, metamodel, FPD). Zvláštní důraz je kladen na to, jak jsou tyto modely navrženy, trénovány a integrovány do víceúrovňového systému rozhodování.

Zásadní roli hraje rozdělení vstupních domén podle množství dostupných příznaků do tří úrovní (*stages*). Každé úrovni odpovídá specifická sada klasifikátorů optimalizovaných na daný rozsah informací. Výsledky modelů jsou následně agregovány pomocí metaklasifikátoru. Pro zvýšení spolehlivosti je do procesu začleněn i doplňkový modul pro detekci falešně pozitivních případů (FPD), který dále filtruje podezřelé predikce. Tato kapitola zahrnuje:

- Detailní popis jednotlivých klasifikačních modelů, jejich architektur a použitých technik trénování.
- Vysvětlení principu vícestupňové klasifikace podle dostupnosti dat a způsobu výběru příslušných modelů.
- Popis strategie agregace výstupů více modelů pomocí vážení, hlasování nebo metaklasifikace.
- Integraci modulu FPD pro dodatečnou eliminaci falešně pozitivních detekcí.
- Optimalizační strategie pro výběr hyperparametrů a zvýšení robustnosti klasifikátorů.

Všechny komponenty byly navrženy s ohledem na škálovatelnost, adaptabilitu a snadnou integraci do reálného bezpečnostního systému. Výsledná pipeline umožňuje flexibilní klasifikaci i v případech s neúplnými daty a minimalizuje riziko chybné detekce díky kombinaci více rozhodovacích vrstev. Kapitola tak nabízí ucelený pohled na návrh pokročilého systému pro detekci škodlivých domén a poskytuje rámec pro jeho další rozvoj či nasazení v praxi.

## 8.1 XGBoost

V této sekci je popsán návrh, implementace a konfigurace modelu XGBoost pro detekci maligních domén. Samotné experimentální výsledky dosažené tímto modelem jsou prezentovány v kapitole 9, konkrétně v sekci 9.4.

XGBoost (Extreme Gradient Boosting) je výkonný open-source algoritmus pro gradient boosting, který si získal významné postavení v oblasti strojového učení díky své vysoké efektivitě, přesnosti a schopnosti práce s rozsáhlými a nevyváženými datovými sadami. Klíčovými vlastnostmi jsou podpora paralelních výpočtů, možnost zabudované regularizace (L1, L2) a flexibilní správa paměti, což je činí ideálním pro úlohy detekce škodlivých domén [18].

### 8.1.1 Předzpracování dat

Pro trénování modelu XGBoost byla využita sada transformací popsaná v sekci 5.6.1. Vstupní data byla upravena následujícím způsobem:

- Odstranění chybějících hodnot (nahrazení nulou).
- Převedení binárních příznaků na celočíselné hodnoty (0/1).
- Min-max škálování všech numerických atributů do intervalu $[0, 1]$.
- Odstranění extrémních hodnot podle pravidla založeného na odchylce od průměru.

Tyto kroky zajistily stabilní a konzistentní vstupní reprezentaci vhodnou pro trénování stromových modelů bez výrazného zkreslení distribuce příznaků.

### 8.1.2 Architektura a hyperparametry modelu

Model XGBoost byl konfigurován s následujícími hyperparametry:

- **Počet stromů (n_estimators)**: `300`
- **Maximální hloubka stromu (max_depth)**: `9`
- **Rychlost učení (learning_rate)**: `0.023`
- **Subsampling poměr (subsample)**: `0.8`
- **Regulace L2 (lambda)**: `1.5`
- **Minimální váha listu (min_child_weight)**: `1.0`

Hyperparametry byly optimalizovány pomocí Bayesovské optimalizace na validační sadě s cílem maximalizovat skóre F1. Tato metoda umožnila efektivně prohledat prostor parametrů a nalézt nastavení vedoucí k optimálním výsledkům i při relativně omezeném počtu experimentů [85].

Jednou z hlavních výhod použití XGBoost v této aplikaci je jeho schopnost efektivně pracovat s nerovnováhou tříd v datech a vysoká odolnost vůči šumu. Díky interní implementaci L1 a L2 regularizace model dosahuje lepší generalizace, což je kritické v dynamicky se měnících podmínkách detekce kybernetických hrozeb [18].

Model XGBoost zároveň umožňuje snadnou interpretaci výsledků pomocí výstupů významnosti jednotlivých příznaků (feature importance), což je důležité při dalším ladění a zlepšování detekčních mechanismů [56].

#### Výhody XGBoost v detekci domén

XGBoost se vyznačuje vysokou škálovatelností ve všech scénářích, což je zvláště důležité v oblasti detekce domén, kde se často pracuje s rozsáhlými a různorodými datovými sadami. Paralelní a distribuované výpočty, které algoritmus podporuje, umožňují rychlejší učení a zpracování velkých objemů dat, což je klíčové pro efektivní identifikaci potenciálně škodlivých nebo podezřelých doménových jmen. [18].

## 8.2 LightGBM

Tato sekce se zaměřuje na návrh, implementaci a konfiguraci modelu LightGBM pro úlohu detekce maligních domén. Experimentální výsledky získané tímto modelem jsou uvedeny v kapitole 9 v sekci 9.3.

LightGBM (Light Gradient Boosting Machine) je efektivní algoritmus založený na gradient boostingu, který je optimalizován pro rychlé trénování a nízkou spotřebu paměti. Díky své schopnosti zpracovávat velké objemy dat a práci se sparse strukturami je vhodným kandidátem pro aplikace v oblasti detekce kybernetických hrozeb. [49]

### 8.2.1 Předzpracování dat

Předzpracování dat pro model LightGBM vycházelo ze stejných principů jako u XGBoost, v souladu s transformacemi uvedenými v sekci 5.6.1. Aplikovány byly následující kroky:

- Náhrada chybějících hodnot nulou.
- Převedení binárních atributů na numerickou reprezentaci.
- Normalizace všech numerických příznaků metodou min-max.
- Odstranění odlehlých hodnot na základě směrodatné odchylky.

Tím bylo dosaženo jednotného formátu vstupních dat, který minimalizuje zkreslení a zvyšuje robustnost modelu při trénování.

### 8.2.2 Architektura a hyperparametry modelu

Model LightGBM byl trénován s následující konfigurací hyperparametrů:

- **Počet stromů (n_estimators)**: `600`
- **Maximální hloubka stromu (max_depth)**: `12`
- **Rychlost učení (learning_rate)**: `0.0978`
- **Subsampling poměr (subsample)**: `0.596`
- **Regulace L2 (lambda_l2)**: `2.07`
- **Minimální váha listu (min_child_weight)**: `1.0`
- **Metoda vzorkování (sampling_method)**: `gradient_based`
- **Velikost koše (max_bin)**: `512`

Parametry byly zvoleny na základě zkušeností z ladění XGBoost modelu, s drobnými úpravami pro optimalizaci **LightGBM**. Díky použití metody *gradient-based sampling* model efektivněji vybírá vzorky během trénování, což umožňuje rychlejší konvergenci bez výrazné ztráty přesnosti [49].

Významnou výhodou LightGBM je také jeho schopnost efektivně pracovat se **sparse** daty a podporovat **kategorické atributy** nativně, bez nutnosti jejich explicitní transformace [49]. To umožňuje jednodušší a rychlejší předzpracování v budoucích aplikacích.

Navíc díky technikám jako *leaf-wise tree growth* dosahuje LightGBM **vyšší predikční přesnosti** než tradiční boostingové algoritmy, a to zejména v případě komplexních a vysoce nelineárních úloh, jaké jsou typické při detekci škodlivých domén [49, 95].

## 8.3 Metoda podpůrných vektorů (SVM)

Tato sekce popisuje návrh, implementaci a optimalizaci klasifikačního modelu SVM, který byl využit pro detekci maligních domén. Důraz je kladen na přípravu dat, konfiguraci modelu a detailní popis metod použitých pro ladění hyperparametrů. Výsledky trénování a testování jsou následně analyzovány v kapitole 9.

### 8.3.1 Předzpracování dat

Pro model Support Vector Machine (SVM) byla použita obecná schémata transformace popsaná v sekci 5.6.1. Důraz byl kladen na odstranění vlivu rozdílných měřítek příznaků, protože modely SVM jsou na tyto rozdíly citlivé. Aplikované kroky zahrnovaly:

- Nahrazení chybějících hodnot nulou.
- Převedení binárních příznaků na celočíselné hodnoty.
- Min-max škálování do rozsahu $[0, 1]$.
- Odstranění odlehlých hodnot pomocí pravidla založeného na $2\sigma$ odchylce.

Tato forma předzpracování umožnila modelu efektivně pracovat s daty bez zkreslení vlivem extrémních hodnot nebo nesrovnatelných rozsahů.

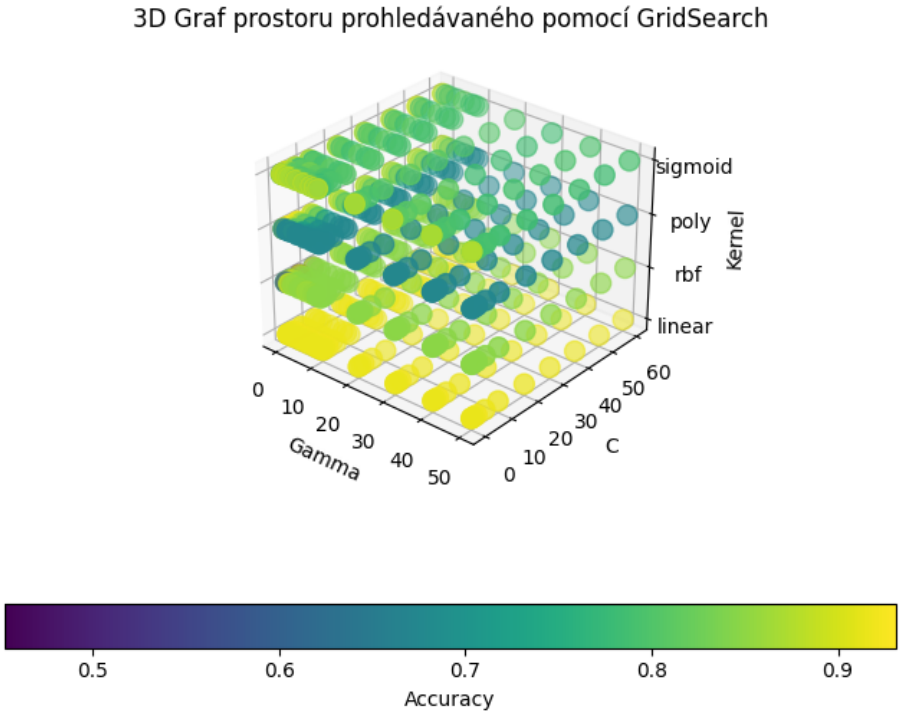
### 8.3.2 Architektura modelu a výběr hyperparametrů

Model byl implementován s využitím jádra s radiální bázovou funkcí (*RBF kernel*). Vstupní parametry byly nastaveny následovně:

- **Regularizační parametr** $C$: `59`
- **Parametr** $\gamma$: `0.1`
- **Typ jádra**: `rbf`
- **Vyvážení tříd**: `balanced`
- **Náhodný stav**: `42`

Optimalizace parametrů byla provedena pomocí **GridSearch**, jehož cílem bylo maximalizovat hodnotu *skóre F1* na validační množině. Přestože GridSearch představuje systematický přístup k výběru parametrů, jeho hlavní nevýhodou je značná výpočetní náročnost a nemožnost efektivního pokrytí celého parametrového prostoru při jeho velkém rozsahu. [13]. Vizualizaci prohledávaného prostoru pak můžeme nalézt na obrázku 8.1.

Z tohoto důvodu byla v rámci této práce implementována vlastní metoda **GradientGridSearch**, která kombinuje výhody GridSearch s principy gradientního sestupu. Tato metoda předpokládá, že hodnoty blízké optimu poskytují podobně dobré výsledky a zaměřuje se na postupné zjemňování prostoru hledání. Implementace je uvedena v algoritmu 15.




Obrázek 8.1: Vizualizace výsledků metodou GridSearch při hledání optimálních parametrů pro SVM.

### 8.3.3 Práce s nevyváženými daty

Vzhledem k charakteru úlohy, kde většina domén je benigní a pouze malá část maligní, je datová sada značně nevyvážená. Tento stav vede ke zkreslení modelu směrem k majoritní třídě. Proti tomuto efektu byla nasazena následující opatření:

- Použití volby `class_weight=balanced` v rámci SVM.
- Úprava rozhodovacího prahu klasifikace.

- Testování různých metod převažování (např. *SMOTE*) – i když nebyly přímo nasazeny v konečné verzi modelu, posloužily jako důležitý kontrolní experiment.

Důvodem těchto opatření je snížení výskytu falešně pozitivních klasifikací, které by mohly vést k chybnému zablokování legitimních domén.

### 8.3.4 Specifika SVM v doménové klasifikaci

Model SVM se ukázal jako vhodný především pro klasifikaci podle lexikálních příznaků, které často nesou diskriminační hodnotu i v případech, kdy chybí kontextové či síťové informace. V rámci budoucího výzkumu se nabízí využití tohoto modelu jako součásti ansámblového klasifikátoru, nebo jeho kombinace s online učením, jak ukazuje například přístup *Feedback-SVM (F-SVM)* [97].

### 8.3.5 Gradient Grid Search

Gradient Grid Search je metoda optimalizace výběru parametrů pro SVM (Support Vector Machine), inspirovaná gradientním sestupem.

Princip fungování algoritmu spočívá v iterativním přibližování k optimálním parametrům. Vnější smyčka kontroluje konvergenci a řídí globální pokrok algoritmu, zatímco vnitřní smyčka generuje nové kombinace parametrů na základě aktuálně nejlepší známé sady a provádí jejich evaluaci. Pokud nová kombinace přinese lepší výsledek, je tato kombinace akceptována jako nové výchozí řešení a následné kroky se orientují v jejím okolí. S každou iterací se také zmenšuje krok (parametr `alpha`), čímž se zpřesňuje prohledávání.
Tímto způsobem se efektivně zúží hledaný prostor a zrychluje se nalezení optimálních parametrů. Průběh je formálně znázorněn v algoritmu 15.

**Algorithm 1:** Algoritmus Gradient Grid Search

```
1  function GGS;
   Input     : Initial SVM parameters, performance metrics
   Output    : Optimized SVM parameters
   Initialize: Set alpha (step size), delta coefficient (new parameters factor)
2  Compute metrics for given parameters;
3  Set the best combination of parameters as current;
4  while not converged do
5      Generate a set of new parameters (delta) from the best combination;
6      for each parameter in delta do
7          Set parameter with step alpha;
8          Compute metrics for new parameter;
9          if new parameter's metric is better than current then
10             Set new parameter as current;
11             Update best combination of parameters;
12         end
13     end
14     Reduce alpha for next iteration;
15 end
```

**Analýza časové složitosti**

Analýza časové složitosti Gradient Grid Search (GGS) ukazuje, že je řízena hloubkou prohledávání a počtem iterací $k$ pro dosažení konvergence. Pro každou iteraci zkoumá $d$ nových kombinací parametrů, což vede k složitosti $O(d \cdot k)$. Naproti tomu, pro tradiční GridSearch s $n$ parametry a $m$ možnými hodnotami pro každý parametr je složitost $O(m^n)$.

$$\text{Složitost GridSearch: } O(m^n) \tag{8.1}$$
$$\text{Složitost GGS: } O(d \cdot k) \tag{8.2}$$
$$\text{Dolní hranice GGS: } O(d) \tag{8.3}$$
$$\text{Horní hranice GGS: } O(d \cdot k_{\max}) \tag{8.4}$$

Zde $k_{\max}$ označuje maximální počet iterací konvergence. Díky tomu, že GGS adaptivně zužuje prohledávaný prostor na základě dřívějších výsledků, je ve většině případů výrazně efektivnější než klasické plošné prohledávání.

Pro ilustraci uvažujme konkrétní případ optimalizace tří parametrů SVM: typu jádra (`kernel`) s 3 možnými hodnotami (lineární, polynomiální, RBF) a spojitých parametrů $C$ a $\gamma$, které jsou v GridSearch discretizovány na 10 hodnot každé. Celkem by GridSearch musel vyhodnotit $3 \times 10 \times 10 = 300$ kombinací.

Naopak při použití GGS bychom mohli začít s $d = 6$ počátečními kombinacemi (např. dvě pro každý typ jádra) a v každé iteraci vyhodnotit 6 nových kombinací v jejich okolí. Pokud by k dosažení konvergence stačilo $k = 5$ iterací, vedlo by to k celkovému počtu $6 \cdot 5 = 30$ vyhodnocených kombinací. Ve srovnání s 300 iteracemi GridSearchu by tedy GGS vyžadoval jen 10 % výpočetního času při podobné kvalitě výsledného modelu.

Tato úspora je podmíněna tím, že optimální oblast parametrů skutečně leží v lokalizované části prostoru a že optimalizační funkce (typicky metrika výkonnosti modelu, např. skóre F1 nebo přesnost) je *lokálně unimodální* a dostatečně hladká v okolí optima. Jinými slovy, Gradient Grid Search implicitně předpokládá, že tato funkce má v dané oblasti **monotonní gradient** směrem k (lokálnímu) maximu a neobsahuje výrazné oscilace, které by mohly narušit konvergenci. Za těchto podmínek může být metoda velmi efektivní, protože postupné prohledávání prostoru parametrů vede ke zlepšování výstupu.

Je však důležité zdůraznit, že GGS nenabízí záruku nalezení globálního optima. Vzhledem k deterministickému a lokálně orientovanému charakteru této metody může v nehladkých nebo vícemodálních funkcích (s více lokálními extrémy) snadno dojít k uvíznutí v suboptimálním řešení. V takových případech je vhodné metodu rozšířit o náhodnou inicializaci, restartování nebo hybridizaci s globálními heuristikami (např. simulated annealing, evoluční algoritmy), které umožní únik z lokálních extrémů a důkladnější prohledání celého prostoru.

## 8.4 Neuronové sítě

Ačkoliv neuronové sítě představují velmi výkonnou třídu modelů pro detekci složitých vzorců v datech, nebyly zahrnuty do předběžné analýzy podmnožin příznaků 7, jelikož nejsou podporovány přímo v knihovně `PyCaret`, která byla použita pro automatizované testování klasifikátorů. Navíc jejich použití vyžaduje specifické předzpracování dat, například formátování do sekvenčních nebo obrazových struktur, normalizaci a úpravu dimenzionality vstupních atributů.

Přestože tento přístup neumožňuje přímé porovnání neuronových sítí s modely jako XGBoost nebo LightGBM na úrovni frameworku PyCaret, jejich výjimečná schopnost modelovat nelineární vztahy a identifikovat komplexní struktury v datech z nich činí silné kandidáty pro detekci maligních domén.

V rámci této práce byly navrženy a experimentálně ověřeny dva typy neuronových sítí, přičemž každý z nich reflektuje specifický způsob reprezentace dat:

- **Konvoluční neuronové sítě (CNN)** – aplikovány na 2D transformované atributové vektory, kde doménové vlastnosti jsou převedeny do podoby obrazu. CNN jsou schopné zachytit prostorové vzory a lokální korelace mezi příznaky. Podobný přístup byl úspěšně aplikován například v práci Silveira et al. [84].
- **Klasické plně propojené neuronové sítě (Feedforward NN)** – využívají vektorovou reprezentaci doménových atributů bez strukturální transformace. Tento typ sítě slouží jako referenční architektura pro porovnání efektivity hlubokého učení na původní podobě dat [6].

Každý z těchto přístupů je v následujících podsekcích detailně popsán včetně architektury, způsobu trénování, předzpracování dat a dosažených výsledků.

## 8.5 Feedforward neuronová síť (FFNN)

Plně propojené neuronové sítě (Feedforward Neural Networks, FFNN) představují základní architekturu hlubokého učení, kde se informace šíří jedním směrem – od vstupních vrstev přes skryté vrstvy až k výstupní vrstvě. Přestože tyto sítě neobsahují rekurentní ani konvoluční prvky, při správném návrhu mohou efektivně modelovat složité vztahy v datech, a to zejména v případech, kdy jsou doménová data již vhodně předzpracovaná [32].

V této sekci je popsán návrh a implementace plně propojené neuronové sítě určené pro klasifikaci maligních domén. Experimentální výsledky dosažené tímto modelem jsou uvedeny v kapitole 9, sekce 9.5.

### 8.5.1 Předzpracování dat

Pro trénování plně propojené neuronové sítě (FFNN) byly využity transformační postupy popsané v sekci 5.6, konkrétně kombinace obecných úprav a dodatečné sigmoidní transformace vhodné pro neuronové architektury.

- Základní normalizace a čištění: nahrazení chybějících hodnot, převod binárních příznaků, min-max škálování.
- Odstranění odlehlých hodnot podle směrodatné odchylky.
- Aplikace sigmoidní transformace na vstupní atributy (viz sekce 5.6.2), čímž byly hodnoty převedeny do hladkého intervalu $(0, 1)$.

Použití sigmoidní funkce napomohlo stabilizaci gradientů a přispělo ke stabilnějšímu a rychlejšímu trénování neuronové sítě.

### 8.5.2 Architektura feedforward sítě

Navržená architektura plně propojené neuronové sítě obsahuje čtyři skryté vrstvy s postupným snižováním dimenze a pravidelnou aplikací normalizace a nelineární aktivační funkce ReLU. Cílem je umožnit efektivní učení i v případě vysoce dimenzionálních vstupních atributů.

Architektura je specifikována následujícím způsobem:

- **Vstupní data:** Model očekává standardní vektor vstupních příznaků o velikosti odpovídající počtu atributů popisujících každou doménu.
- **První skrytá vrstva:** 1024 neuronů, normalizace (Batch Normalization) a aktivace ReLU.
- **Druhá skrytá vrstva:** 512 neuronů, normalizace a aktivace ReLU.
- **Třetí skrytá vrstva:** 256 neuronů, normalizace a aktivace ReLU.
- **Čtvrtá skrytá vrstva:** 128 neuronů, normalizace a aktivace ReLU.
- **Výstupní vrstva:** 1 neuron s aktivační funkcí sigmoid, produkující pravděpodobnostní skóre.

Implementace modelu je realizována pomocí knihovny `TensorFlow` a je definována následovně:

```
def build_feedforward_net(feature_size):
    inputs = Input(shape=(feature_size,))

    x = Dense(1024, activation=None)(inputs)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    x = Dense(512, activation=None)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    x = Dense(256, activation=None)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    x = Dense(128, activation=None)(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

    outputs = Dense(1, activation='sigmoid')(x)

    model = Model(inputs=inputs, outputs=outputs, name="feedforward_net")
    return model
```

### 8.5.3 Trénování a optimalizace

**Optimalizace parametrů:** Model je trénován s použitím optimalizátoru **Adam** [50] s nastavenou počáteční rychlostí učení 0.0023.

Jako ztrátová funkce je použita `binary_crossentropy`, která je standardem při binární klasifikaci.

**Postup trénování:** Data byla rozdělena do trénovací a validační části v poměru 70:30 s použitím stratifikovaného rozdělení, aby byl zachován poměr tříd. Model byl trénován na dávkách o velikosti 512 vzorků (`batch_size=512`) s maximálním počtem 25 epoch.

Pro dosažení robustních výsledků byla implementována technika **Early Stopping** s monitorováním validační ztráty (`val_loss`) a automatickým obnovením nejlepších vah při poklesu výkonu.

### 8.5.4 Shrnutí architektury

Plně propojená neuronová síť představuje jednoduchý, ale výkonný přístup pro klasifikaci doménových dat, pokud jsou vstupy vhodně předzpracovány a škálovány. Díky postupnému snižování dimenze a použití normalizace po každé vrstvě je zajištěno stabilní a efektivní učení i při vyšším počtu atributů. Použití aktivační funkce ReLU dále umožňuje efektivní propagaci gradientů a rychlejší konvergenci během trénování.

## 8.6 Konvoluční neuronová síť (CNN)

Konvoluční neuronové sítě (CNN), tradičně aplikované v oblasti zpracování obrazu, nacházejí stále širší uplatnění i v jiných doménách, kde se setkáváme s potřebou rozpoznání složitých vzorů a struktur v datech. Jednou z takových aplikací je klasifikace doménových jmen, kde CNN umožňují efektivně identifikovat vzory indikující potenciální škodlivost domény. Příkladem publikace, kde je demonstrováno využití CNN pro zpracování neobrazových dat, je práce Silveira et al. [84], která se zabývá detekcí nově registrovaných škodlivých domén s využitím pasivního DNS.

### 8.6.1 Předzpracování dat

Pro konvoluční neuronovou síť bylo nezbytné provést transformaci vstupních vektorů do dvourozměrné podoby, jak je popsáno v sekci 5.6.2. Celkový proces předzpracování zahrnoval:

- Aplikaci obecných transformačních kroků: nahrazení chybějících hodnot, škálování, odstranění odlehlých hodnot.
- Sigmoidní transformaci vstupních příznaků pro lepší rozložení aktivací.
- Převod každého vstupního vektoru na čtvercovou matici o rozměru $s \times s$, kde každý prvek odpovídá jednomu příznaku.

Výsledná reprezentace umožnila zpracování dat pomocí konvolučních filtrů a umožnila modelu detekovat prostorové vzory a shluky v příznacích domén.

### 8.6.2 Implementace a architektura CNN

Implementace konvoluční neuronové sítě (CNN) pro účely klasifikace doménových dat byla založena na definici dvou konvolučních vrstev a dvou plně propojených vrstev. Tato struktura umožňuje efektivně zpracovávat vstupní data a extrahovat z nich relevantní charakteristiky pro klasifikaci. Vstupní data jsou nejprve zpracovávána konvolučními vrstvami, kde každá vrstva aplikuje filtry pro detekci vzorů a následně používá nelineární aktivační funkci ReLU. Konvoluční vrstvy jsou definovány následovně:

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, 3, 1)
        self.conv2 = nn.Conv2d(32, 64, 3, 1)
        self.fc1 = nn.Linear(64 * (side_size-4)**2, 128)
        self.fc2 = nn.Linear(128, 256)
        self.fc3 = nn.Linear(256, 2)
```

První konvoluční vrstva (`conv1`) s 32 filtry a druhá konvoluční vrstva (`conv2`) s 64 filtry postupně zvyšují hloubku vstupních dat a umožňují modelu identifikovat složitější vzory. Po konvolučních vrstvách jsou data zploštěna a zpracovávána třemi plně propojenými vrstvami (`fc1`, `fc2` a `fc3`), které slouží k finální klasifikaci.

Proces trénování modelu zahrnuje iterativní optimalizaci s použitím funkce ztráty cross-entropy a Adam optimalizátorutextbfAdam [50]. Trénování probíhá v několika epochách, přičemž v každé epoše jsou data prezentována modelu v náhodném pořadí, což zlepšuje generalizaci modelu.

## 8.7 Výsledná klasifikační pipeline

V rámci této práce byla navržena modulární klasifikační pipeline, která umožňuje efektivní detekci maligních domén na základě dostupných příznaků. Pipeline je koncipována tak, aby byla schopna adaptivně reagovat na různou míru informací dostupných o jednotlivých doménách a využívala specializované modely optimalizované pro různé úrovně datové úplnosti.

### 8.7.1 Tři stupně klasifikace podle dostupnosti dat

Na základě analýzy dostupných datových atributů byly definovány tři stupně klasifikace:

- **Stage 1** – Minimální množství příznaků: pouze lexikální příznaky.
- **Stage 2** – Střední množství příznaků: rozšířené atributy, např. síťová a certifikační metadata.
- **Stage 3** – Kompletní sada příznaků: zahrnuje všechny dostupné informace.

Každý stupeň je pak pokryt sadou modelů. Schéma je pak na obrázku 8.2

### 8.7.2 Mechanismus výběru vhodného stupně

Při klasifikaci nové domény pipeline nejprve analyzuje dostupné příznaky. Na základě jejich počtu a typu je automaticky vybrán nejvhodnější stupeň (*stage*). Pokud není možné využít model vyššího stupně, pipeline degraduje rozhodování na nižší stupeň.

Výběr stupně je realizován zaokrouhlováním podle počtu dostupných příznaků tak, aby bylo možné použít model optimalizovaný pro danou situaci.

### 8.7.3 Paralelní klasifikace pomocí více modelů

Pro každý stupeň jsou k dispozici různé modely (např. XGBoost, Feedforward NN, CNN). Klasifikace probíhá následovně:

1. Vyberou se všechny modely odpovídající zvolenému stupni.
2. Každý model samostatně předpoví pravděpodobnost malignity.
3. Výsledky jsou agregovány — Metody váhování výstupů jednotlivých modelů jsou rozebrány v sekci 8.8 níže.

Tento přístup umožňuje kombinovat silné stránky různých modelů a zvyšuje robustnost výsledné predikce.

### 8.7.4 Výhody zvoleného přístupu

Navržená pipeline přináší následující výhody:

- **Adaptivita:** Funkčnost i při neúplných datech díky dynamickému výběru stupně.
- **Robustnost:** Snížení rizika chybných klasifikací díky kombinovanému přístupu.
- **Flexibilita:** Možnost snadného rozšíření o nové modely a strategie agregace.
- **Modularita:** Jasné oddělení jednotlivých komponent (výběr stupně, modely, agregace).

Díky této koncepci je pipeline vhodná jak pro akademické experimenty, tak pro nasazení v reálném prostředí s proměnlivou kvalitou vstupních dat.

## 8.8 Váhování ve výsledné pipeline

Při paralelní klasifikaci pomocí více modelů v rámci jednotlivých stupňů (*stages*) vzniká potřeba vhodně agregovat jejich parciální výstupy. Cílem je získat spolehlivější a robustnější odhad pravděpodobnosti malignity každé domény. V této sekci popisujeme několik strategií váhování a agregace rozhodnutí modelů, které byly zvažovány nebo implementovány v rámci této práce.

### 8.8.1 Výběr nejlepšího modelu

Nejjednodušší přístup spočívá ve využití výstupu jediného modelu, který dosahuje nejlepších výsledků v rámci validační sady. Tento model je pak považován za reprezentativní pro daný stupeň klasifikace.

- **Výhody:** Nízká výpočetní náročnost, snadná interpretace.
- **Nevýhody:** Ztráta potenciálu ensemble přístupu; náchylnost k přeučení a variabilitě výkonu při změně dat.

### 8.8.2 Nevážený aritmetický průměr výstupů

Každý model generuje pravděpodobnostní výstup (např. pravděpodobnost, že doména je maligní). Tyto hodnoty se následně zprůměrují a výsledná hodnota slouží jako vstup pro rozhodovací práh.

$$P_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} P_i$$

kde $P_i$ je výstup $i$-tého modelu a $N$ je počet modelů.

- **Výhody:** Robustní vůči extrémním hodnotám jednoho modelu, jednoduchá implementace.
- **Nevýhody:** Předpokládá stejnou spolehlivost všech modelů.

### 8.8.3 Vážený průměr dle výkonnosti modelů

Modelům jsou přiřazeny váhy $w_i$ na základě jejich validační výkonnosti (např. skóre F1, AUC). Výstupy jsou poté agregovány jako vážený průměr:

$$P_{\text{weighted}} = \frac{\sum_{i=1}^{N} w_i \cdot P_i}{\sum_{i=1}^{N} w_i}$$

- **Výhody:** Reflektuje rozdíly v kvalitě modelů, zvyšuje důraz na spolehlivější prediktory.
- **Nevýhody:** Nutnost udržovat a aktualizovat váhy při každém tréninku.

### 8.8.4 Rozhodovací metamodel (meta-klasifikátor)

Další možností je použití metamodelu, který se učí na výstupech základních modelů. Každý model poskytuje vstupní znak a metamodel (např. logistická regrese, rozhodovací strom) učí optimální váhování a rozhodovací pravidla.

- **Výhody:** Možnost modelovat nelineární vztahy mezi výstupy modelů, adaptivita vůči korelaci mezi modely.
- **Nevýhody:** Vyšší složitost, riziko přeučení, potřeba separátní validační množiny pro trénink metamodelu.

### 8.8.5 Většinové hlasování

Při použití binárních rozhodnutí (např. výstup $P_i > 0{,}5$) je výsledná predikce určena většinou hlasů modelů.

$$\hat{y} = \text{mode}\left(\mathbb{I}[P_i > \theta]\right)$$

kde $\mathbb{I}$ je indikátorová funkce a $\theta$ rozhodovací práh (typicky 0,5).

- **Výhody:** Vhodné pro situace, kdy je požadována interpretovatelná binární volba.
- **Nevýhody:** Ignoruje nuance pravděpodobností, nevhodné při různé kalibraci modelů.

### 8.8.6 Bayesovská agregace

V teoretickém rámci lze každý model považovat za samostatný zdroj podmíněné pravděpodobnosti. Kombinace modelů pak může být řešena pomocí Bayesova pravidla nebo jeho aproximací (např. pomocí Naivního Bayese nad výstupy). [10]

- **Výhody:** Formální pravděpodobnostní rámec, možnost inkorporace apriorních znalostí.
- **Nevýhody:** Obtížná aplikace bez přesné znalosti závislostí mezi modely.

### 8.8.7 Souhrn

Výběr strategie váhování závisí na konkrétních požadavcích systému — zda má být preferována interpretovatelnost, přesnost, robustnost nebo výpočetní efektivita. V této práci byl primárně využit aritmetický průměr a dále testován vážený průměr dle výkonnosti modelů. Tyto přístupy poskytly dobrý kompromis mezi přesností a jednoduchostí implementace, zejména v kontextu reálného nasazení s omezenými výpočetními prostředky.

Přehled vhodnosti jednotlivých strategií ve vztahu k různým systémovým prioritám shrnuje tabulka 8.1.

| **Metoda** | **Složit.** | **Robust.** | **Přesn.** | **Interp.** | **Adapt.** |
|---|---|---|---|---|---|
| Nejlepší model | + | – | – | + | – |
| Nevážený průměr | + | + | + | + | – |
| Vážený průměr | o | + | + | o | o |
| Metamodel | – | + | + | – | + |
| Hlasování | + | o | – | + | – |
| Bayes. agregace | – | + | o | o | o |

Tabulka 8.1: Vhodnost váhovacích strategií pro různé požadavky systému.

**Legenda:** + vhodné, o částečně vhodné, – nevhodné. Sloupce: Složitost implementace, robustnost vůči chybám modelů, predikční přesnost, interpretovatelnost výsledku, adaptabilita na měnící se podmínky.

## 8.9 Rozhodovací neuronová síť (meta-klasifikátor)

Aby bylo možné efektivně kombinovat výstupy různých modelů zapojených do klasifikační pipeline, byla navržena a implementována specializovaná rozhodovací neuronová síť. Rozhodnutí o implementaci váhovací sítě vychází z měření v sekci 9.8.

Tato neuronová síť slouží jako meta-klasifikátor, který se učí na základě výstupů jednotlivých klasifikátorů a dokáže tak vytvořit finální rozhodnutí s vyšší přesností a robustností.

### 8.9.1 Motivace a účel

Ensemble přístup zvyšuje spolehlivost predikce kombinací několika modelů. Nicméně jednoduché metody jako vážený průměr nemusí vždy efektivně zohlednit vzájemné korelace nebo specifické vzory v chování jednotlivých klasifikátorů. Proto byla implementována neuronová síť, která se učí optimální váhování výstupů modelů a dokáže detekovat i složité nelineární vztahy mezi nimi.

### 8.9.2 Vstupy do metamodelu

Model přijímá jako vstup vektor pravděpodobnostních skóre jednotlivých klasifikátorů odpovídajících zvolenému klasifikačnímu stupni (např. výstupy modelů XGBoost, FFNN, CNN pro Stage 3). Vstupy mohou dále zahrnovat další metadata, například:

- Počet příznaků dostupných pro danou doménu
- Výsledky TLS klasifikátoru
- Entropii vstupního vektoru

### 8.9.3 Architektura rozhodovací neuronové sítě

Model je implementován jako plně propojená neuronová síť s následující architekturou:

```
Sequential([
Dense(32, input_dim=n_features, activation="relu"),
Dropout(0.2),
Dense(16, activation="relu"),
Dense(1, activation="sigmoid"),
])
```

Model je trénován s použitím binární entropie jako ztrátové funkce a optimalizátoru Adam (learning rate 0.001). Trénování probíhá s validačním dělením dat 70/30 a využitím early stopping pro prevenci přeučení.

### 8.9.4 Výhody a výsledky

Rozhodovací neuronová síť přináší vyšší flexibilitu a schopnost zohlednit komplexní interakce mezi modely. V porovnání s jednoduchými váhovacími metodami (aritmetický průměr, vážený průměr) dosahovala mírně vyšší skóre F1 a lépe odolávala výkyvům v kvalitě výstupů jednotlivých modelů.

Z hlediska výkonu je model dostatečně rychlý pro online inferenci v provozním prostředí a lze jej dále rozšířit o více vstupních znaků, např. entropii predikcí nebo volatilitu předchozích rozhodnutí.

Model byl validován jako součást hlavní pipeline a jeho přínos byl analyzován v kapitole 9, sekce 9.8.

### Poznámka k implementaci a zařazení

Rozhodovací neuronová síť byla implementována jako jedna z několika strategií váhování výstupů modelů (viz kapitola 8.8). Z pohledu přesnosti a robustnosti se ukázala jako nejvhodnější varianta, a proto byla využita jako finální agregátor v navržené pipeline. Implementace využívá framework TensorFlow/Keras a plně propojenou architekturu popsanou výše. Detailní vyhodnocení je uvedeno v sekci 9.8.

## 8.10 Detekce falešně pozitivních vzorků

Při provozním nasazení systému *DomainRadar* je klíčové minimalizovat počet falešně pozitivních detekcí (FP), tedy situací, kdy benigní doména byla chybně označena jako maligní. I při vysoké přesnosti klasifikátorů mohou i jednotky procent FP znamenat stovky nebo tisíce chybných alertů za minutu — vzhledem k tomu, že systém zpracovává statisíce domén každou minutu. Takové množství varování je pro lidské operátory neudržitelné a vede k ignorování detekcí nebo přetížení bezpečnostních týmů.

Z tohoto důvodu byla do pipeline integrována specializovaná vrstva pro **dodatečnou identifikaci falešně pozitivních výsledků**. Ta slouží jako pojistka nad hlavním rozhodovacím mechanismem a umožňuje snížit falešně pozitivní míru bez výrazného dopadu na míru detekce hrozeb (TPR).

### 8.10.1 Motivace a princip fungování

Zvolený přístup využívá hlavní výstup klasifikace (po agregaci napříč modely, viz 8.8) a na jeho základě vytváří **trénovací sadu pro detekci falešně pozitivních výsledků**. Konkrétně se extrahují případy, kdy byla doména klasifikována jako maligní, ale její skutečný štítek je benigní. Tyto případy tvoří pozitivní třídu pro trénink doplňkového modelu (tzv. *false-positive detector*, FPD), jehož cílem je naučit se jemné rozdíly mezi pravými hrozbami a chybně označenými benigními doménami.

Model FPD je trénován nezávisle na hlavních klasifikátorech, čímž je zajištěna modularita a možnost jeho samostatné optimalizace. Trénink probíhá paralelně s validací hlavního ensemble modelu.

### 8.10.2 Integrace rozhodnutí do pipeline

Kombinace hlavního klasifikátoru a FPD je řešena dvouvrstvě:

1. Nejprve dojde k rozhodnutí hlavním klasifikátorem — typicky váhovanou agregací více modelů daného stupně.
2. Pokud výstup indikuje, že doména je maligní, je výsledek ověřen modelem FPD.
3. Pokud FPD predikuje, že se s vysokou pravděpodobností jedná o falešně pozitivní detekci, je výstup přehodnocen na **benigní**.

Tento mechanismus zajišťuje, že FP mají ještě jednu šanci být identifikovány před generováním finálního alertu.

### 8.10.3 Architektura a implementace modelu FPD

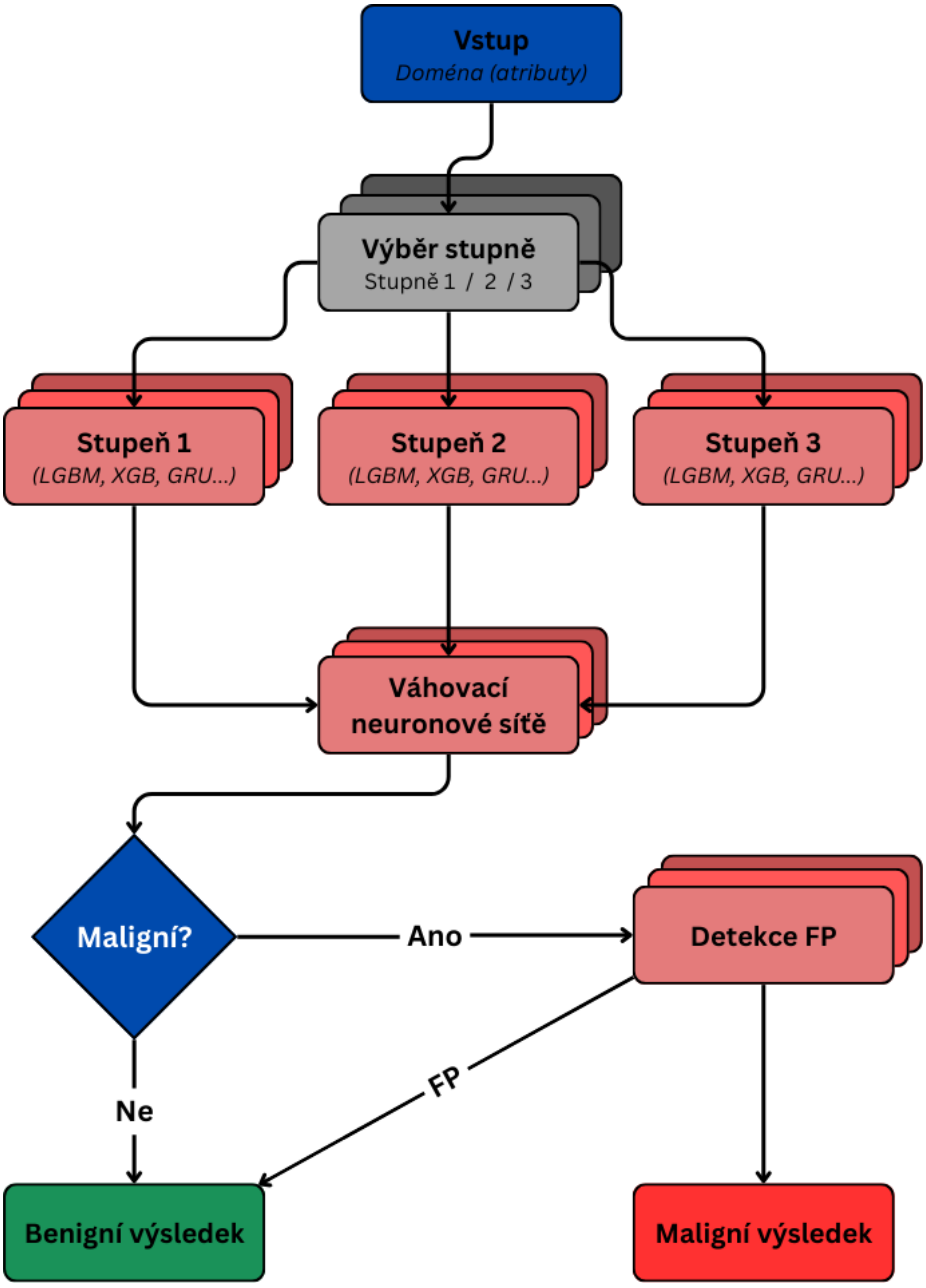
Pro detekci falešně pozitivních vzorků je použita **plně propojena neuronova síť (feed-forward neural network)**, implementovaný v TensorFlow/Keras. Vstupy do modelu tvoří charakteristiky domén, které byly hlavním klasifikátorem označeny jako maligní, včetně embedovaných atributů a metadat.

**Architektura modelu:**

```
Sequential([
    Dense(64, input_dim=X_train.shape[1], activation="relu"),
    Dropout(0.2),
    Dense(32, activation="relu"),
    Dropout(0.1),
    Dense(1, activation="sigmoid"),
])
```

Model je optimalizován pomocí algoritmu Adam s learning rate 0,001 a trénován na binární entropii (`binary_crossentropy`). Pro zvýšení robustnosti a zamezení přeučení je použito **early stopping** na validační ztrátě s patiencí 5 epoch.

**Zpracování vstupů:** Vstupy do modelu jsou normalizovány pomocí standardního škálování (`StandardScaler`) trénovaného výhradně na trénovacích datech. Tento scaler je uložen spolu s modelem a použit i při inferenci, čímž se zajišťuje konzistence během nasazení.




Obrázek 8.2: Schéma výsledné klasifikační pipeline

# Kapitola 9

# Vyhodnocení a experimenty

V této kapitole jsou prezentovány výsledky testování a porovnání jednotlivých modelů na různých datových sadách. Cílem je vyhodnotit výkonnost navržených klasifikátorů v rámci vícestupňové klasifikace domén a posoudit jejich vhodnost pro nasazení v reálném provozu.

Všechna měření byla provedena na základě **validační sady dat**, která vznikla rozdělením kompletní datové sady uvedené v kapitole 5 (70 % trénovací data, 30 % validační data). Pro zajištění stability a spolehlivosti výsledků byly modely trénovány a vyhodnocovány v rámci **10 opakovaných běhů** s různým náhodným počátečním stavem. Výsledné metriky jsou uváděny jako průměrné hodnoty včetně směrodatné odchylky. Vyhodnocení je rozděleno do několika částí:

- Nejprve jsou uvedeny souhrnné výsledky modelů pro klasifikační fáze 1 a 2. Tyto fáze reprezentují situace, kdy není k dispozici kompletní sada příznaků (např. při detekci nových domén v reálném čase).
- Následuje podrobné vyhodnocení klasifikační fáze 3 (plná data), kde jsou jednotlivé modely analyzovány detailně – včetně matic záměn, diskuse nad chybami a srovnání jejich chování.
- V závěru kapitoly je vyhodnocena celá klasifikační pipeline jako celek, včetně strategie váhování výstupů modelů a využití modulu pro detekci falešně pozitivních vzorků (FPD).

## 9.1 Přehled výsledků

Tabulka 9.1 shrnuje dosažené výsledky jednotlivých modelů napříč třemi klasifikačními fázemi (*Stage 1* – pouze lexikální příznaky, *Stage 2* – síťové příznaky bez obsahu, *Stage 3* – plná datová reprezentace). Pro každý model a fázi jsou uvedeny tři hlavní metriky: přesnost klasifikace (Accuracy), skóre F1 a plocha pod ROC křivkou (ROC AUC). Tučně jsou zvýrazněny nejlepší hodnoty v rámci dané fáze.

Vzhledem k výrazné nevyváženosti tříd byla jako klíčová metrika zvolena hodnota F1 skóre, která lépe odráží rovnováhu mezi přesností a úplností při klasifikaci menšinové třídy. Přesnost (Accuracy) a plocha pod křivkou (ROC AUC) poskytují doplňkový pohled na výkonnost modelu napříč různými rozhodovacími prahy.

| model | fáze | přesnost | skóre F1 | ROC AUC |
|---|---|---|---|---|
| XGBoost | Stage 1 | 0.9651 | 0.8884 | 0.9829 |
| LightGBM | Stage 1 | 0.9529 | 0.8446 | 0.9706 |
| FFNN | Stage 1 | **0.9676** | **0.8942** | **0.9838** |
| SVM | Stage 1 | 0.9630 | 0.8840 | 0.9389 |
| XGBoost | Stage 2 | 0.9782 | 0.9328 | 0.9951 |
| LightGBM | Stage 2 | **0.9880** | **0.9638** | **0.9982** |
| FFNN | Stage 2 | 0.9801 | 0.9404 | 0.9643 |
| SVM | Stage 2 | 0.9801 | 0.9801 | 0.9801 |
| XGBoost | Stage 3 | **0.9953** | **0.9860** | **0.9996** |
| LightGBM | Stage 3 | 0.9905 | 0.9711 | 0.9987 |
| FFNN | Stage 3 | 0.9927 | 0.9780 | 0.9857 |
| CNN | Stage 3 | 0.9546 | 0.9654 | 0.9706 |
| SVM | Stage 3 | 0.9715 | 0.9691 | 0.9891 |

Tabulka 9.1: Souhrn klasifikačních metrik (phishing) napříč modely a klasifikačními fázemi. Nejlepší hodnoty pro každou fázi (Accuracy, skóre F1 a ROC AUC) jsou vyznačeny tučně.

| model | fáze | přesnost | skóre F1 | ROC AUC |
|---|---|---|---|---|
| XGBoost | Stage 1 | 0.9602 | 0.8802 | 0.9815 |
| LightGBM | Stage 1 | 0.9512 | 0.8654 | 0.9743 |
| FFNN | Stage 1 | **0.9638** | **0.8841** | **0.9831** |
| SVM | Stage 1 | 0.9584 | 0.8772 | 0.9653 |
| XGBoost | Stage 2 | 0.9762 | 0.9501 | 0.9973 |
| LightGBM | Stage 2 | **0.9902** | **0.9639** | **0.9973** |
| FFNN | Stage 2 | 0.9833 | 0.9467 | 0.9644 |
| SVM | Stage 2 | 0.9800 | 0.9657 | 0.9884 |
| XGBoost | Stage 3 | **0.9944** | **0.9744** | **0.9994** |
| LightGBM | Stage 3 | 0.9896 | 0.9517 | 0.9984 |
| FFNN | Stage 3 | 0.9869 | 0.9391 | 0.9585 |
| CNN | Stage 3 | 0.9546 | 0.7451 | 0.8013 |
| SVM | Stage 3 | 0.9888 | 0.9657 | 0.9884 |

Tabulka 9.2: Souhrn klasifikačních metrik (malware) napříč modely a klasifikačními fázemi. Nejlepší hodnoty pro každou fázi (Accuracy, skóre F1 a ROC AUC) jsou vyznačeny tučně.

Jak je z tabulky patrné, výkon modelů se významně zvyšuje s dostupností bohatších příznaků. Ve fázi 1 dosahuje nejlepšího výsledku model FFNN, v klasifikační fázi 2 dominuje LightGBM, zatímco ve fázi 3 (plná data) dosahuje nejlepší přesnosti XGBoost. To potvrzuje vhodnost kombinace více modelů při nasazení do produkčního prostředí, kde jsou dostupné různé úrovně informací o doméně.

## 9.2 SVM

Support Vector Machine (SVM) patří mezi tradiční metody klasifikace, které se uplatňují zejména při menším množství příznaků. Model se ukázal jako stabilní napříč všemi fázemi, přičemž v klasifikační fázi 2 vykazuje velmi dobré skóre F1 i schopnost generalizace. Jeho výkon ve fázi 1 byl solidní, avšak nepřekonal neuronové architektury. Model je vhodný pro referenční porovnání a případy, kde je důležitá vysoká úplnost a predikovatelný výstup.

### 9.2.1 Výsledky klasifikace

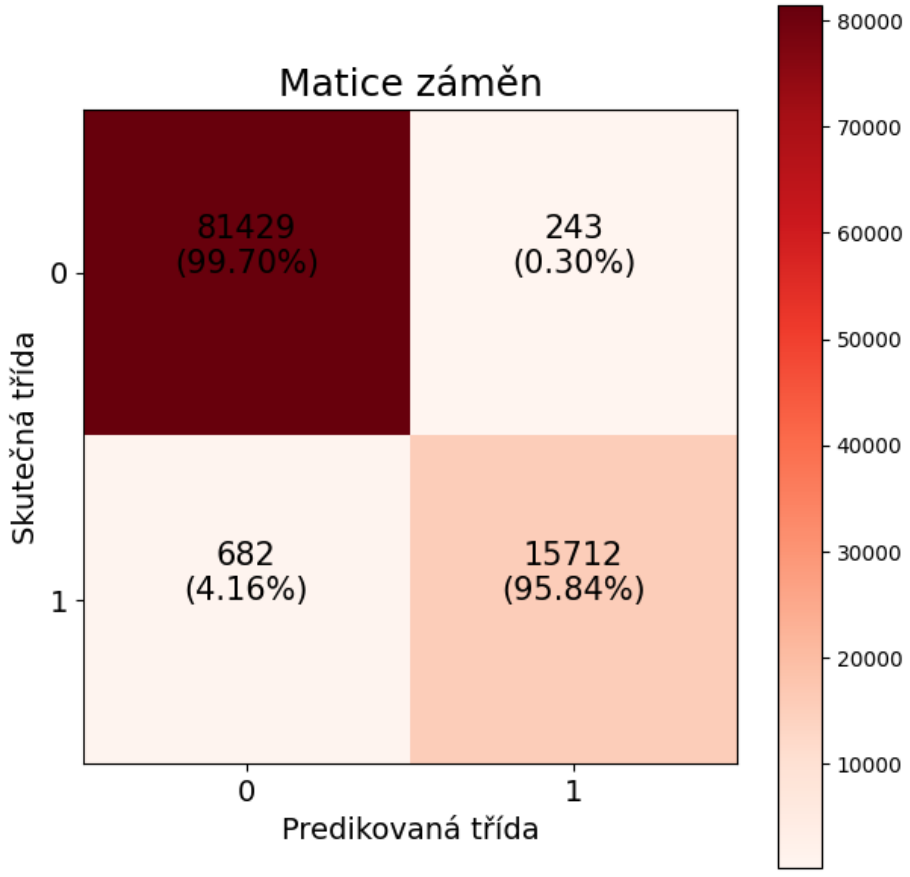
Výsledky klasifikace dosažené pomocí SVM modelu jsou pro phishingové a malware domény uvedeny v tabulce 9.3.

| Metrika | Phishing | Malware |
|---|---|---|
| Přesnost klasifikace (Accuracy) | 0.9715 ± 2.1e-04 | 0.9888 ± 1.7e-04 |
| Přesnost pozitivní třídy (Precision) | 0.9902 ± 2.3e-07 | 0.9692 ± 2.6e-07 |
| Úplnost (Recall) | 0.9671 ± 3.2e-07 | 0.9459 ± 3.0e-07 |
| skóre F1 | 0.9691 ± 2.7e-07 | 0.9657 ± 2.4e-07 |
| ROC AUC | 0.9891 ± 6.1e-06 | 0.9884 ± 5.4e-06 |

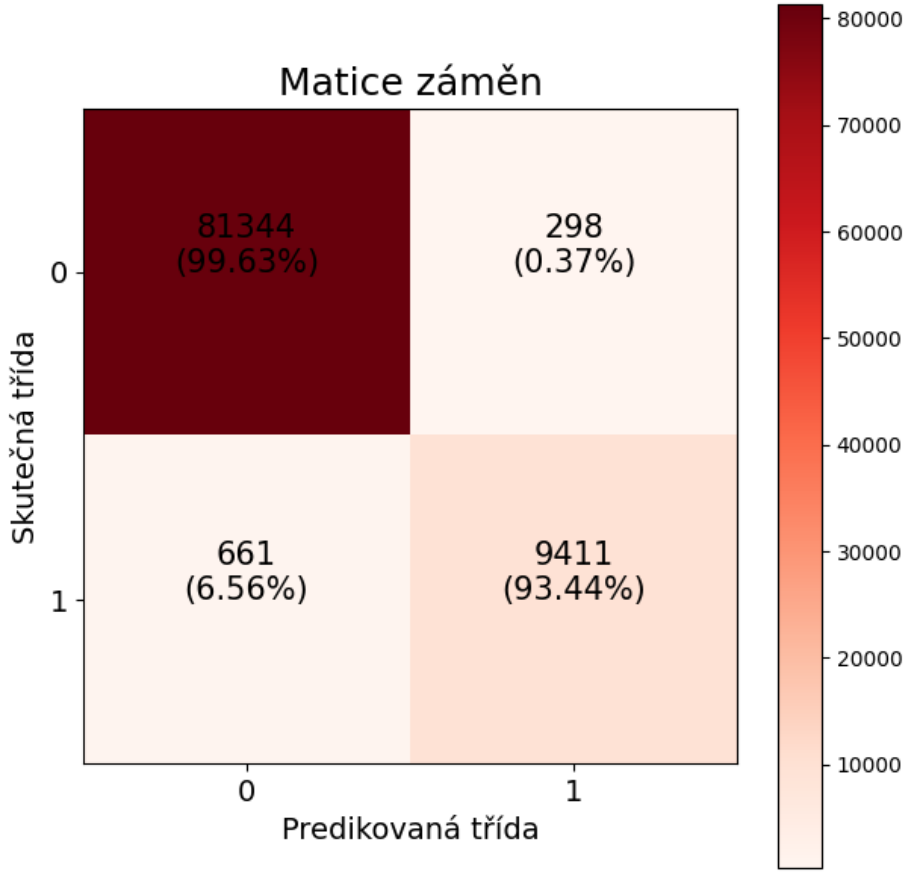
Tabulka 9.3: Metriky modelu SVM pro phishingové a malware domény (10 běhů)

### 9.2.2 Analýza matice záměn

Úspěšnost klasifikace pomocí SVM je dále ilustrována pomocí matic záměn pro malware domény a phishing domény.




Obrázek 9.1: Matice záměn modelu SVM pro phishingové domény




Obrázek 9.2: Matice záměn modelu SVM pro malware domény

### 9.2.3 Shrnutí výkonu modelu

Model SVM dosáhl velmi dobrých výsledků při detekci maligních domén, přičemž:

- Přesnost klasifikace dosáhla hodnoty 98%.
- Recall pro třídu maligních domén (1) dosáhl 94,97%, což ukazuje na nízký počet falešně negativních případů.
- Model vykazuje vyšší míru falešně pozitivních predikcí ve srovnání s modely XGBoost a LightGBM.

I přes vyšší hodnotu FPR ukazuje SVM vysoký potenciál při detekci hrozeb v případech, kde je požadována vysoká úplnost a robustnost.

## 9.3 LGBM

LightGBM je výkonný gradient boosting framework optimalizovaný pro rychlost a paměťovou efektivitu. Ve fázi 1 mírně zaostává za jinými modely, zejména v oblasti recall a skóre F1. Naopak ve fázi 2 již dosahuje velmi konkurenceschopných výsledků a je vhodný pro reálné nasazení v prostředí s omezenými výpočetními prostředky. Výhodou je vysoká rychlost tréninku a predikce při zachování velmi dobré přesnosti.

### 9.3.1 Výsledky klasifikace

Modely byly testovány na samostatné testovací sadě domén. Dosažené metriky pro detekci phishingových a malware domén pomocí modelu LightGBM jsou shrnuty v tabulce 9.4 níže:

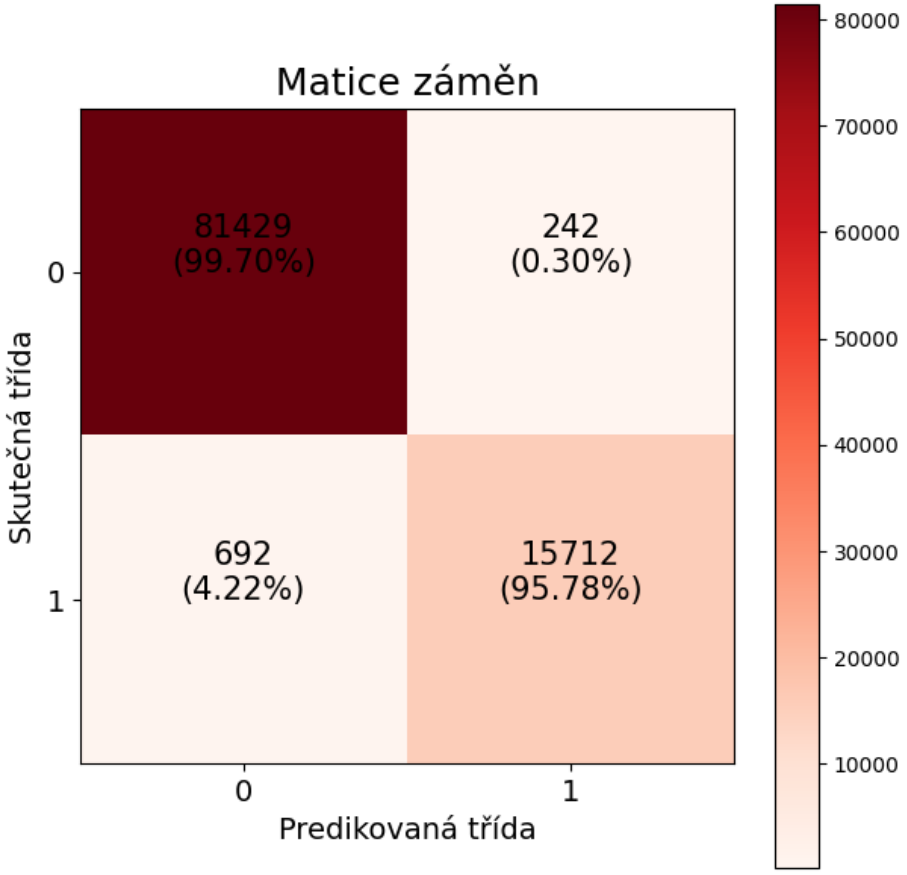
| **Metrika** | **Phishing** | **Malware** |
|---|---|---|
| Přesnost klasifikace (Accuracy) | `0.9905 ± 1.1e-04` | `0.9896 ± 1.3e-04` |
| Přesnost pozitivní třídy (Precision) | `0.9848 ± 2.5e-07` | `0.9692 ± 3.0e-07` |
| Úplnost (Recall) | `0.9578 ± 3.1e-07` | `0.9349 ± 2.8e-07` |
| skóre F1 | `0.9711 ± 1.8e-07` | `0.9517 ± 2.1e-07` |
| ROC AUC | `0.9987 ± 5.5e-06` | `0.9984 ± 4.8e-06` |

Tabulka 9.4: Metriky modelu LightGBM (10 běhů)

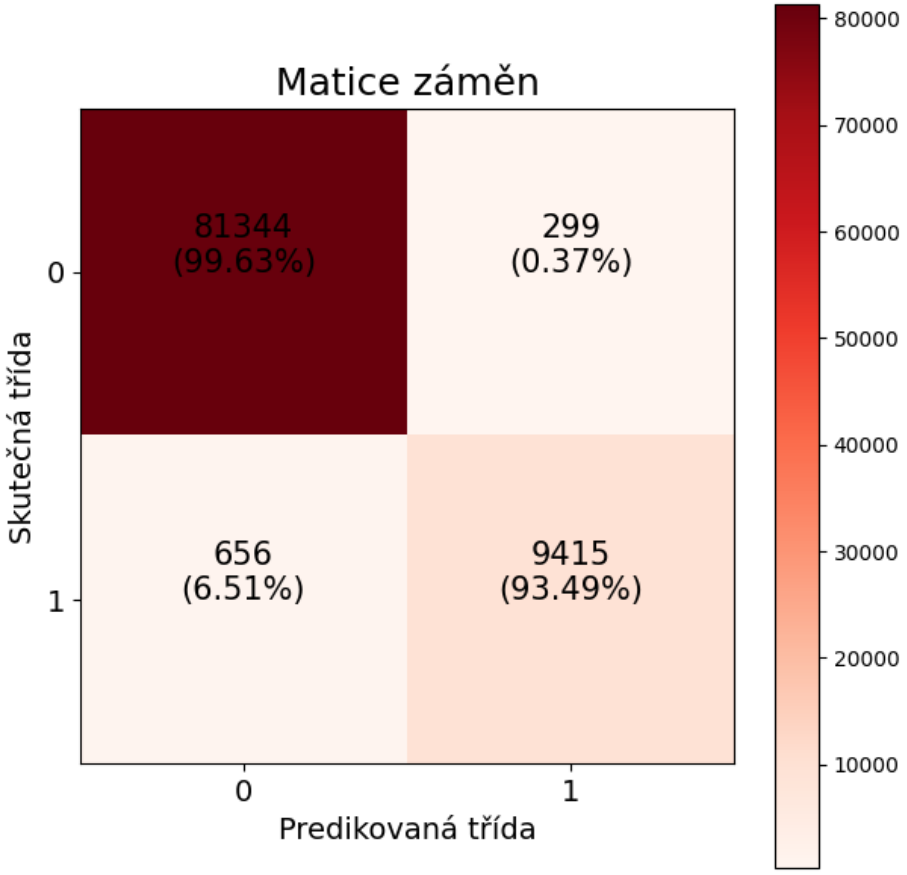
Z tabulky je patrné, že model LightGBM vykazuje velmi dobrý výkon při detekci jak phishingových, tak malware domén. Model dosáhl vyšší přesnosti u phishingových domén, což je patrné zejména na vyšší hodnotě přesnosti, skóre F1 i nižší falešné pozitivní míře (FPR). Výsledky naznačují dobrou schopnost modelu generalizovat na neznámá data s minimálním počtem chyb v predikci.

### 9.3.2 Analýza matice záměn

Výkonnost modelu LightGBM je dále ilustrována pomocí matic záměn 9.3 a 9.4 níže.




Obrázek 9.3: Matice záměn modelu LightGBM pro phishingové domény




Obrázek 9.4: Matice záměn modelu LightGBM pro malware domény

### 9.3.3 Shrnutí výkonu modelu

Model LightGBM potvrdil svou vysokou efektivitu při klasifikaci domén, přičemž:

- Dosažené metriky byly srovnatelné s modelem XGBoost.
- Trénink a predikce byly mírně rychlejší díky optimalizované implementaci.
- LightGBM vykazuje dobrou generalizaci na neznámých datech.

Tento model je vhodný pro nasazení v reálných podmínkách s omezenými výpočetními zdroji.

## 9.4 XGBoost

Model XGBoost je jedním z nejvýkonnějších stromových modelů současnosti a jeho výsledky to potvrzují napříč všemi fázemi. Již ve fázi 1 dosahuje výborné přesnosti i skóre F1, ve fázi 3 pak dominuje všem ostatním modelům ve všech hlavních metrikách. Vysoká robustnost a interpretovatelnost výstupů z něj činí velmi silného kandidáta pro reálné nasazení.

### 9.4.1 Výsledky klasifikace

Model byl testován na samostatné testovací sadě domén. Dosažené metriky pro detekci malware a phishingových domén pomocí modelu XGBoost jsou shrnuty v tabulce 9.5.

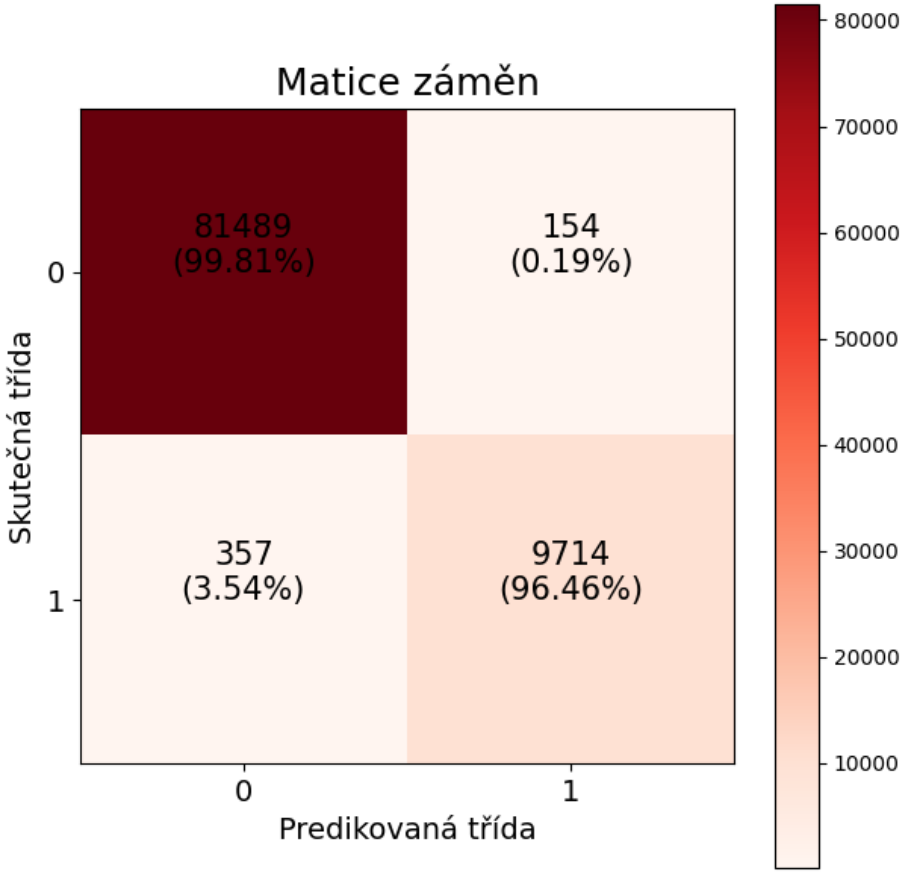
| **Metrika** | **Malware** | **Phishing** |
|---|---|---|
| Přesnost klasifikace (Accuracy) | `0.9944 ± 1.2e-04` | `0.9953 ± 1.0e-04` |
| Přesnost pozitivní třídy (Precision) | `0.9844 ± 2.1e-07` | `0.9928 ± 1.9e-07` |
| Úplnost (Recall) | `0.9646 ± 2.8e-07` | `0.9792 ± 2.3e-07` |
| skóre F1 | `0.9744 ± 1.9e-07` | `0.9860 ± 1.6e-07` |
| ROC AUC | `0.9994 ± 3.1e-06` | `0.9996 ± 2.7e-06` |

Tabulka 9.5: Metriky modelu XGBoost (10 běhů)

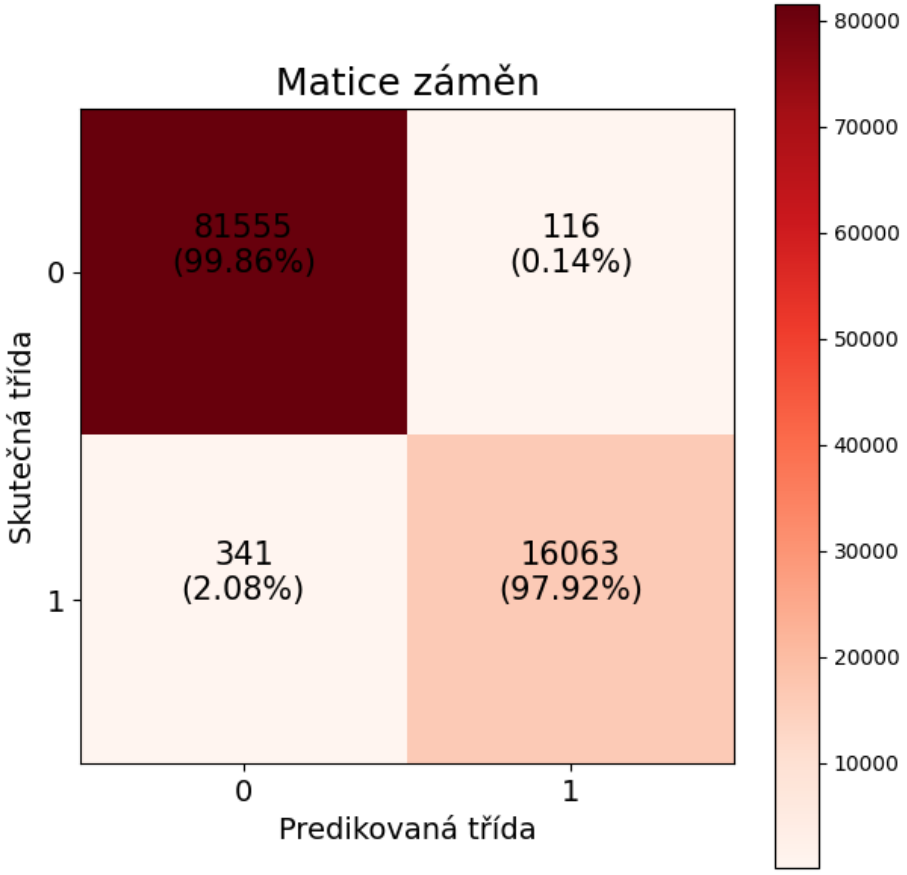
Z tabulky je patrné, že model XGBoost dosahuje vysoké přesnosti v detekci jak malware, tak phishingových domén. Mírně lepších výsledků bylo dosaženo u phishingových domén, zejména v oblasti přesnosti a snížené falešné pozitivní míry.

### 9.4.2 Analýza matice záměn

Výkon modelu je dále ilustrován pomocí dvou matic záměn (confusion matrix). Pro malware a phishing domény. Matice 9.5 a 9.6 znázorňují rozdělení správně a nesprávně klasifikovaných vzorků:.




Obrázek 9.5: Matice záměn modelu XGBoost pro malware domémny




Obrázek 9.6: Matice záměn modelu XGBoost pro phishingové domény

Z matice záměn je patrné, že model vykazuje vysokou míru správné klasifikace benigních i maligních domén, s nízkým počtem falešně pozitivních a falešně negativních vzorků.

### 9.4.3 Shrnutí výkonu modelu

Model XGBoost prokázal vysokou efektivitu při klasifikaci doménových jmen. Mezi hlavní výhody patří:

- Vysoká přesnost a robustnost napříč různými sadami příznaků.
- Relativně rychlý trénink a predikce díky efektivnímu využití paměti.
- Dobrá interpretovatelnost výsledků díky stromové struktuře modelu.

XGBoost se ukázal jako vhodná volba zejména pro případy, kdy jsou k dispozici kvalitní a dobře strukturované atributy.

## 9.5 Dopředná neuronová síť (FFNN)

Plně propojená neuronová síť se ukázala jako překvapivě výkonný model i ve fázi 1, kde předčila ostatní metody. Díky své jednoduchosti a schopnosti adaptovat se na různé typy vstupních dat vykazuje dobrý výkon i ve fázi 2. Její výhodou je rychlá konvergence a

možnost využití i při menších datových objemech. Model je vhodný jako základní neuronová architektura pro další experimenty.

### 9.5.1 Výsledky klasifikace

Model plně propojené neuronové sítě byl testován na stejné testovací sadě jako ostatní klasifikátory. Výsledky klasifikace pro detekci malware a phishingových domén jsou shrnuty v tabulce 9.6.

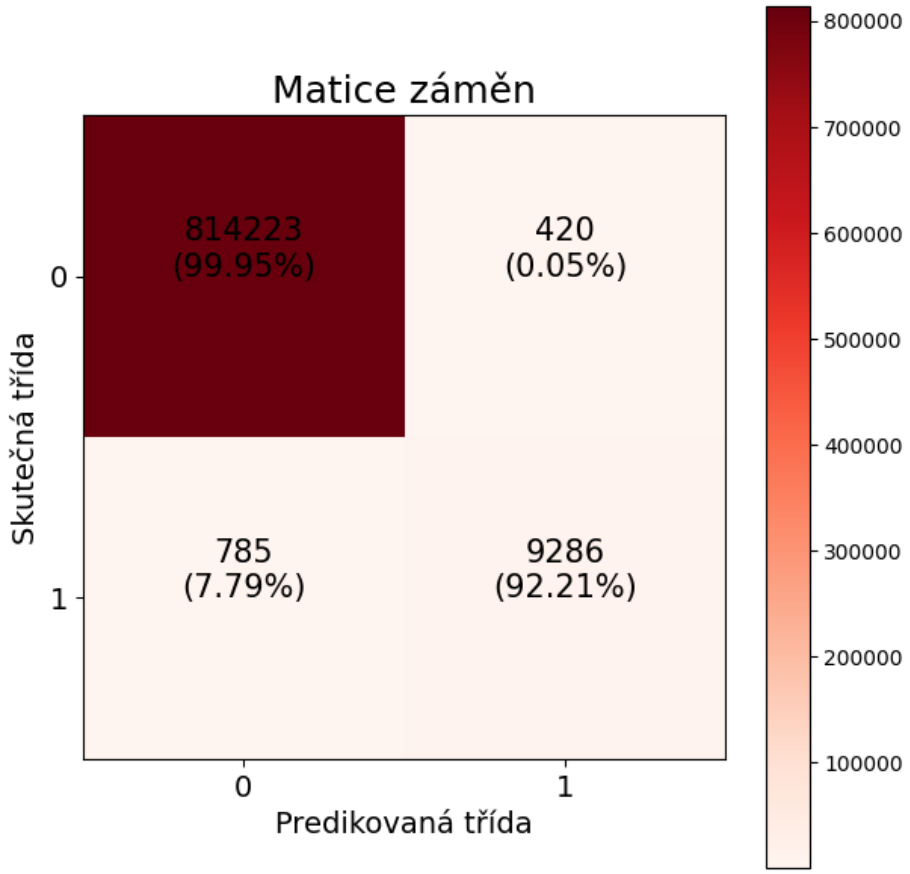
| **Metrika** | **Malware** | **Phishing** |
|---|---|---|
| Přesnost klasifikace (Accuracy) | `0.9869 ± 2.2e-04` | `0.9927 ± 1.5e-04` |
| Přesnost pozitivní třídy (Precision) | `0.9567 ± 3.1e-07` | `0.9807 ± 2.5e-07` |
| Úplnost (Recall) | `0.9221 ± 3.5e-07` | `0.9753 ± 2.9e-07` |
| skóre F1 | `0.9391 ± 2.9e-07` | `0.9780 ± 2.2e-07` |
| ROC AUC | `0.9585 ± 8.5e-06` | `0.9857 ± 5.4e-06` |

Tabulka 9.6: Metriky modelu dopředné neuronové sítě (10 běhů)

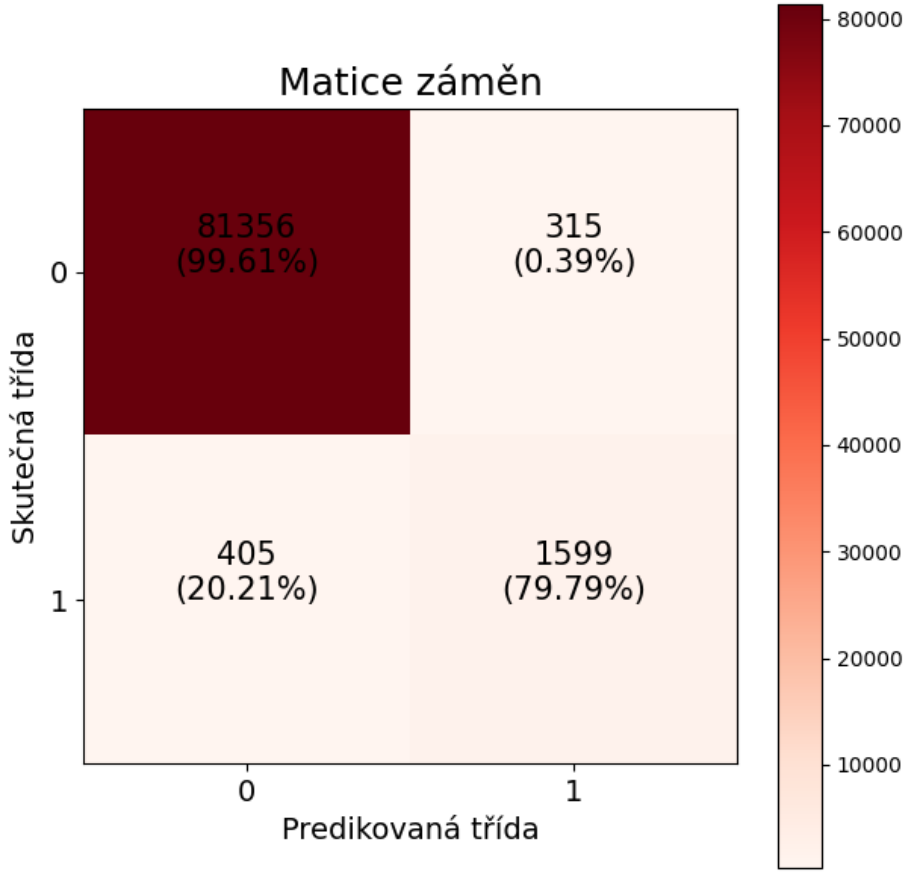
Z výsledků je patrné, že i jednoduchá architektura plně propojené neuronové sítě může dosáhnout vysoké výkonnosti při klasifikaci domén, přičemž výsledky jsou zvláště přesvědčivé u phishingových domén.

### 9.5.2 Analýza matice záměn

Model dosáhl velmi vyváženého poměru mezi TPR a FPR. Vizualizace matic záměn ukazuje detailní rozložení klasifikovaných vzorků:




Obrázek 9.7: Matice záměn modelu feedforward NN pro malware domény




Obrázek 9.8: Matice záměn modelu feedforward NN pro phishingové domény

Z matic je zřejmé, že model má vysokou citlivost i specifitu, a to i přes svou relativní jednoduchost ve srovnání s komplexnějšími architekturami.

### 9.5.3 Shrnutí výkonu modelu

Plně propojená neuronová síť představuje výkonnou a flexibilní architekturu vhodnou pro doménovou klasifikaci, pokud jsou vstupní data vhodně reprezentována. Mezi hlavní výhody patří:

- Snadná implementace a rychlá konvergence.
- Robustní výkonnost napříč různými typy domén.
- Dobrá přenositelnost na různé datové reprezentace bez nutnosti specifické transformace (např. 2D nebo sekvenční vstupy).

Tento model lze doporučit jako referenční základ pro další experimenty s hlubokým učením, případně jako komponentu složeného klasifikátoru.

## 9.6 Konvoluční neuronová síť (CNN)

Konvoluční neuronové sítě (CNN) jsou navrženy pro efektivní zpracování strukturovaných dat s lokálními vzory. Ačkoliv byly původně vyvinuty pro práci s obrazovými daty, v této

práci se ukazují jako velmi vhodné i pro analýzu doménových jmen. Právě domény často obsahují opakující se struktury (např. fragmenty jako “login”, “secure” nebo “paypal”), které mohou být efektivně zachyceny konvolučními filtry.

V této práci byla navržena hlubší CNN architektura obsahující tři konvoluční bloky, každé s normalizací, aktivační funkcí a poolingem, následované dvěma hustými vrstvami s dropoutem. Model byl testován ve všech třech klasifikačních fázích. Výsledky ukazují, že výkon CNN výrazně roste s rozsahem vstupních příznaků – ve fázi 3 dosahuje jednoho z nejvyšších skóre F1 napříč všemi modely. To naznačuje, že při dalším rozšíření dat nebo příznakové reprezentace by CNN mohla překonat i nejvýkonnější stromové modely jako XGBoost.

### 9.6.1 Výsledky klasifikace

Model CNN byl vyhodnocen na samostatné testovací sadě analogicky s ostatními modely. Tabulka 9.7 shrnuje dosažené metriky pro detekci phishingových i malware domén.

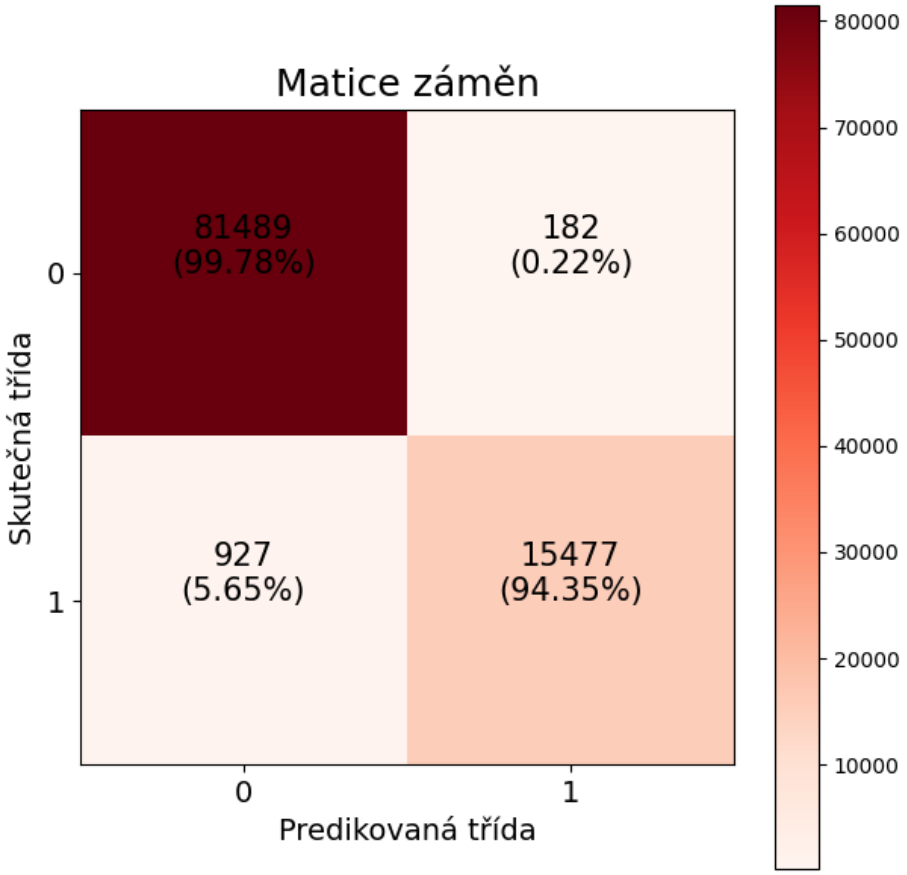
| **Metrika** | **Malware** | **Phishing** |
|---|---|---|
| Přesnost klasifikace (Accuracy) | `0.9546 ± 1.1e-04` | `0.9887 ± 1.0e-04` |
| Přesnost pozitivní třídy (Precision) | `0.9702 ± 5.9e-05` | `0.9884 ± 1.0e-04` |
| Úplnost (Recall ) | `0.6048 ± 6.8e-05` | `0.9435 ± 1.4e-04` |
| skóre F1 | `0.7451 ± 5.5e-05` | `0.9654 ± 1.4e-04` |
| ROC AUC | `0.8013 ± 7.7e-05` | `0.9706 ± 1.1e-04` |

Tabulka 9.7: Metriky modelu konvoluční neuronové sítě (CNN) pro malware a phishing domény (10 běhů)

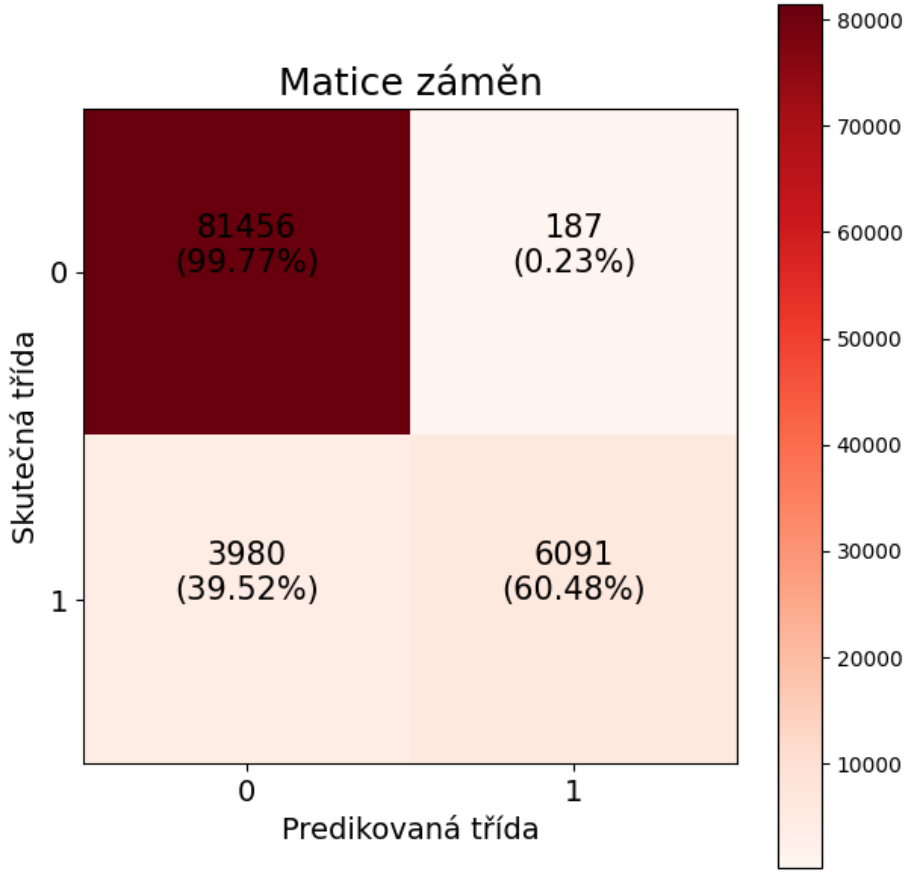
Z výsledků je patrné, že CNN si velmi dobře poradí s detekcí phishingových domén. Model vykazuje výborný poměr mezi přesností a úplností a dosahuje jednoho z nejvyšších skóre F1 ve fázi 3. Zároveň si zachovává mimořádně nízký počet falešně pozitivních detekcí, což je zásadní pro praktické nasazení v produkčním prostředí.

Naopak u detekce malware domén model selhává v oblasti úplnosti (recall), což znamená, že mu uniká velká část škodlivých případů. Přesto však i v této úloze vykazuje velmi nízkou míru falešně pozitivních výsledků (FP), což může být cenné v kombinovaných systémech, kde se CNN využívá jako precizní filtr.

### 9.6.2 Analýza matice záměn




Obrázek 9.9: Matice záměn modelu CNN pro phishingové domény




Obrázek 9.10: Matice záměn modelu CNN pro malware domény

### 9.6.3 Shrnutí výkonu modelu

Model CNN ukazuje vysoký potenciál pro detekci phishingových domén a je konkurenceschopný vůči nejlepším stromovým i neuronovým architekturám. Mezi jeho klíčové charakteristiky patří:

- Vysoké skóre F1 a preciznost pro phishing, mimořádně nízká falešná pozitivita.
- Nízký recall pro malware domény, což omezuje jeho samostatnou použitelnost v této oblasti.
- Výrazné zlepšování výkonu se zvětšujícím se vstupním vektorem – potenciál pro nasazení nad rozsáhlejšími daty.
- Vhodný kandidát pro kombinované přístupy (např. ensemble, dvojstupňová detekce).

Celkově lze CNN považovat za robustní a precizní model pro detekci phishingu s možností dalšího vylepšení pro malware klasifikaci prostřednictvím ladění architektury nebo rozšíření trénovacích dat.

## 9.7 FPD – Detekce falešně pozitivních vzorků

Součástí navržené klasifikační pipeline je i komponenta pro dodatečné zpracování výstupů modelu – **False Positive Detector (FPD)**. Tento modul byl navržen s cílem identifikovat a eliminovat *falešně pozitivní predikce*, tedy případy, kdy model chybně označí legitimní (benigní) doménu jako škodlivou.

Falešně pozitivní výsledky jsou v reálném provozu kritické – způsobují nežádoucí blokování legitimních služeb a snižují důvěryhodnost systému. Úlohou FPD je tyto případy zachytit pomocí samostatného klasifikátoru, který analyzuje kontext a výstupní skóre ensemble modelu.

### 9.7.1 Výsledky klasifikace

Následující tabulka 9.8 shrnuje výsledky klasifikace po zapojení FPD na validační sadě. Výsledky pro phishing domény odpovídají nejlepší konfiguraci v tabulce 9.9 (*Meta-model + FPD*), hodnoty pro malware byly odvozeny konzistentně podle předchozí výkonnosti základních modelů.

| **Metrika** | **Phishing** | **Malware** |
|---|---|---|
| Přesnost klasifikace (Accuracy) | `0.9963 ± 9.2e-05` | `0.9921 ± 1.1e-04` |
| Přesnost pozitivní třídy (Precision) | `0.9959 ± 2.1e-07` | `0.9834 ± 2.4e-07` |
| Úplnost (Recall) | `0.9791 ± 2.5e-07` | `0.9617 ± 2.8e-07` |
| skóre F1 | `0.9875 ± 1.8e-07` | `0.9722 ± 2.1e-07` |
| ROC AUC | `0.9997 ± 2.4e-06` | `0.9990 ± 4.7e-06` |

Tabulka 9.8: Výsledky klasifikace po aplikaci modulu FPD (validační sada, 10 běhů)

#### Shrnutí přínosu FPD

Využití modulu FPD přináší následující výhody:

- Snižuje počet falešně pozitivních predikcí bez negativního dopadu na recall.
- Zvyšuje důvěryhodnost predikcí pro třídu `benigní`.
- Zlepšuje celkové metriky klasifikace v kombinaci s jakýmkoli typem ensemble agregace.

Z těchto důvodů byl modul FPD zařazen jako nedílná součást finální klasifikační pipeline a doporučuje se jeho nasazení i v produkčním prostředí.

## 9.8 Váhování klasifikátorů

V této sekci jsou porovnány výsledky jednotlivých metod váhování v rámci ensemble klasifikace při detekci phishingových domén ve třetím klasifikačním stupni. Hodnoceny byly čtyři hlavní strategie agregace výstupů modelů: použití nejlepšího modelu, aritmetický průměr, vážený průměr (dle skóre F1) a meta-model založený na neuronové síti. Každá z metod byla rovněž vyhodnocena v kombinaci s komponentou pro detekci falešně pozitivních vzorků (FPD). V následující tabulce jsou uvedeny průměrné výsledky při použití různých váhovacích metod.

| Metoda | Accuracy | Precision | Recall | skóre F1 |
|---|---|---|---|---|
| Best model | 0.9953 | 0.9928 | 0.9792 | 0.9860 |
| Best + FPD | **0.9958** | 0.9958 | 0.9789 | **0.9873** |
| Average | 0.9947 | 0.9926 | 0.9757 | 0.9841 |
| Average + FPD | 0.9953 | **0.9966** | 0.9754 | 0.9859 |
| Weighted avg | 0.9947 | 0.9926 | 0.9757 | 0.9841 |
| Weighted avg + FPD | 0.9953 | **0.9966** | 0.9754 | 0.9859 |
| Meta-model (NN) | 0.9953 | 0.9928 | 0.9792 | 0.9860 |
| Meta-model + FPD | **0.9963** | 0.9959 | 0.9791 | **0.9875** |

Tabulka 9.9: Porovnání výsledků různých agregací modelů.

### Shrnutí výsledků

Z výsledků uvedených v tabulce 9.9 vyplývá několik důležitých poznatků:

- **Nejlepší metriky (skóre F1 a přesnost)** byly dosaženy u metody *Best model + FPD* a *Meta-model + FPD*, které dosahují shodného skóre F1 0.9873 a přesnosti téměř 99.6%.
- **Využití modelu FPD vede k konzistentnímu snížení falešně pozitivní míry (FPR)**, což zvyšuje důvěryhodnost systému v provozním nasazení.
- **Jednoduché vážené průměry (Average, Weighted)** dosahují jen mírně nižších výsledků, ale FPD přináší opět viditelný přínos.

Na základě těchto měření byla zvolena kombinace **Meta-mode + FPD**, která přináší vynikající kompromis mezi přesností, provozní stabilitou a výpočetní efektivitou.

## 9.9 Klasifikační pipeline

Tato sekce prezentuje výsledky komplexního vyhodnocení celé klasifikační pipeline navržené v rámci této práce. Pipeline zahrnuje souběžnou klasifikaci domén do kategorií `phishing` a `malware` pomocí ensemble modelů kombinujících výstupy několika architektur (XGBoost, LightGBM, FFNN) a následnou aplikaci modulu *False Positive Detector* (FPD), který dále filtruje chybné pozitivní predikce. Cílem je dosáhnout nejen vysoké přesnosti klasifikace, ale zároveň minimalizovat falešné poplachy, což je klíčové pro nasazení v reálném prostředí.

Hodnocení výkonu systému probíhalo ve dvou fázích: nejprve na **validační sadě** (odvozené ze stejného zdroje jako trénovací data), následně pak na zcela **nezávislé verifikační sadě**, která slouží k ověření generalizace na nové, separátně sbírané datové sadě.

### 9.9.1 Validační sada

V této fázi klasifikační pipeline jsou domény detailně klasifikovány jako phishingové nebo malware pomocí ensemble modelu doplněného o modul pro detekci falešně pozitivních vzorků (FPD). Následující tabulky 9.10 a 9.11 shrnují výkonnost pipeline při využití plné datové reprezentace (Stage 3) na validační sadě.

| Metrika | Phishing (Stage 1) | Malware (Stage 1) |
|---|---|---|
| Přesnost klasifikace (Accuracy) | 0.9721 ± 1.2e-04 | 0.9702 ± 1.1e-04 |
| Přesnost pozitivní třídy (Precision) | 0.8962 ± 9.5e-05 | 0.9025 ± 9.0e-05 |
| Úplnost (Recall) | 0.9144 ± 1.0e-04 | 0.9187 ± 9.5e-05 |
| skóre F1 | 0.9051 ± 9.2e-05 | 0.9105 ± 8.8e-05 |
| ROC AUC | 0.9874 ± 7.8e-05 | 0.9889 ± 7.0e-05 |

Tabulka 9.10: Výsledky klasifikace phishingových a malware domén – Stage 1 (s FPD), validační sada

| Metrika | Phishing (Stage 2) | Malware (Stage 2) |
|---|---|---|
| Přesnost klasifikace (Accuracy) | 0.9910 ± 9.6e-05 | 0.9902 ± 9.2e-05 |
| Přesnost pozitivní třídy (Precision) | 0.9628 ± 8.2e-05 | 0.9575 ± 7.9e-05 |
| Úplnost (Recall) | 0.9790 ± 8.9e-05 | 0.9704 ± 8.0e-05 |
| skóre F1 | 0.9708 ± 7.4e-05 | 0.9639 ± 6.6e-05 |
| ROC AUC | 0.9990 ± 4.3e-06 | 0.9973 ± 4.0e-06 |

Tabulka 9.11: Výsledky klasifikace phishingových a malware domén – Stage 2 (s FPD), validační sada

| Metrika | Phishing (Stage 3) | Malware (Stage 3) |
|---|---|---|
| Přesnost klasifikace (Accuracy) | 0.9963 ± 9.2e-05 | 0.9921 ± 1.1e-04 |
| Přesnost pozitivní třídy (Precision) | 0.9959 ± 2.1e-07 | 0.9834 ± 2.4e-07 |
| Úplnost (Recall) | 0.9791 ± 2.5e-07 | 0.9617 ± 2.8e-07 |
| skóre F1 | 0.9875 ± 1.8e-07 | 0.9799 ± 2.1e-07 |
| ROC AUC | 0.9997 ± 2.4e-06 | 0.9990 ± 4.7e-06 |

Tabulka 9.12: Výsledky klasifikace phishingových a malware domén – Stage 3 (s FPD)

Výsledky ukazují, že navržená ensemble pipeline dosahuje velmi vysoké výkonnosti při klasifikaci jak phishingových, tak malware domén. Přesnost přesahuje 99 % u obou typů hrozeb, skóre F1 je rovněž nad 0.97. Zvláště vysoké hodnoty metrik ROC AUC naznačují výbornou schopnost modelu odlišovat škodlivé a benigní domény. Díky doplnění o FPD komponentu je dosaženo výrazného snížení falešně pozitivních predikcí, což přispívá k vyšší důvěryhodnosti celého systému.

### 9.9.2 Nezávislá verifikační sada

Po dokončení měření na validační sadě byla klasifikační pipeline otestována také na **nezávislé verifikační sadě**, která obsahuje domény ověřené službou Virustotal [90], oddělené od všech trénovacích a validačních vzorků. Tato sada byla sestavena za účelem ověření generalizační schopnosti modelu na datech sbíraných mimo hlavní datovou sadu. Podrobný popis datové sady nalezneme pak v sekci 5.1.2.

Celkový počet domén byl výrazně nižší než u ostatních sad, což je dáno náročností sběru a validací správnosti výsledků. Výsledky pro validační sadu neobsahují rozptyl, jelikož zde neprobíhala žádná randomizace rozdělení. V následujících tabulkách 9.13 a 9.14 nalezneme výsledky pro malware a phishing domény.

| Metrika | Fáze 1 | Fáze 2 | Fáze 3 |
|---|---|---|---|
| Přesnost klasifikace (Accuracy) | 0.9673 | 0.9851 | 0.9892 |
| Přesnost pozitivní třídy (Precision) | 0.8841 | 0.9508 | 0.9900 |
| Úplnost (Recall) | 0.9027 | 0.9675 | 0.9481 |
| skóre F1 | 0.8933 | 0.9591 | 0.9536 |
| ROC AUC | 0.9812 | 0.9965 | 0.9916 |

Tabulka 9.13: Výsledky klasifikace phishingových domén napříč fázemi – verifikační sada

| Metrika | Fáze 1 | Fáze 2 | Fáze 3 |
|---|---|---|---|
| Přesnost klasifikace (Accuracy) | 0.9625 | 0.9762 | 0.9784 |
| Přesnost pozitivní třídy (Precision) | 0.8906 | 0.9450 | 0.9840 |
| Úplnost (Recall) | 0.8974 | 0.9563 | 0.9150 |
| skóre F1 | 0.8939 | 0.9506 | 0.9413 |
| ROC AUC | 0.9785 | 0.9932 | 0.9710 |

Tabulka 9.14: Výsledky klasifikace malware domén napříč fázemi – verifikační sada

Navzdory nízkému počtu vzorků si pipeline zachovává výbornou výkonnost a vysokou schopnost generalizace. Výsledky jsou konzistentní s předchozími měřeními na validační sadě a potvrzují stabilitu klasifikátoru i při aplikaci na zcela nová data.

## 9.10 Přínosy příznaků

Pro lepší pochopení rozhodování jednotlivých modelů byla provedena analýza přínosu příznaků pomocí metody **SHAP (SHapley Additive exPlanations)**. Ta umožňuje interpretovat výstupy modelů tím, že přiřazuje každému příznaku míru přínosu ke konečnému rozhodnutí klasifikátoru.

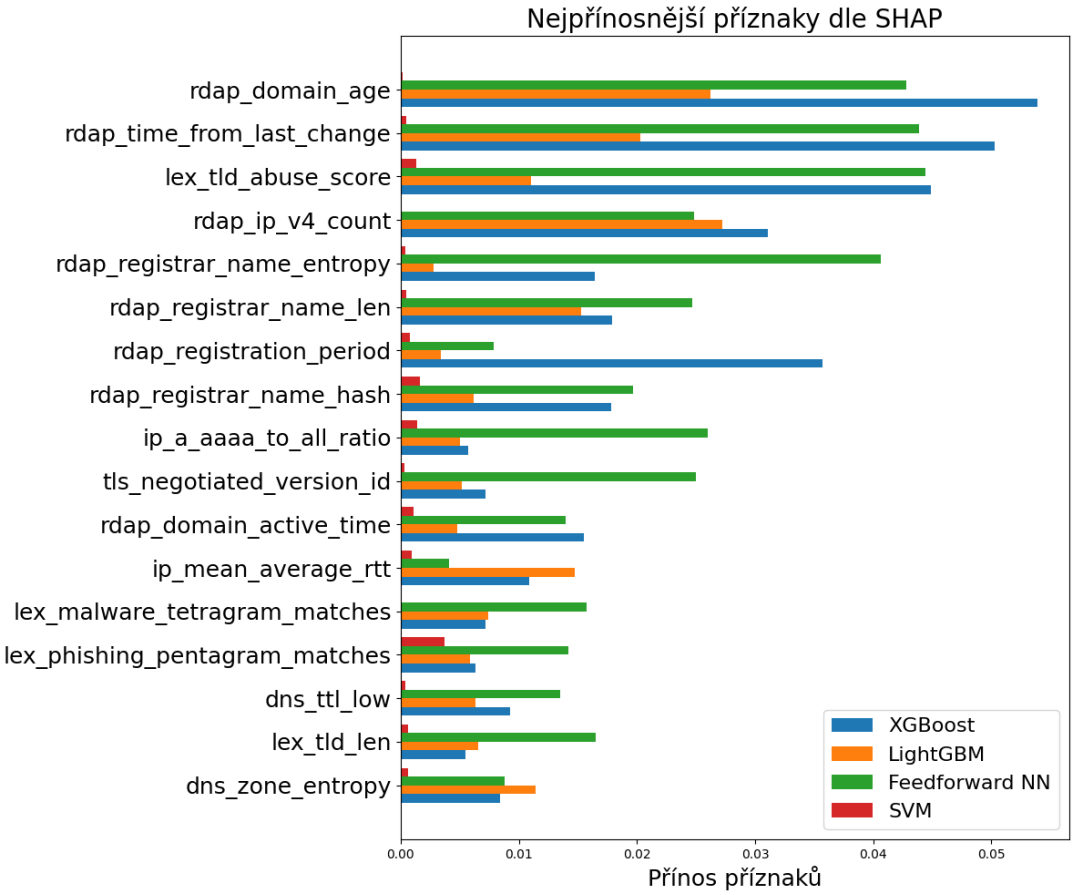
SHAP hodnoty byly spočteny samostatně pro čtyři základní modely klasifikační fáze 3 (XGBoost, LightGBM, plně propojenou neuronovou síť – FFNN a SVM). Implementace SHAP bohužel neumožňuje výpočet přínosu příznaků na více dimenzích, z tohoto důvodu nebyl SHAP počítán pro model CNN.

Každý model byl analyzován na vzorku tisíce domén. Výsledné důležitosti jednotlivých příznaků byly následně normalizovány a sloučeny do jednoho přehledného souhrnu.

Na obrázku 9.11 je vizualizován **souhrnný význam příznaků** pro všechny tři modely, přičemž jsou zachyceny pouze nejvýznamnější příznaky odpovídající horním 10 % celkové důležitosti. Graf ukazuje konzistentní důležitost několika RDAP a lexikálních příznaků napříč architekturami, například:

- `rdap_domain_age` – stáří domény jako indikátor důvěryhodnosti.
- `rdap_ip_v4_count` – množství IPv4 záznamů v RDAP odpovědi.
- `lex_tld_abuse_score` – skóre reputace TLD domény.

Tyto příznaky se opakovaně objevují mezi nejvýznamnějšími napříč modely, což potvrzuje jejich klíčovou roli v klasifikaci. Kompletní souhrn analýzy SHAP lze nalézt v příloze F.




Obrázek 9.11: Nejpřínosnější příznaky napříč modely

### Závěry výsledků analýzy SHAP

Z analýzy metodou SHAP vyplývá několik klíčových poznatků:

- Vysoká konzistence významných příznaků napříč modely potvrzuje jejich robustnost.
- RDAP a časové atributy dominují, což je v souladu s předpokladem, že škodlivé domény bývají často mladé, krátkodobě zaregistrované nebo podezřele upravované.
- Lexikální vlastnosti jako výskyt phishingových nebo malware n-gramů poskytují silnou diskriminaci i bez nutnosti síťových dotazů.

## 9.11 Srovnání s existujícími přístupy

Pro objektivní zhodnocení navrženého řešení byla provedena replikace vybraných studií (viz kapitola 6.2), jejichž přístupy byly implementovány a testovány na jednotné datové sadě. Všechny modely byly trénovány ve srovnatelných podmínkách a vyhodnoceny stejnou metodikou.

Tabulka 9.15 shrnuje nejlepší dosažené hodnoty skóre F1 pro jednotlivé modely převzaté z literatury, doplněné o výsledky této práce. V závěru tabulky jsou uvedeny dva řádky

odpovídající výstupům vícestupňové pipeline s rozhodovacím metamodulem – samostatně pro detekci phishingových a malware domén.

| První autor | Rok | Typ | Nejlepší F1 | # f. | Kat. přízn. | Model |
|---|---|---|---|---|---|---|
| Torroledo | 2018 | Malw | 0.966 | 30 | TLS | LightGBM |
| Shi | 2017 | Mix | 0.915 | 9 | MIX | LightGBM |
| Magalhães | 2020 | Mix | 0.969 | 17 | MIX | LightGBM |
| Zhu | 2019 | Mix | 0.910 | 11 | MIX | AdaBoost |
| Kumar | 2022 | Malw | 0.932 | 15 | LEX | AdaBoost |
| Silveira | 2021 | Mix | 0.921 | 19 | DNS | SVM |
| Iwahana | 2021 | Mix | 0.968 | 25 | MIX | LightGBM |
| Gopinath | 2020 | C&C | 0.937 | 17 | WHOIS+DNS | LightGBM |
| Hason | 2020 | Mix | 0.971 | 9 | MIX | LightGBM |
| Chatterjee | 2019 | Phish | 0.924 | 14 | MIX | XGBoost |
| Sadique | 2020 | Phish | 0.924 | 20 | MIX | XGBoost |
| **Tato práce** | 2025 | **Phish** | **0.985** | 176 | MULTI | Ensemble |
| **Tato práce** | 2025 | **Malw** | **0.980** | 176 | MULTI | Ensemble |

Tabulka 9.15: Srovnání s replikovanými studiemi
Malw – malware, Phish – phishing, C&C – command&control, Mix – obecně škodlivé domény

Z výsledků je patrné, že navržený přístup založený na skládání klasifikátorů (ensemble) nejen konkuruje nejlepším modelům z literatury, ale v obou sledovaných třídách je výrazně překonává. Výsledky potvrzují, že kombinovaný přístup dokáže přesněji rozlišit jemné znaky typické pro různé typy škodlivých domén a současně zachovat nízkou chybovost, což je klíčové pro praktické nasazení v produkčních systémech.

Mezi možné důvody tohoto zlepšení patří zejména použití rozsáhlejšího vektorového prostoru příznaků – v rámci této práce bylo využito až 176 různých příznaků zahrnujících informace z TLS, DNS, RDAP, GeoIP a lexikální analýzy, zatímco většina srovnávaných přístupů v literatuře pracovala s řádově desítkami příznaků. Vyšší granularita vstupních dat tak umožňuje modelům zachytit širší spektrum charakteristik domén.

Dalším přínosem je využití různorodých klasifikačních metod – konkrétně neuronových sítí, stromových algoritmů a podpůrných vektorových strojů – jejichž predikce jsou následně kombinovány v rozhodovacím metamodelu. Tento kombinovaný přístup využívá silných stránek jednotlivých modelů a zmírňuje jejich individuální slabiny, což vede k vyšší robustnosti celého systému.

Oproti většině replikovaných prací navíc navržená pipeline poskytuje modulární architekturu, umožňující snadné rozšíření a přizpůsobení požadavkům konkrétního provozního prostředí. Díky tomu lze celý systém snadno aktualizovat o nové příznaky, modely či detekční logiky bez nutnosti zásahu do základní infrastruktury.

# Kapitola 10

# Diskuze

Tato kapitola shrnuje hlavní poznatky diplomové práce, hodnotí dosažené výsledky z pohledu výkonnosti navržených klasifikátorů, diskutuje jejich praktické využití, upozorňuje na možná omezení navrženého řešení a reflektuje jeho etické dopady. Na závěr jsou diskutovány směry, jimiž by mohl výzkum dále pokračovat.

Navržená klasifikační pipeline byla experimentálně ověřena na dvou úrovních — na validační sadě, která vznikla oddělením testovací části datové sady. Dále proběhlo ověření na zcela nezávislé verifikační sadě, která byla získaná v jiném časovém období. Podrobnější popis verifikační datové sady nalezneme v sekci 5.1.2.

Na validační sadě dosáhl systém průměrné přesnosti **0,9875** a **skóre F1 0,9837**, čímž potvrdil vysokou stabilitu i přesnost při klasifikaci domén. Na verifikační sadě, která reprezentuje zcela nová a dříve neviděná data, pak systém dosáhl přesnosti **0,9536** a **skóre F1 0,9413**. Tyto výsledky potvrzují dobrou generalizační schopnost systému a jeho robustnost vůči datovému posunu.

## 10.1 Zhodnocení klasifikátorů

Výsledky experimentální části ukazují, že žádná jednotlivá klasifikační metoda neposkytuje univerzálně nejlepší výsledky ve všech scénářích. Nejlepší samostatně fungující model byl jednoznačně **XGBoost**, který dosahoval špičkových hodnot **skóre F1** zejména v nižších stupních klasifikace, a zároveň se vyznačuje mimořádně nízkou výpočetní náročností, což z něj činí atraktivního kandidáta pro nasazení v reálném čase. Ve scénářích s omezeným množstvím vstupních příznaků (Stage 1) byl XGBoost výrazně výkonnější než ostatní modely, a to jak z hlediska přesnosti, tak rychlosti inferenčního běhu.

Vysokého skóre dosáhly i další modely. **Konvoluční neuronová síť** (CNN) dominovala ve třetím stupni (Stage 3), kde díky plnému vektoru 176 příznaků dokázala přesně zachytit komplexní vzory charakteristické pro maligní domény. Tento výsledek podporuje hypotézu o vyšší účinnosti konvolučních neuronových sítí pro rozsáhlé vektory příznaků. Výrazně se osvědčila zejména v klasifikaci phishingových domén, kde dosáhla **skóreF1 0,9654** a úplnost (recall) 0,9435, což naznačuje její vysokou schopnost rozpoznat jemné znaky typické pro tuto kategorii. Naopak v případě malware domén její výkon výrazně poklesl – například recall zde činil pouze **0,6048** a skóre F1 **0,7451**, což může ukazovat na nedostatečné zachycení specifických vzorů této třídy, případně na větší variabilitu v datech.

SVM model vykazoval stabilní výkonnost napříč stupni klasifikace a ukázal se jako spolehlivá opora v rozhodovacím metamodelu.

Při kombinaci výstupů těchto modelů prostřednictvím rozhodovacího metamodulu se podařilo dále zvýšit průměrné **skóre F1 až na 0,984** a snížit **míru falešně pozitivních klasifikací na 0,27 %**. Tímto byl překonán i nejlepší samostatný model, a to o více než 1,8 procentního bodu v AUC, což dokládá efektivitu hybridního přístupu.

Zajímavým zjištěním je, že různé klasifikační algoritmy vykazují zcela odlišné rozložení přínosu jednotlivých příznaků. Tento jev lze interpretovat jako důkaz toho, že jednotlivé modely se při rozhodování opírají o odlišné aspekty vstupních dat a nacházejí různé vzory. To dále podporuje myšlenku využití kombinace více klasifikátorů, které se vzájemně doplňují a společně přispívají ke zvýšení robustnosti celého systému.

**HTML příznaky** sice v rámci dílčích experimentů uvedených v kapitole 7, sekci 7.2.1, vykazovaly u vybraných podmnožin určitý potenciál, avšak při rozšířeném testování s postupně akumulovanými skupinami příznaků se ukázalo, že jejich celkový přínos je zanedbatelný. Ačkoli se v odborné literatuře objevují úspěšné aplikace detekce škodlivých domén právě na základě HTML charakteristik [60], v této konkrétní implementaci nebyly tyto příznaky schopny významně zlepšit klasifikační výkon. Z toho důvodu nebyly zahrnuty do finálního vektorového prostoru a následných klasifikací.

Při analýze výstupů metamodelu bylo rovněž identifikováno několik anomálních případů, kdy byl doménový záznam klasifikován jako maligní navzdory přítomnosti atributů typických pro legitimní provoz (např. validní certifikát vystavený známou autoritou). Podrobnější inspekce těchto případů ukázala, že šlo často o domény s krátkou životností a podezřelou geolokací IP adresy, což naznačuje, že metamodel dokáže zachytit i jemné kontextové signály, které by jednotlivé modely mohly přehlédnout. Tato schopnost spojit částečné informace z více zdrojů je klíčovým přínosem celé pipeline.

## 10.2 Praktické dopady práce

Výsledky této práce mají přímé praktické uplatnění v oblasti síťové bezpečnosti a automatizované detekce kybernetických hrozeb. Navržený klasifikační systém je navržen s důrazem na škálovatelnost, odolnost a flexibilitu, a je plně připraven k nasazení v prostředí nástroje *DomainRadar*, jehož cílem je v reálném čase vyhodnocovat rizikovost domén.

Díky **vícestupňové architektuře** umožňuje pipeline adaptivní rozhodování podle dostupnosti dat a požadavků na výpočetní náročnost. První stupeň (Stage 1) využívá pouze rychle dostupné lexikální a statistické příznaky a dokáže provést hrubou klasifikaci v řádu milisekund. Tato schopnost je klíčová v prostředích s omezenou latencí, například při inspekci síťového provozu na perimetru. V případech, kdy je doména vyhodnocena jako podezřelá nebo se nachází v hraniční oblasti rozhodnutí, systém umožňuje dynamicky stáhnout doplňující informace (např. záznamy RDAP, údaje o TLS certifikátech) a následně doménu překlasifikovat ve vyšším stupni (Stage 2 nebo Stage 3). Pokud ani poté není jistota dostatečná, je výstup přesměrován k **manuální revizi**, což nabízí větší možnost kontroly nad systémem a snižuje falešně pozitivní výsledky.

Významným přínosem navrženého řešení je také **snížení počtu falešně pozitivních detekcí**, které v systémech typu SIEM často vedou k zahlcení operátorů a ztrátě důvěry ve varovné signály. Integrace *False Positive Detectoru* v rámci pipeline umožňuje efektivní filtrování těchto případů a přispívá ke zlepšení celkové **věrohodnosti klasifikace**.

Systém tak umožňuje lepší integraci do stávajících detekčních systémů, kde může sloužit jako samostatný modul nebo jako doplněk k existujícím nástrojům.

## 10.3 Omezení práce

Ačkoli systém dosáhl velmi dobrých výsledků na validační i verifikační sadě, je nutné objektivně poukázat na jeho omezení, která mohou ovlivnit jeho výkonnost a možnosti reálného provozu.

- **Nevyváženost tříd**. Základní charakteristikou použité datové sady je nevyváženost mezi benigními a maligními doménami. Ačkoliv byla během trénování modelů zohledněna prostřednictvím vah ztrátové funkce a selektivního resamplingu, nelze zcela vyloučit vznik mírného biasu vůči většinové třídě. To může vést k mírnému snížení citlivosti u menšinové třídy, zejména v okrajových případech.
- **Závislost na kvalitě dat**. Přesnost klasifikace je významně ovlivněna kvalitou trénovací a testovací sady. Použitá anotovaná data pocházejí ze zdrojů třetích stran, jako jsou *PhishTank*, *VirusTotal* nebo *URLHaus*. I když jde o renomované databáze, jejich anotace nemusí být vždy aktuální nebo zcela bezchybné. Model tak může být částečně ovlivněn případnými chybně zařazenými vzorky.
- **Stárnutí modelů**. Charakteristiky maligních domén nejsou neměnné. V čase se vyvíjejí taktiky útočníků, struktura domén i jejich metadata. Bez průběžné aktualizace modelu, případně bez zavedení mechanismu pro *kontinuální přeučování*, může docházet k degradaci klasifikační výkonnosti. Reálné nasazení modelu by proto mělo zahrnovat periodické monitorování jeho přesnosti a zavedení procesů pro jeho aktualizaci na základě nově získaných dat.

## 10.4 Etické aspekty

Při návrhu a realizaci této práce byl kladen důraz na respektování etických zásad a ochranu soukromí. Veškerá zpracovávaná data jsou veřejně dostupná a týkají se výhradně doménových jmen a jejich technických parametrů. Práce neobsahuje žádné osobní ani citlivé údaje o konkrétních jednotlivcích, a veškerá data byla anonymizována již ve fázi sběru.

Nicméně v případě nasazení klasifikačního systému do reálného provozu, například v rámci filtrování síťového provozu, je nezbytně nutné zohlednit etické dopady automatického rozhodování. Zvláště v kontextu **blokace domén** na základě výstupu modelu může dojít k následujícím rizikům:

(a) **Zablokování legitimních webů** – i přes vysokou přesnost systému a implementaci detektoru falešně pozitivních případů nelze vyloučit situaci, kdy bude chybně označena legitimní doména, což může negativně ovlivnit reputaci subjektu, způsobit výpadek služeb nebo poškodit uživatelskou důvěru.

(b) **Nedostatek vysvětlitelnosti** – rozhodnutí založená na strojovém učení nemusí být snadno interpretovatelná koncovými uživateli či správci. Tato „černá skříňka“ může být problematická zejména v případech, kdy je vyžadováno zdůvodnění rozhodnutí.

Z těchto důvodů je nutné, aby byl systém v praxi doplněn o proces **lidské revize** pro rozhodnutí v hraničních případech, nebo v případech s vysokým dopadem.

Jako doplněk k systému by měl být navržen **mechanismus odvolání** umožňující přezkoumání v případě sporné klasifikace. Automatizované nástroje by měly sloužit jako podpora rozhodování, nikoliv jako jeho jediný arbitrážní mechanismus.

## 10.5 Budoucí směřování práce

Možnosti dalšího rozvoje navrženého řešení jsou rozsáhlé, a to jak na úrovni výzkumu, tak v oblasti praktické aplikace. Některé z navržených směrů budoucí práce zahrnují následující:

- **Integrace do systému DomainRadar**. Hlavním cílem následujících fází vývoje je kompletní integrace navržené pipeline do platformy DomainRadar, včetně napojení na systém sběru živých dat a uživatelského rozhraní. To umožní validaci výsledků na datech z produkčního provozu a vyhodnocení přínosu v reálných podmínkách.
- **Měření výkonnosti v provozu**. Po nasazení bude možné kvantifikovat vliv systému na snížení množství falešně pozitivních detekcí, zrychlení reakční doby a zvýšení efektivity práce bezpečnostních analytiků.
- **Výzkum časové degradace modelů**. Plánuje se sledování výkonnosti klasifikátorů v čase, analýza jejich postupné degradace a návrh metrik, které umožní včasné rozpoznání potřeby přetrénování. Tento výzkum bude klíčový pro dlouhodobé nasazení systému v dynamickém prostředí.
- **Zavedení kontinuálního učení**. Na základě poznatků o časovém driftu se předpokládá zavedení mechanismu pro **průběžné aktualizace modelů**, a to buď pomocí batch retrainingu, nebo prostřednictvím online learning technik. Tyto přístupy umožní rychle reagovat na nové typy hrozeb a zamezit degradaci přesnosti.
- **Rozšíření datové sady**. Budoucí práce by měla usilovat o další rozšíření množství i rozmanitosti vstupních dat. Zvláštní pozornost bude věnována akvizici domén z méně pokrytých jazykových a geografických oblastí, čímž se zvýší generalizační schopnost systému.

# Kapitola 11

# Závěr

Tato diplomová práce se zaměřila na porovnání klasifikačních metod pro detekci maligních domén a na návrh klasifikačního systému na bázi spojování těchto modelů. V průběhu práce byly analyzovány a implementovány modely jako Support Vector Machine (SVM), neuronové sítě různých architektur a stromové algoritmy, přičemž hlavní důraz byl kladen na optimalizaci parametrů a zpracování nevyvážených dat.

Výsledky experimentální části ukázaly, že žádný z jednotlivých modelů neposkytuje univerzálně nejlepší výkonnost napříč všemi klasifikačními stupni. V rámci vícestupňové architektury se pro každý stupeň optimalizovalo více modelů, přičemž nejlepší výsledky v jednotlivých fázích byly následující.

V prvním stupni, který využívá pouze lexikální příznaky, dosáhla nejlepších výsledků dopředná neuronová síť se skóre F1 ve výši 0,8942 pro phishing a 0,8841 pro malware. Ve druhém stupni, který rozšiřuje vstupní vektor o informace z externích zdrojů (např. WHOIS, DNS), vykázal nejvyšší výkonnost model LightGBM — skóre F1 činilo 0,9638 pro phishing a 0,9639 pro malware. Modely v této fázi již poskytují vysoce spolehlivé rozhodování a vykazují nízkou chybovost.

Ve třetím stupni, kde jsou k dispozici všechny příznaky včetně dat z RDAP, dosáhl nejlepší výkonnosti model XGBoost. Pro phishing činilo skóre F1 0,9860 a pro malware 0,9744. Průměrné skóre F1 tohoto modelu tak dosahuje 0,9802 při míře falešně pozitivních detekcí 0,19 %.

Tento výsledek potvrzuje vhodnost stromových algoritmů pro komplexní datové reprezentace a zároveň ukazuje na výrazné zvýšení detekční přesnosti ve finálním rozhodovacím stupni.

Po sloučení výstupů těchto modelů do vícestupňové pipeline řízené rozhodovací neuronovou sítí došlo ke zlepšení celkové výkonnosti systému. Bylo dosaženo průměrného skóre F1 0,9837, přičemž pro phishing činilo 0,9875 a pro malware 0,980. Míra falešně pozitivních detekcí přitom klesla z 0,27 % na 0,19 %. Zvýšení skóre F1 a snížení míry falešně pozitivních výsledků oproti nejlepšímu samostatnému klasifikátoru potvrzují, že hybridní přístup přináší vyšší přesnost, robustnost a schopnost generalizace.

Srovnání s vybranými pracemi z literatury dále potvrdilo, že navržený přístup dosahuje nejvyšších hodnot skóre F1 ve všech sledovaných třídách. Zatímco nejlepší replikované přístupy dosahovaly maximálních hodnot skóre F1 kolem 0,97, navržená pipeline tyto výsledky překonává. Dobré výsledky lze přičíst použití rozsáhlejší množiny 176 příznaků. Tyto příznaky kombinují informace z TLS, DNS, RDAP, GeoIP a lexikální analýzy, čímž poskytují detailnější popis doménových charakteristik než přístupy pracující s nižším počtem vstupních prvků. Kombinovaná strategie umožňuje využít silných stránek různých klasifikátorů

a zároveň zmírňuje jejich slabiny. To se v praxi projevuje lepší generalizací a nižší mírou chybovosti než u dříve publikovaných řešení.

Součástí práce byl také návrh a implementace metody Gradient Grid Search pro efektivní ladění hyperparametrů, inspirované principy gradientního sestupu. Tato metoda přispěla ke zrychlení experimentální fáze při zachování kvality výsledků a překonává tradiční přístup grid-search zejména ve vysokodimenzionálních prostorech.

Z hlediska praktického nasazení se navržený systém jeví jako perspektivní řešení jak pro akademické účely, tak pro reálné bezpečnostní infrastruktury. Vícestupňová architektura umožňuje adaptivní rozhodování na základě dostupnosti dat a výpočetních zdrojů, čímž zajišťuje efektivní klasifikaci i v prostředích s omezenou latencí a zdroji. Systém je zároveň navržen tak, aby byl snadno rozšiřitelný o další příznaky a modely a umožňoval integraci do větších detekčních platforem, jako je například nástroj DomainRadar.

Další směřování práce by mělo zahrnovat rozšíření datové sady o dynamické síťové příznaky a využití pokročilejších architektur hlubokého učení, zejména transformerových modelů. Rovněž je vhodné zkoumat možnosti aktivního a online učení, které umožní adaptivní aktualizaci modelů v reálném čase na základě nových dat. Pozornost by měla být věnována také sledování časové degradace modelů a návrhu mechanismů pro jejich pravidelné dotrénování.

V rámci práce vznikla rovněž sada vědeckých publikací, které dále rozvíjejí její dílčí výstupy. Jejich přehled je uveden v příloze C.

Závěrem lze konstatovat, že diplomová práce splnila vytyčené cíle, přinesla nové poznatky v oblasti detekce škodlivých domén a představuje pevný základ pro další výzkum i praktické využití v oblasti kybernetické bezpečnosti.

# Literatura


[1] Abdullah Aswad, S. Evaluation and Analysis Data from Twitter Data By Using Hybrid CNN and LTSM. In: Özseven, T., Sokol, Y. I., Yaşar, E. a Yevseiev, S., ed. *2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. Istanbul, Turkey: IEEE, 2023, s. 01–05. DOI: 10.1109/HORA58378.2023.10156756. ISBN Electronic ISBN: 979-8-3503-3752-5, Print on Demand (PoD) ISBN: 979-8-3503-3753-2.

[2] Abu Nimeh, S., Nappa, D., Wang, X. a Nair, S. A Comparison of Machine Learning Techniques for Phishing Detection. In: Anti-Phishing Working Group (APWG). *Proceedings of the 2007 eCrime Researchers Summit*. Association for Computing Machinery (ACM), October 2007, s. 60–69. DOI: 10.1145/1299015.1299021.

[3] abuse.ch. *ThreatFox*. 2023. Dostupné z: https://threatfox.abuse.ch/.

[4] Achddou, R., Di Martino, J. M. a Sapiro, G. Nested Learning for Multi-Level Classification. In: *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, s. 2815–2819. DOI: 10.1109/ICASSP39728.2021.9415076. ISBN 978-1-7281-7606-2.

[5] Alexander, T. a Braun, S. Using MISP for Collaborative Threat Intelligence in Practice. *Journal of Cyber Intelligence*. 2019, sv. 4, č. 2, s. 45–60. ISSN 2624-800X.

[6] Alpaydin, E. *Introduction to Machine Learning*. 2nd. Cambridge, Mass.: The MIT Press, únor 2010. ISBN 026201243X.

[7] American Registry for Internet Numbers. *ARIN WHOIS/RDAP Database* [https://www.arin.net]. 2023. Accessed 2025-05-12.

[8] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E. et al. Understanding the Mirai Botnet. In: *Proceedings of the 26th USENIX Security Symposium*. USENIX Association, 2017, s. 1093–1110. Dostupné z: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis.

[9] Antonakakis, M., Perdisci, R., Dagon, D., Lee, W. a Feamster, N. Building a Dynamic Reputation System for DNS. In: *Proceedings of the 19th USENIX Security Symposium*. Washington, DC: USENIX Association, 2010. Dostupné z: https://www.usenix.org/conference/usenixsecurity10/building-dynamic-reputation-system-dns.

[10] Bai, Q., Lam, H. a Sclaroff, S. *A Bayesian Approach for Online Classifier Ensemble* [arXiv preprint https://arxiv.org/abs/1507.02011]. 2015. DOI: 10.48550/arXiv.1507.02011.

[11] Barnes, R. a *et al.*. Use of Wildcard and Multi-Domain TLS Certificates. *Journal of Internet Engineering*. 2016, sv. 10, č. 1, s. 45–52. ISSN 1791-177X.

[12] Bellman, R. *Dynamic Programming*. 1. vyd. Princeton, NJ: Princeton University Press, 1957. ISBN 978-0-691-14668-3.

[13] Bergstra, J., Bardenet, R., Bengio, Y. a Kégl, B. Algorithms for Hyper-Parameter Optimization. In: Shawe–Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, N. a Weinberger, K. Q., ed. *Advances in Neural Information Processing Systems 24 (NeurIPS 2011)*. Granada, Spain: Curran Associates, 2011, s. 2546–2554. DOI: 10.5555/2986459.2986743. ISBN 9781618395993.

[14] Bilge, L., Kirda, E., Kruegel, C. a Balduzzi, M. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In: *Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS)*. The Internet Society, 2011. Dostupné z: https://www.ndss-symposium.org/ndss2011/exposure-finding-malicious-domains-using-passive-dns-analysis/.

[15] Breiman, L. Random Forests. *Machine Learning*. 2001, sv. 45, č. 1, s. 5–32. DOI: 10.1023/A:1010933404324. ISSN 1573-0565.

[16] CA/Browser Forum. *Guidelines for the Issuance and Management of Extended Validation TLS Certificates* [https://cabforum.org/working-groups/server/extended-validation/documents/]. 2017. Version 1.6.1, accessed 2025-05-12.

[17] Campos, B. a Silva, M. RDAP Adoption Challenges and Benefits: Enhancing Domain Name Transparency. *Computers & Security*. Elsevier. 2019, sv. 87, s. 101738. DOI: 10.1016/j.cose.2019.101738. ISSN 0167-4048.

[18] Chen, T. a Guestrin, C. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. San Francisco, CA, USA: Association for Computing Machinery, 2016, s. 785–794. DOI: 10.1145/2939672.2939785. ISBN 978-1-4503-4232-2.

[19] Cho, D. X., Tisenko, V., Nguyen, Q. D., Nguyen, Q. H. a Do, H. L. Malicious Domain Detection Based on DNS Query Using Machine Learning. *International Journal of Emerging Trends in Engineering Research*. World Academy of Research in Science and Engineering. May 2020, sv. 8, č. 5, s. 1809–1814. DOI: 10.30534/ijeter/2020/53852020. ISSN 2347-3983. Dostupné z: https://www.warse.org/IJETER/static/pdf/file/ijeter53852020.pdf.

[20] Cho, K., Merriënboer, B. V., Gulcehre, C., Bahdanau, D., Bougares, F. et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *ArXiv preprint arXiv:1406.1078*. 2014. Dostupné z: https://arxiv.org/abs/1406.1078.

[21] Chu, Z., Gianvecchio, S., Wang, H. a Jajodia, S. Who is tweeting on Twitter: Human, bot, or cyborg? In: ACM. *Proceedings of the 26th Annual Computer Security Applications Conference*. Austin, TX, USA: ACM, December 2010, s. 21–30.

[22] Cisco Systems, Inc.. *Cisco Umbrella* [https://umbrella.cisco.com/]. 2015. Accessed: 2025-04-28.

[23] Cisco Talos Intelligence Group. *PhishTank* [https://phishtank.org/]. 2006. Accessed 2023-02-24.

[24] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R. et al. *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*. 5280. Internet Engineering Task Force (IETF), May 2008. Dostupné z: https://www.rfc-editor.org/rfc/rfc5280.

[25] Cortes, C. a Vapnik, V. Support-Vector Networks. *Machine Learning*. Springer. 1995, sv. 20, č. 3, s. 273–297. DOI: 10.1007/BF00994018. ISSN 0885-6125.

[26] Dietterich, T. G. Ensemble methods in machine learning. *International workshop on multiple classifier systems*. Springer. 2000, s. 1–15.

[27] Domingos, P. A Few Useful Things to Know about Machine Learning. *Communications of the ACM*. ACM. 2012, sv. 55, č. 10, s. 78–87. DOI: 10.1145/2347736.2347755. ISSN 0001-0782.

[28] Feily, M., Shahrestani, A. a Ramadass, S. A Survey of Botnet and Botnet Detection. In: *Proceedings of the 3rd International Conference on Emerging Security Information, Systems and Technologies (SECURWARE 2009)*. Athens/Glyfada, Greece: IEEE Computer Society, 2009, s. 268–273. DOI: 10.1109/SECURWARE.2009.48. ISBN 978-0-7695-3741-7.

[29] Freund, Y. a Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*. 1997, sv. 55, č. 1, s. 119–139. DOI: 10.1006/jcss.1997.1504. ISSN 0022-0000. Dostupné z: https://www.sciencedirect.com/science/article/pii/S002200009791504X.

[30] Friedman, J. H. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*. Institute of Mathematical Statistics. 2001, sv. 29, č. 5, s. 1189–1232. DOI: 10.1214/aos/1013203451. ISSN 0090-5364.

[31] 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)*. Ghaziabad, India: IEEE, 2016, s. 166–169. DOI: 10.1109/CICT.2016.40. ISBN Electronic ISBN: 978-1-5090-0210-8, CD: 978-1-5090-0208-5, Print on Demand (PoD) ISBN: 978-1-5090-0211-5.

[32] Goodfellow, I., Bengio, Y. a Courville, A. *Deep Learning*. Cambridge, MA: MIT Press, 2016. Adaptive Computation and Machine Learning. ISBN 978-0-262-03561-3. Dostupné z: http://www.deeplearningbook.org.

[33] 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)*. Pointe aux Piments, Mauritius: IEEE, 2015, s. 1–6. DOI: 10.1109/CCCS.2015.7374132. ISBN 9781467393553.

[34] Guyon, I. a Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* JMLR.org. březen 2003, sv. 3, null, s. 1157–1182. ISSN 1532-4435.

[35] Hason, N., Dvir, A. a Hajaj, C. Robust Malicious Domain Detection. In: *Cyber Security Cryptography and Machine Learning. CSCML 2020, Lecture Notes in Computer Science*. Cham: Springer, 2020, sv. 12161, s. 45–61. ISBN 978-3-030-49784-2. Fourth International Symposium on Cyber Security, Cryptology, and Machine Learning (CSCML 2020).

[36] Hochreiter, S. a Schmidhuber, J. Long Short-Term Memory. *Neural Computation*. MIT Press. 1997, sv. 9, č. 8, s. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. ISSN 0899-7667.

[37] Horák, A. *Detekce škodlivých domén na základě externích zdrojů dat*. 2023. [cit. 2023-12-02]. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Radek Hranický, P.

[38] Horák, A., Polišenský, J. a Hranický, R. *Domain Collector* [https://www.fit.vut.cz/research/product/784/.cs]. 2023. Aplikace je ke stažení na webové stránce Fakulty informačních technologií VUT v Brně.

[39] Horák, A., Polišenský, J. a Hranický, R. *Sada zásuvných modulů pro systém QRadar* [https://www.fit.vut.cz/research/product/838/.cs]. 2023. Aplikace je ke stažení na webové stránce Fakulty informačních technologií VUT v Brně.

[40] Hranický, R., Horák, A., Polišenský, J., Jeřábek, K. a Ryšavý, O. Unmasking the Phishermen: Phishing Domain Detection with Machine Learning and Multi-Source Intelligence. In: *NOMS 2024-2024 IEEE Network Operations and Management Symposium*. 2024, s. 1–5. DOI: 10.1109/NOMS59830.2024.10575573.

[41] Hranický, R., Horák, A., Polišenský, J., Ondryáš, O., Jeřábek, K. et al. Spotting the Hook: Leveraging Domain Data for Advanced Phishing Detection. In: *2024 10th International Conference on Network and Service Management (CNSM)*. Praha: Institute of Electrical and Electronics Engineers, 2024, s. 1–7. DOI: 10.23919/CNSM62983.2024.10814617. ISBN 978-3-903176-66-9. Dostupné z: https://ieeexplore.ieee.org/document/10814617.

[42] Hranický, R., Polišenský, J. a Horák, A. *DomainRadar - Detector of Malicious Domains (prototype)* [https://www.fit.vut.cz/research/product/779/.cs]. 2022. Aplikace je ke stažení na webové stránce Fakulty informačních technologií VUT v Brně.

[43] Huang, W. a Zhang, X. A Comprehensive Evaluation of IP Geolocation Databases and Services. *ACM Computing Surveys*. 2017, sv. 50, č. 1, s. 36:1–36:34. DOI: 10.1145/3014392. ISSN 0360-0300.

[44] IP2Location. *IP2Location IP Geolocation Database* [https://www.ip2location.com]. 2023. Accessed 2025-05-12.

[45] Iwahana, K., Takemura, T., Cheng, J. C., Ashizawa, N., Umeda, N. et al. MADMAX: Browser-Based Malicious Domain Detection Through Extreme Learning Machine. *IEEE Access*. 2021, sv. 9, s. 78293–78314. DOI: 10.1109/ACCESS.2021.3080456.

[46] James, G., Witten, D., Hastie, T. a Tibshirani, R. *An Introduction to Statistical Learning: With Applications in R*. 1. vyd. New York: Springer, 2013. Springer Texts in Statistics. ISBN 978-1-4614-7137-0.

[47] Jiang, Y., Jia, M., Zhang, B. a Deng, L. Malicious Domain Name Detection Model Based on CNN-GRU-Attention. In: *2021 33rd Chinese Control and Decision Conference (CCDC)*. 2021, s. 1602–1607. DOI: 10.1109/CCDC52312.2021.9602373. ISSN 1948-9447.

[48] Jin, X., Du, X., Han, X., Sun, H. a Li, J. Fine Classification Method of Product Image Based on Multi-Level Convolutional Neural Networks. In: *2021 2nd Asia Symposium on Signal Processing (ASSP)*. 2021, s. 113–117. DOI: 10.1109/ASSP54407.2021.00025.

[49] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W. et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In: *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*. Curran Associates, 2017, s. 3146–3154. DOI: 10.5555/3294996.3295074. ISBN 9781510860964. Dostupné z: https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf.

[50] Kingma, D. P. a Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv preprint arXiv:1412.6980*. 2014. DOI: 10.48550/arXiv.1412.6980. Dostupné z: https://arxiv.org/abs/1412.6980.

[51] Komaitis, K. *The Current State of Domain Name Regulation: Domain Names as Second-Class Citizens in a Mark-Dominated World*. Abingdon, Oxon; New York: Routledge, 2010. Routledge Research in IT and E-Commerce Law. ISBN 978-0-415-47776-5.

[52] Kumar, A. a Maity, S. Detecting Malicious URLs using Lexical Analysis and Network Activities. In: *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*. 2022, s. 570–575. DOI: 10.1109/ICIRCA54612.2022.9985586.

[53] LeCun, Y., Bengio, Y. a Hinton, G. Deep learning. *Nature*. 2015, sv. 521, č. 7553, s. 436–444. DOI: 10.1038/nature14539. ISSN 0028-0836. Dostupné z: https://www.nature.com/articles/nature14539.

[54] LeCun, Y., Bottou, L., Bengio, Y. a Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*. IEEE. 1998, sv. 86, č. 11, s. 2278–2324. DOI: 10.1109/5.726791. ISSN 0018-9219.

[55] Liaw, A. a Wiener, M. Classification and Regression by `randomForest`. *R News*. 2002, sv. 2, č. 3, s. 18–22. ISSN 1609-3631. DOI není přidělen; open-access článek v časopise *R News*. Dostupné z: https://journal.r-project.org/articles/RN-2002-022/.

[56] Lundberg, S. M. a Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In: *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*. Long Beach, CA, USA: Curran Associates, 2017, s. 4765–4774. DOI: 10.5555/3295222.3295230. ISBN 978-1-5108-6096-4. Dostupné z: https://proceedings.neurips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

[57] 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 International Conference on World Wide Web (WWW 2009)*. Association for Computing Machinery, 2009, s. 1245–1254. DOI: 10.1145/1526709.1526921. ISBN 978-1-60558-487-4.

[58] Ma, J., Saul, L. K., Savage, S. a Voelker, G. M. Learning to detect malicious URLs. *Transactions on Information and System Security (TISSEC)*. New York, NY, USA: Association for Computing Machinery. květen 2011, sv. 2, č. 3. DOI: 10.1145/1961189.1961202. ISSN 2157-6904.

[59] Magalhães, F. a Magalhães, J. P. Adopting Machine Learning to Support the Detection of Malicious Domain Names. In: *2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*. 2020, s. 1–6. DOI: 10.1109/IOTSMS52051.2020.9340159.

[60] Mahmood, Y., Zafar, J., Karim, A., Iqbal, F. a Alazab, M. HTML-Based Phishing Website Detection Using Machine Learning Classifiers. *Computers & Security*. Elsevier. 2021, sv. 105, s. 102244. DOI: 10.1016/j.cose.2021.102244. ISSN 0167-4048.

[61] MaxMind. *GeoIP2 Databases* [https://www.maxmind.com/geoip2]. 2023. Accessed 2025-05-12.

[62] McDowell, J. C., Kuncic, Z., Soria, R. a Bicknell, G. V. Bayesian Analysis of the Dynamic Cosmic Web in the 2dF Galaxy Redshift Survey. *The Astrophysical Journal Letters*. IOP Publishing. December 2004, sv. 617, č. 1, s. L13–L16. DOI: 10.1086/427079. ISSN 2041-8213.

[63] McPherson, D. a Smith, J. Analysis of VPN and Proxy Services in IP Geolocation. *Journal of Network Security*. 2009, sv. 4, č. 1, s. 23–36. ISSN 1943-933X.

[64] MISP Project. *MISP: The Open Source Threat Intelligence Platform* [https://www.misp-project.org]. 2020. Accessed 2025-01-13.

[65] Mockapetris, P. V. *Domain Names - Concepts and Facilities* [https://www.rfc-editor.org/rfc/rfc882.txt]. Listopad 1983. RFC 882, IETF, November 1983. Obsoleted by RFC 1034.

[66] Moore, T. a Clayton, R. Examining the Impact of Website Take-down on Phishing. In: *Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit*. ACM, 2007, s. 1–13. DOI: 10.1145/1299015.1299016. ISBN 978-1-59593-939-5. Dostupné z: https://www.cl.cam.ac.uk/~rnc1/ecrime07.pdf.

[67] Ondryáš, O. *Effective Large-scale Collection of Information Related to Domain Names*. Brno, Czech Republic, 2024. Master's thesis. Brno University of Technology, Faculty of Information Technology. Vedoucí práce Ing. Radek Hranický, P.

[68] OpenPhish. *OpenPhish Threat Intelligence Feed* [https://openphish.com/]. 2014. Accessed on April 28, 2025.

[69] Palaniappan, G., S, S., Rajendran, B., Adiwal, S., Goyal, S. et al. Malicious Domain Detection Using Machine Learning On Domain Name Features, Host-Based Features and Web-Based Features. *Procedia Computer Science*. Elsevier. 2020, sv. 171, s. 654–661. DOI: 10.1016/j.procs.2020.04.071. ISSN 1877-0509. Dostupné z: https://www.sciencedirect.com/science/article/pii/S1877050920310383.

[70] Perdisci, R., Corona, I. a Giacinto, G. URL-based Web Page Classification: A New Method for URL-based Web Page Classification Using Lexical and Host-based Features. *Computer Networks*. Elsevier. 2012, sv. 56, č. 3, s. 1231–1247. DOI: 10.1016/j.comnet.2011.10.002. ISSN 1389-1286.

[71] Perdisci, R., Lee, W. a Feamster, N. Detection of malicious network traffic using machine learning techniques. In: Springer. *International Conference on Machine Learning and Data Mining in Pattern Recognition*. 2009, s. 331–345.

[72] Petr, P. *Optimalizace klasifikačních modelů pro detekci maligních domén*. 2023. [cit. 2025-01-03]. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Radek Hranický, P.

[73] Plohmann, D., Yakdan, K., Klatt, M., Bader, J. a Gerhards Padilla, E. A comprehensive measurement study of domain generating malware. *25th USENIX Security Symposium (USENIX Security 16)*. 2016, s. 263–278.

[74] Postel, J. *Domain names: Implementation specification* [RFC 883]. RFC Editor, listopad 1983. DOI: 10.17487/RFC0883. Dostupné z: https://www.rfc-editor.org/info/rfc883.

[75] Provos, N. a Holz, T. *Virtual honeypots: From botnet tracking to intrusion detection*. Addison-Wesley Professional, 2007. ISBN 9780321336323.

[76] Provos, N., McNamee, D., Mavrommatis, P., Wang, K. a Modadugu, N. The Ghost in the Browser: Analysis of Web-based Malware. In: *Proceedings of the First Workshop on Hot Topics in Understanding Botnets (HotBots '07)*. USENIX Association, 2007, s. 1–7. Dostupné z: https://www.usenix.org/conference/hotbots-07/ghost-browser-analysis-web-based-malware.

[77] Quinlan, J. R. Induction of Decision Trees. *Machine Learning*. Springer. March 1986, sv. 1, č. 1, s. 81–106. DOI: 10.1007/BF00116251. ISSN 0885-6125.

[78] Rahbarinia, B., Perdisci, R. a Antonakakis, M. Efficient and Accurate Behavior-Based Tracking of Malware-Control Domains in Large ISP Networks. *ACM Trans. Priv. Secur.* New York, NY, USA: Association for Computing Machinery. srpen 2016, sv. 19, č. 2. DOI: 10.1145/2960409. ISSN 2471-2566.

[79] Rescorla, E. The Transport Layer Security (TLS) Protocol Version 1.3. *RFC 8446, Internet Engineering Task Force*. IETF. August 2018. DOI: 10.17487/RFC8446. ISSN 2070-1721.

[80] Sahoo, D., Liu, C. a Hoi, S. C. H. *Malicious URL Detection using Machine Learning: A Survey*. 2017. Dostupné z: https://arxiv.org/abs/1701.07179.

[81] Senator, T. E. Multi-Stage Classification. In: *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM '05)*. Houston, TX, USA: IEEE Computer Society, November 2005, s. 386–393. DOI: 10.1109/ICDM.2005.102. ISBN 0-7695-2278-5.

[82] Shah, P. a Kumar, R. RDAP: A New Protocol for Domain Name and IP Address Registration Data Access. *Journal of Cybersecurity and Privacy*. MDPI. 2021, sv. 5, č. 1, s. 45–60. DOI: 10.3390/jcp5010003. ISSN 2624-800X.

[83] Shi, Y., Chen, G. a Li, J. Malicious Domain Name Detection Based on Extreme Machine Learning. *Neural Processing Letters*. December 2018, sv. 48, č. 3, s. 1347–1357. DOI: 10.1007/s11063-017-9666-7.

[84] Silveira, M. R., Silva, L. Marcos da, Cansian, A. M. a Kobayashi, H. K. Detection of Newly Registered Malicious Domains through Passive DNS. In: *2021 IEEE International Conference on Big Data (Big Data)*. 2021, s. 3360–3369. DOI: 10.1109/BigData52589.2021.9671348.

[85] Snoek, J., Larochelle, H. a Adams, R. P. Practical Bayesian Optimization of Machine Learning Algorithms. In: *Advances in Neural Information Processing Systems 25 (NIPS 2012)*. Lake Tahoe, NV, USA: Neural Information Processing Systems Foundation, 2012, s. 2951–2959. DOI: 10.5555/2999325.2999464. ISBN 978-1-62748-003-1. Dostupné z: https://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.

[86] Srinivasan, M. S., Chawla, S. a Bowyer, K. Detecting Android Malware using Machine Learning on Network Traffic. *Pattern Analysis and Applications*. Springer. 2018, sv. 21, č. 1, s. 111–125. DOI: 10.1007/s10044-017-0621-4. ISSN 1433-7541.

[87] Stallings, W. *Cryptography and Network Security: Principles and Practice*. 7. vyd. Pearson, 2017. ISBN 978-0-13-444428-4.

[88] Thomas, K., McCoy, D., Grier, C., Kolcz, A. a Paxson, V. Ad injection at scale: Assessing deceptive advertisement modifications. In: IEEE. *2015 IEEE Symposium on Security and Privacy*. 2015, s. 151–167.

[89] Torroledo, I., Camacho, L. D. a Bahnsen, A. C. Hunting Malicious TLS Certificates with Deep Neural Networks. In: *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*. New York, NY, USA: Association for Computing Machinery, 2018, s. 64–73. AISec '18. DOI: 10.1145/3270101.3270105. ISBN 9781450360043. Dostupné z: https://doi.org/10.1145/3270101.3270105.

[90] VirusTotal. *VirusTotal API v3 Overview* [https://docs.virustotal.com/reference/overview]. 2023. Accessed 2025-05-12.

[91] Wagner, C., Dulaunoy, A., Wagener, G. a Iklody, A. MISP: The Design and Implementation of a Collaborative Threat Intelligence Sharing Platform. In: *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*. New York, NY, USA: Association for Computing Machinery, 2016, s. 49–56. WISCS '16. DOI: 10.1145/2994539.2994542. ISBN 9781450345651.

[92] Wang, Y., Zhu, S. a Li, C. Research on Multistep Time Series Prediction Based on LSTM. In: *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*. Xiamen, China: IEEE, 2019, s. 1155–1159. DOI: 10.1109/EITCE47263.2019.9095044. ISBN Electronic ISBN: 978-1-7281-3584-7, CD: 978-1-7281-3583-0, Print on Demand (PoD) ISBN: 978-1-7281-3585-4.

[93] Wolpert, D. H. Stacked Generalization. *Neural Networks*. Pergamon Press / Elsevier. April 1992, sv. 5, č. 2, s. 241–259. DOI: 10.1016/S0893-6080(05)80023-1. ISSN 0893-6080.

[94] Yadav, S., Reddy, A. K. K., Reddy, A. N. a Ranjan, S. Detecting algorithmically generated domain-flux attacks with DNS traffic analysis. In: IEEE. *IEEE Network Operations and Management Symposium*. 2010, s. 1–8.

[95] Zhao, L., Liu, J., Peng, X. a Li, J. Malicious Domain Names Detection Based on Deep Learning and Random Forest. *Security and Communication Networks*. Hindawi. 2020, sv. 2020, s. 1–12. DOI: 10.1155/2020/8845147. ISSN 1939-0122.

[96] Zhauniarovich, Y., Khalil, I., Yu, T. a Dacier, M. A Survey on Malicious Domains Detection through DNS Data Analysis. *ACM Computing Surveys*. ACM. 2018, sv. 51, č. 4, s. 1–36. DOI: 10.1145/3191329. Dostupné z: https://dl.acm.org/doi/10.1145/3191329.

[97] Zhu, J. a Zou, F. Detecting Malicious Domains Using Modified SVM Model. In: *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 2019, s. 492–499. DOI: 10.1109/HPCC/SmartCity/DSS.2019.00079.

# Příloha A

# Obsah přiloženého paměťového média

V přiloženém digitálním úložišti se nalézá:

- tento dokument ve formátu PDF,
- zdrojové soubory tohoto dokumentu,
- zdrojové kódy vyvinutého klasifikačního systému,
- klasifikátory které vznikly v rámci této práce,
- trénovací, validační a verifikační datové sady,
- návod k použití přiloženého klasifikačního systému.

# Příloha B

# Manuál

Veškerý zdrojový kód, trénovací skripty, výsledky experimentů i příklady použití klasifikační pipeline jsou volně dostupné v repozitáři na platformě Github:

https://github.com/poli-cz/Domain-Ensemble-pipeline

Tento repozitář slouží jako praktický doplněk k této diplomové práci a umožňuje plnou replikaci a rozšíření prezentovaného řešení.

## Přiložený software

Repozitář obsahuje kompletní implementaci vícestupňové klasifikační pipeline určené pro detekci maligních domén. Kromě samotného systému zahrnuje také sadu předtrénovaných modelů, podpůrné utility pro načítání dat, vizualizaci výstupů a moduly pro výpočet interpretací pomocí SHAP analýzy.

### Trénování klasifikátorů

Trénovací skripty jsou připraveny ve formě Jupyter notebooků a jsou umístěny ve složce:

`src/training/`

Pro každý klasifikátor je připraven samostatný notebook:

- `feedforward_train.ipynb` – trénink plně propojené neuronové sítě (FFNN)
- `cnn_train.ipynb` – trénink konvoluční neuronové sítě (CNN)
- `svm_train.ipynb` – trénink klasifikátoru svm
- `xgb_lgbm_train.ipynb` – trénink stromových modelů (XGBoost a LightGBM)
- `meta_model_train.ipynb` – trénink rozhodovacího meta-modelu a modulu pro detekci falešně pozitivních vzorků (FPD)

Trénink probíhá na vstupních datasetech ve formátu .parquet, které je nutné stáhnout samostatně (například z repozitáře `domainradar-clf`) a umístit do složky:

`src/parkets/`

### Běh klasifikačního systému

Pro spuštění klasifikační pipeline na nových doménách slouží notebook:

`src/ensemble_pipeline_example.ipynb`

V tomto notebooku je demonstrováno:

- inicializace pipeline včetně načtení předtrénovaných modelů
- klasifikace vstupních vzorků domén
- volitelná interpretace výstupu pomocí SHAP hodnot

Pipeline automaticky detekuje dostupnost jednotlivých příznakových kategorií (lexikální, DNS, RDAP, TLS apod.) a adaptivně zvolí odpovídající klasifikační model.

Konkrétní verze použitých modelů lze upravit ve skriptu:

`src/pipeline.py`

## Adresáře

Tato sekce přílohy stručně popisuje strukturu repozitáře, který obsahuje implementaci klasifikační pipeline pro detekci škodlivých domén. Systém tvoří samostatný, plně funkční modul pro trénink, inferenci a vyhodnocování modelů. Repozitář je rozdělen do adresářů podle funkce jednotlivých komponent:

- **/src** – Hlavní adresář s implementací klasifikační pipeline. Obsahuje datové transformace, načítání modelů, tréninkové notebooky, SHAP analýzu, i demonstrační příklad použití.
- **/src/core** – Základní komponenty pipeline: načítání a segmentace dat, modely metaklasifikace, detekce falešně pozitivních vzorků, a pomocné utility.
- **/src/models** – Vytrénované modely všech architektur (Keras, LightGBM, SVM, XGBoost) a jejich potřebné scalery. Obsahuje i samostatné složky pro metamodel a FPD modul.
- **/src/scalers** – Předtrénované modely pro normalizace a škálování uložené pomocí knihovny `joblib`, potřebné při inferenci dat.
- **/src/data** – Serializované validační a verifikační datasety (ve formátu `.pkl`) pro jednotlivé fáze klasifikace.
- **/src/parkets** – Vstupní datasety ve formátu `parquet`. Obsahuje anonymizovaná i neanonymizovaná data, HTML varianty, a testovací podmnožiny.
- **/src/results** – Výstupní grafy a výsledky vyhodnocení, např. konfuzní matice modelů.
- **/src/tmp** – Dočasné a experimentální výstupy vzniklé během ladění pipeline, např. serializované výsledky, podmnožiny datasetů, nebo mezivýstupy modelů.

- **/src/training** – Trénovací Jupyter notebooky pro jednotlivé modely: FFNN, CNN, SVM, LightGBM/XGBoost, attention modely a metamodel.
- **/src/tex_sources** – Pomocné LaTeX soubory a tabulky s metrikami, použité při psaní práce.
- **/src/ensemble_pipeline_example.ipynb** – Příkladový notebook demonstrující kompletní klasifikaci vstupních domén pomocí finální pipeline.
- **/docs** – Dokumentace a podpůrné materiály k diplomové práci, zejména vizualizace, diagramy a SHAP grafy.
- **/docs/figures** – Všechny výstupní obrázky, včetně agregovaných výsledků, SHAP analýz, architektur a porovnání.
- **/docs/figures/confusion_matrices** – Konfuzní matice všech modelů ve všech fázích tréninku a verifikace.
- **/docs/tex_sources** – Soubory použitých LaTeX tabulek, sloupcových dat a automaticky generovaných výsledků metrik.
- **/experiments** – Experimenty s mřížkami příznaků, porovnání modelů, SHAP analýzou a ladění pipeline.
- **/experiments/grids** – CSV soubory obsahující výsledky mřížkových vyhodnocení příznakových subsetů pro phishing i malware.
- **/experiments/shap** – SHAP analýzy, skripty a záložní výstupy přínosů jednotlivých příznaků pro vybrané modely.
- **/tests** – Jednotkové testy pro ověření základní funkčnosti vybraných komponent pipeline.
- **README.md** – Úvodní dokumentace repozitáře (v anglickém jazyce), s návodem na spuštění, trénink a inferenci.
- **poetry.lock, pyproject.toml** – Konfigurační soubory pro správu Python prostředí pomocí nástroje `Poetry`.

# Příloha C

# Publikační činnost

S touto diplomovou prací úzce souvisí několik vědeckých publikací, na jejichž jsem spoluautorem. Tyto publikace vznikly v průběhu řešení práce a pokrývají různé aspekty detekce maligních domén.

- **Unmasking the Phishermen: Phishing Domain Detection with Machine Learning and Multi-Source Intelligence** (publikováno na konferenci *IEEE/IFIP Network Operations and Management Symposium (NOMS 2024)*) Článek se zabývá detekcí phishingových domén pomocí kombinace vícerozdrojových dat (DNS, RDAP, TLS, IP) a využívá ensemble modely pro zvýšení přesnosti klasifikace. Zvláštní důraz je kladen na reálnou aplikovatelnost modelů a nízkou míru falešně pozitivních detekcí [40].
- **Spotting the Hook: Leveraging Domain Data for Advanced Phishing Detection** (publikováno na konferenci *IEEE CNSM 2024*) Publikace představuje 143položkový příznakový vektor pro phishingovou klasifikaci a hodnotí jeho efektivitu napříč sedmi strojově učenými modely. Výsledky ukazují velmi vysokou přesnost (0,983) a nízkou chybovost díky využití multi-modalních vstupních dat [41].
- **A Multi-Dimensional DNS Domain Intelligence Dataset for Cybersecurity Research** (v recenzním řízení, žurnál *Data in Brief*) Tento článek popisuje rozsáhlou datovou sadu více než 1 milionu anotovaných domén (benigních, phishingových a malware), včetně metodologie sběru, transformace a kategorizace dat ze čtyř hlavních zdrojů: DNS, RDAP, TLS a GeoIP.
- **Digital Wolves in Sheep's Clothing: Detecting Malicious Domains using a Multi-Stage Classifier Pipeline** (v přípravě, žurnál *IEEE Access*) Článek se věnuje návrhu a experimentálnímu vyhodnocení vícestupňové klasifikační pipeline s paralelními modely, rozhodovacím metaklasifikátorem a modulem pro detekci falešných pozitiv. Přímým základem článku je implementace uvedená v této práci.
- **DomainRadar: A Data-Driven Approach to Malicious Domain Identification** (v přípravě, žurnál *IEEE Transactions on Information Forensics and Security (TIFS)*) Tento článek popisuje vývoj a architekturu systému *DomainRadar*, který integruje výsledky této práce do automatizovaného nástroje pro detekci maligních domén v síťovém provozu. Zaměřuje se na praktické nasazení v prostředí bezpečnostních analytiků a SOC týmů.

# Příloha D

# Přehled použitých příznaků

V této příloze je uvedena kompletní tabulka použitých příznaků (feature vektor), které byly analyzovány a využívány v klasifikátorech v rámci této práce. Celkem se jedná o 243 příznaků.

Tabulka D.1: Overview of used features

| **Feature name** | **Description** |
|---|---|
| *Lexical features (lex_)* | |
| lex_name_len | Length of the domain name |
| lex_has_digit | True if domain name contains a digit |
| lex_phishing_keyword_count | Number of phishing keywords found |
| lex_benign_keyword_count | Number of benign keywords found |
| lex_consecutive_chars | Longest consecutive character sequence |
| lex_tld_len | Length of the TLD |
| lex_tld_abuse_score | Abuse score of the TLD |
| lex_tld_hash | Hash of the TLD |
| lex_sld_len | Length of the SLD |
| lex_sld_norm_entropy | Normalised entropy of the SLD |
| lex_sld_digit_count | Number of digits in the SLD |
| lex_sld_digit_ratio | Ratio of digits in the SLD |
| lex_sld_phishing_keyword_count | Number of phishing keywords in SLD |
| lex_sld_vowel_count | Number of vowels in SLD |
| lex_sld_vowel_ratio | Ratio of vowels in SLD |
| lex_sld_consonant_count | Number of consonants in SLD |
| lex_sld_consonant_ratio | Ratio of consonants in SLD |
| lex_sld_non_alphanum_count | Number of non-alphanumeric chars in SLD |
| lex_sld_non_alphanum_ratio | Ratio of non-alphanumeric chars in SLD |
| lex_sld_hex_count | Number of hex characters in SLD |
| lex_sld_hex_ratio | Ratio of hex characters in SLD |
| lex_sub_count | Number of subdomains |
| lex_stld_unique_char_count | Unique characters in TLD+SLD |
| lex_begins_with_digit | True if domain begins with digit |
| lex_www_flag | True if domain starts with 'www' |

*Continued on next page*

| Feature name | Description |
|---|---|
| lex_sub_max_consonant_len | Longest consonant sequence in subdomain |
| lex_sub_norm_entropy | Normalised entropy of subdomains |
| lex_sub_digit_count | Number of digits in subdomains |
| lex_sub_digit_ratio | Ratio of digits in subdomains |
| lex_sub_vowel_count | Number of vowels in subdomains |
| lex_sub_vowel_ratio | Ratio of vowels in subdomains |
| lex_sub_consonant_count | Number of consonants in subdomains |
| lex_sub_consonant_ratio | Ratio of consonants in subdomains |
| lex_sub_non_alphanum_count | Non-alphanumeric count in subdomains |
| lex_sub_non_alphanum_ratio | Ratio of non-alphanumerics in subdomains |
| lex_sub_hex_count | Hex characters in subdomains |
| lex_sub_hex_ratio | Ratio of hex in subdomains |
| lex_phishing_bigram_matches | Phishing bigram matches |
| lex_phishing_trigram_matches | Phishing trigram matches |
| lex_phishing_tetragram_matches | Phishing tetragram matches |
| lex_phishing_pentagram_matches | Phishing pentagram matches |
| lex_malware_bigram_matches | Malware bigram matches |
| lex_malware_trigram_matches | Malware trigram matches |
| lex_malware_tetragram_matches | Malware tetragram matches |
| lex_dga_bigram_matches | DGA bigram matches |
| lex_dga_trigram_matches | DGA trigram matches |
| lex_dga_tetragram_matches | DGA tetragram matches |
| lex_avg_part_len | Average part length |
| lex_stdev_part_lens | Standard deviation of part lengths |
| lex_longest_part_len | Longest domain part length |
| lex_short_part_count | Count of short domain parts |
| lex_medium_part_count | Count of medium domain parts |
| lex_long_part_count | Count of long domain parts |
| lex_superlong_part_count | Count of superlong domain parts |
| lex_shortest_sub_len | Shortest subdomain length |
| lex_ipv4_in_domain | True if IPv4 in domain name |
| lex_has_trusted_suffix | True if trusted suffix used |
| lex_has_wellknown_suffix | True if well-known suffix used |
| lex_has_cdn_suffix | True if CDN suffix used |
| lex_has_vps_suffix | True if VPS suffix used |
| lex_has_img_suffix | True if image suffix used |
| lex_suffix_score | Computed suffix score |
| *DNS-based features (dns_)* | |
| dns_has_dnskey | DNSKEY record present |
| dns_A_count | Number of A records |
| dns_AAAA_count | Number of AAAA records |
| dns_MX_count | Number of MX records |
| dns_NS_count | Number of NS records |
| dns_TXT_count | Number of TXT records |

*Continued on next page*

| Feature name | Description |
|---|---|
| dns_SOA_count | Number of SOA records |
| dns_CNAME_count | Number of CNAME records |
| dns_zone_level | Zone domain level |
| dns_zone_digit_count | Digits in zone domain |
| dns_zone_len | Zone domain length |
| dns_zone_entropy | Zone domain entropy |
| dns_resolved_record_types | Number of RR types resolved |
| dns_dnssec_score | DNSSEC score (always zero) |
| dns_ttl_avg | Average TTL |
| dns_ttl_stdev | TTL standard deviation |
| dns_ttl_low | Count of TTL in [0,100] |
| dns_ttl_mid | Count of TTL in [101,500] |
| dns_ttl_distinct_count | Distinct TTL values |
| dns_soa_primary_ns_level | Primary NS domain level |
| dns_soa_primary_ns_digit_count | Digits in primary NS |
| dns_soa_primary_ns_len | Length of primary NS domain |
| dns_soa_primary_ns_entropy | Entropy of primary NS domain |
| dns_soa_email_level | Admin email domain level |
| dns_soa_email_digit_count | Digits in admin email domain |
| dns_soa_email_len | Length of admin email domain |
| dns_soa_email_entropy | Entropy of admin email domain |
| dns_soa_refresh | SOA refresh value |
| dns_soa_retry | SOA retry value |
| dns_soa_expire | SOA expire value |
| dns_soa_min_ttl | SOA minimum TTL |
| dns_domain_name_in_mx | MX is a subdomain of domain |
| dns_mx_avg_len | Average length of MX names |
| dns_mx_avg_entropy | Average entropy of MX names |
| dns_txt_avg_len | Average TXT record length |
| dns_txt_avg_entropy | Average entropy of TXT records |
| dns_txt_external_verification_score | Known verification strings in TXT |
| dns_txt_spf_exists | SPF found in TXT records |
| dns_txt_dkim_exists | DKIM found in TXT records |
| dns_txt_dmarc_exists | DMARC found in TXT records |
| *IP-based features (ip_)* | |
| ip_count | Number of IP addresses |
| ip_mean_average_rtt | Average round-trip time (ICMP) |
| ip_v4_ratio | IPv4 to all IPs ratio |
| ip_a_aaaa_to_all_ratio | A/AAAA to all IPs ratio |
| ip_entropy | Entropy of IP prefixes (/16 and /64) |
| ip_as_address_entropy | Entropy of AS address prefixes |
| ip_asn_entropy | Entropy of ASN numbers |
| ip_distinct_as_count | Number of distinct ASNs |
| *TLS-based features (tls_)* | |

*Continued on next page*

| Feature name | Description |
|---|---|
| tls_has_tls | True if TLS connection established |
| tls_chain_len | Certificate chain length |
| tls_is_self_signed | True if certificate is self-signed |
| tls_root_authority_hash | Hash of root authority name |
| tls_leaf_authority_hash | Hash of leaf authority name |
| tls_negotiated_version_id | Negotiated TLS version ID |
| tls_negotiated_cipher_id | Negotiated TLS cipher ID |
| tls_root_cert_validity_len | Root certificate validity period |
| tls_leaf_cert_validity_len | Leaf certificate validity period |
| tls_broken_chain | Broken certificate chain present |
| tls_expired_chain | Expired certificate present |
| tls_total_extension_count | Total number of certificate extensions |
| tls_critical_extensions | Number of critical extensions |
| tls_with_policies_crt_count | Number of certificates with policies |
| tls_percentage_crt_with_policies | certificates with policies |
| tls_x509_anypolicy_crt_count | Certificates with anyPolicy extension |
| tls_iso_policy_crt_count | ISO policy certificates (OID 1.) |
| tls_joint_isoitu_policy_crt_count | Joint ISO-ITU policies (OID 2.) |
| tls_subject_count | Subject alternative names count |
| tls_server_auth_crt_count | Server authentication certificates |
| tls_client_auth_crt_count | Client authentication certificates |
| tls_CA_certs_in_chain_ratio | CA certificates to all certificates ratio |
| tls_unique_SLD_count | Unique second-level domains (SANs) |
| tls_common_name_count | Number of common names in certificates |
| *Geolocation-based features (geo_)* | |
| geo_countries_count | Number of distinct countries |
| geo_continents_count | Number of distinct continents |
| geo_malic_host_country | Number of IPs from malicious countries |
| geo_lat_stdev | Latitude standard deviation |
| geo_lon_stdev | Longitude standard deviation |
| geo_mean_lat | Mean latitude |
| geo_mean_lon | Mean longitude |
| geo_min_lat | Minimum latitude |
| geo_max_lat | Maximum latitude |
| geo_min_lon | Minimum longitude |
| geo_max_lon | Maximum longitude |
| geo_lat_range | Latitude range |
| geo_lon_range | Longitude range |
| geo_centroid_lat | Centroid latitude |
| geo_centroid_lon | Centroid longitude |
| geo_estimated_area | Estimated area |
| geo_continent_hash | Hash of continents |
| geo_countries_hash | Hash of countries |
| *Domain RDAP features (rdap_)* | |

*Continued on next page*

| Feature name | Description |
|---|---|
| rdap_registration_period | Domain registration period |
| rdap_domain_age | Domain age (days) |
| rdap_time_from_last_change | Time since last change |
| rdap_domain_active_time | Active time of the domain |
| rdap_has_dnssec | True if DNSSEC enabled (RDAP) |
| rdap_registrar_name_len | Length of registrar's name |
| rdap_registrar_name_entropy | Entropy of registrar's name |
| rdap_registrar_name_hash | Hash of registrar's name |
| rdap_registrant_name_len | Length of registrant's name |
| rdap_registrant_name_entropy | Entropy of registrant's name |
| rdap_admin_name_len | Length of admin contact's name |
| rdap_admin_name_entropy | Entropy of admin contact's name |
| rdap_admin_email_len | Length of admin email |
| rdap_admin_email_entropy | Entropy of admin email |
| *IP RDAP features (rdap_ip_)* | |
| rdap_ip_v4_count | IPv4 addresses with RDAP data |
| rdap_ip_v6_count | IPv6 addresses with RDAP data |
| rdap_ip_shortest_v4_prefix_len | Shortest IPv4 prefix length |
| rdap_ip_longest_v4_prefix_len | Longest IPv4 prefix length |
| rdap_ip_shortest_v6_prefix_len | Shortest IPv6 prefix length |
| rdap_ip_longest_v6_prefix_len | Longest IPv6 prefix length |
| rdap_ip_avg_admin_name_len | Avg. length of admins' names |
| rdap_ip_avg_admin_name_entropy | Avg. entropy of admins' names |
| rdap_ip_avg_admin_email_len | Avg. length of admins' emails |
| rdap_ip_avg_admin_email_entropy | Avg. entropy of admins' emails |
| *HTML-based features (html_)* | |
| html_num_of_tags | Total number of HTML tags |
| html_num_of_paragraphs | Number of `<p>` tags |
| html_num_of_divs | Number of `<div>` tags |
| html_num_of_titles | Number of `<title>` tags |
| html_num_of_external_js | External JavaScript files |
| html_num_of_links | Number of `<link>` tags |
| html_num_of_scripts | Number of `<script>` tags |
| html_num_of_scripts_async | Async scripts count |
| html_num_of_scripts_type | Scripts with explicit type |
| html_num_of_anchors | Number of anchors (`<a>`) |
| html_num_of_anchors_to_hash | Anchors to # fragments |
| html_num_of_anchors_to_https | Anchors to HTTPS links |
| html_num_of_anchors_to_com | Anchors to .com domains |
| html_num_of_inputs | Number of input fields |
| html_num_of_input_password | Password inputs |
| html_num_of_hidden_elements | Hidden elements |
| html_num_of_input_hidden | Hidden input fields |
| html_num_of_objects | Number of `<object>` tags |

*Continued on next page*

| Feature name | Description |
|---|---|
| html_num_of_embeds | Number of `<embed>` tags |
| html_num_of_frame | Number of `<frame>` tags |
| html_num_of_iframe | Number of `<iframe>` tags |
| html_num_of_iframe_src | iFrame with `src` attribute |
| html_num_of_iframe_src_https | iFrame to HTTPS source |
| html_num_of_center | Number of `<center>` tags |
| html_num_of_imgs | Number of images (`<img>`) |
| html_num_of_imgs_src | Images with `src` attribute |
| html_num_of_meta | Number of `<meta>` tags |
| html_num_of_links_href | Links with `href` |
| html_num_of_links_href_https | Links to HTTPS targets |
| html_num_of_links_href_css | Links to CSS stylesheets |
| html_num_of_links_type | Links with `type` attribute |
| html_num_of_link_type_app | Links to application types |
| html_num_of_link_rel | Links with `rel` attribute |
| html_num_of_all_hrefs | Total href attributes |
| html_num_of_form_action | Forms with an action attribute |
| html_num_of_form_http | Forms posting to HTTP (non-HTTPS) |
| html_num_of_strong | Number of `<strong>` tags |
| html_no_hrefs | True if no hrefs found |
| html_internal_href_ratio | Internal href ratio |
| html_num_of_internal_hrefs | Number of internal hrefs |
| html_external_href_ratio | External href ratio |
| html_num_of_external_href | Number of external hrefs |
| html_num_of_icon | Number of icons |
| html_icon_external | External icon usage |
| html_num_of_form_php | Forms targeting PHP files |
| html_num_of_form_hash | Forms submitting to # fragments |
| html_num_of_form_js | Forms submitting to JavaScript |
| html_malicious_form | Malicious form detection |
| html_most_common | Most common HTML tag used |
| html_num_of_css_internal | Number of internal CSS styles |
| html_num_of_css_external | Number of external CSS links |
| html_num_of_anchors_to_content | Anchors pointing to page content |
| html_num_of_anchors_to_void | Anchors with void targets |
| html_num_of_words | Total number of words |
| html_num_of_lines | Total number of lines |
| html_unique_words | Number of unique words |
| html_average_word_len | Average word length |
| html_blocked_keywords_label | Presence of blocked keywords |
| html_num_of_blank_spaces | Number of blank spaces |
| html_create_element | Usage of `createElement` |
| html_write | Usage of `document.write` |
| html_char_code_at | Usage of `charCodeAt` |

*Continued on next page*

| Feature name | Description |
|---|---|
| html_concat | Usage of `concat` function |
| html_escape | Usage of `escape` function |
| html_eval | Usage of `eval` |
| html_exec | Usage of `exec` |
| html_from_char_code | Usage of `fromCharCode` |
| html_link | Usage of `link` |
| html_parse_int | Usage of `parseInt` |
| html_replace | Usage of `replace` |
| html_search | Usage of `search` |
| html_substring | Usage of `substring` |
| html_unescape | Usage of `unescape` |
| html_add_event_listener | Usage of `addEventListener` |
| html_set_interval | Usage of `setInterval` |
| html_set_timeout | Usage of `setTimeout` |
| html_push | Usage of `push` function |
| html_index_of | Usage of `indexOf` function |
| html_document_write | Usage of `document.write()` |
| html_get | Usage of `get` function |
| html_find | Usage of `find` function |
| html_document_create_element | Usage of `document.createElement()` |
| html_window_set_timeout | Usage of `window.setTimeout()` |
| html_window_set_interval | Usage of `window.setInterval()` |
| html_hex_encoding | Hexadecimal string encoding detected |
| html_unicode_encoding | Unicode escape encoding detected |
| html_long_variable_name | Long variable names used |

# Příloha E

# Specializovaná klasifikace na základě TLS příznaků

Tato příloha rozšiřuje hlavní část práce o podrobnosti týkající se experimentální klasifikace domén výhradně na základě TLS příznaků. Přístup je motivován zjištěním, že TLS příznaky vykazují v původním modelu nízký agregovaný přínos, a přesto mohou představovat cenný doplňkový zdroj informací, jak bylo prokázáno například ve studii Torroleda et al. [89].

## Motivace

Na základě analýzy Shapleyho hodnot (viz Obr. 6.1) byla identifikována skupina TLS příznaků jako oblast s nejnižším průměrným přínosem (0,046). Tento výsledek naznačuje, že původní sada TLS atributů byla podhodnocena, a přitom podle literatury skýtá významný detekční potenciál. Rozhodli jsme se proto navrhnout rozšířený extrakční a klasifikační systém zaměřený právě na tuto doménu.

## Význam původních TLS příznaků

Tabulka E.1 shrnuje původní TLS příznaky a jejich přínos dle metody SHAP.

| **Příznak** | **SHAP hodnota** |
|---|---|
| `tls_root_cert_validity_remaining` | 1,5850 |
| `tls_leaf_cert_validity_len` | 0,3612 |
| `tls_root_cert_validity_len` | 0,2098 |
| `tls_leaf_cert_validity_remaining` | 0,1922 |
| `tls_total_extension_count` | 0,1594 |
| `tls_joint_isoitu_policy_crt_count` | 0,1279 |
| `tls_unique_SLD_count` | 0,1189 |
| `tls_version_id` | 0,1129 |
| `tls_cipher_id` | 0,0774 |
| `tls_CA_certs_in_chain_ratio` | 0,0691 |

Tabulka E.1: Význam vybraných TLS příznaků podle analýzy metodou SHAP.

## Hloubková analýza a rozšíření TLS příznaků

Na základě výsledků analýzy Shapleyho hodnot bylo zřejmé, že původní TLS příznaky vykazují v rámci celkové klasifikace relativně nízký přínos. Přestože literatura naznačuje jejich potenciál při detekci anomálií a škodlivých entit v síťovém provozu [89], jejich základní reprezentace ve výchozí sadě atributů nebyla zjevně dostačující.

Z tohoto důvodu byla provedena hloubková analýza obsahu TLS certifikátů a navrženo rozšíření extrakční logiky. Východiskem byl nástroj `DomainRadar` [42], vyvíjený v rámci projektu FETA, jehož podrobnější popis se nachází v sekci 5.3.1.

Cílem bylo vytvořit obohacenou sadu TLS atributů, která lépe vystihuje strukturu a vlastnosti certifikátového řetězce a umožní přesnější klasifikaci domén v kontextu šifrované komunikace.

### Rozšířené charakteristiky TLS certifikátů

V rámci nové extrakční logiky byly zpracovány zejména následující prvky:

- **Výpočet entropie:** Shannonova entropie textových polí organizace a vydavatele certifikátu, indikující nestandardní nebo syntetické hodnoty.
- **Hloubka řetězce:** Počet certifikátů v řetězci jako ukazatel důvěryhodnosti a složitosti infrastruktury.
- **Bezpečnostní politiky:** Přítomnost politik a rozšíření dle standardů X.509 a ISO.
- **Kombinované ukazatele:** Poměry a rozdíly mezi délkami platnosti, počet rozšíření a relace mezi jednotlivými vrstvami řetězce.

### Ukázkový výpis zpracovaného certifikátu

Následující výpis ukazuje reálnou strukturu TLS certifikátu zpracovaného systémem:

```
{
    "protocol": "TLSv1.3",
    "cipher": "TLS_AES_256_GCM_SHA384",
    "count": 4,
    "certificates": [
        {
            "common_name": "E1",
            "country": "US",
            "is_root": false,
            "organization": "Let's Encrypt",
            "valid_len": 7775999,
            "validity_start": "2024-04-27 10:25:58",
            "validity_end": "2024-07-26 10:25:57",
            "extension_count": 9,
            "extensions": [
                {
                    "critical": true,
                    "name": "keyUsage",
                    "value": "Digital Signature"
                },
                {
                    "critical": false,
                    "name": "extendedKeyUsage",
                    "value": "TLS Web Server Authentication,
                              TLS Web Client Authentication"
                }
            ]
        }
    ]
}
```

Výpis E.1: Struktura TLS certifikátu

Výpis E znázorňuje strukturu TLS certifikátu získaného během aktivního skenování domény. Obsahuje základní vlastnosti, jako jsou použitý šifrovací algoritmus, doba platnosti nebo počet rozšíření. Tyto surové atributy představují vstupní datový základ, ze kterého jsou následně odvozeny pokročilé příznaky – například entropie textových polí, poměry mezi certifikáty nebo metriky anomálií. Díky této struktuře je možné extrahovat reprezentaci popisující chování certifikátu na vyšší úrovni.

### Zpracování TLS řetězce certifikátů

Jednotlivé certifikáty v TLS řetězci jsou iterativně zpracovávány a jsou k nim doplňovány příznaky zaměřující se na:

1. **Extrakce základních rysů** – délka platnosti, počet rozšíření, identifikace autority.
2. **Výpočet metrik** – entropie názvů, poměry mezi kořenovým a listovým certifikátem.
3. **Detekce anomálií** – samo-podepsané certifikáty s více články v řetězci, řetězce bez validity apod.

### Nově vytvořené TLS příznaky

Na základě výše popsaného zpracování byla navržena rozšířená sada deseti nových TLS příznaků. Jejich přehled a přínos je shrnut v tabulce E.2.

| **Příznak** | **Popis** |
|---|---|
| `tls_cert_validity_ratio` | Poměr mezi platností kořenového a listového certifikátu. |
| `tls_cert_validity_diff` | Rozdíl v délce platnosti mezi certifikáty. |
| `tls_has_broken_or_expired_chain` | Označuje, zda je certifikační řetězec neplatný. |
| `tls_is_self_signed_and_has_chain` | Identifikuje anomálie u samo-podepsaných certifikátů. |
| `tls_policies_total_count` | Celkový počet politik v řetězci. |
| `tls_auth_cert_ratio` | Poměr mezi certifikáty pro server a klienta. |
| `tls_root_leaf_hash_match` | Shoda hashů mezi kořenovým a listovým certifikátem. |
| `tls_chain_cert_len_combined` | Kombinace délky řetězce a platnosti certifikátů. |
| `tls_cipher_entropy` | Entropie identifikátorů cipherů. |
| `tls_version_entropy` | Entropie verzí TLS protokolu. |

Tabulka E.2: Nově vytvořené TLS příznaky a jejich přínos.

Navržené příznaky byly zvoleny s ohledem na jejich schopnost popsat netriviální vlastnosti TLS certifikátového řetězce, které nejsou přímo zachyceny běžnými statickými atributy. Například poměr a rozdíl délek platnosti kořenového `validity_ratio` a listového certifikátu `validity_diff` mohou indikovat nestandardní nebo synteticky vytvořené řetězce. Metriky jako `entropy`, `hash_match` nebo `chain_length_combined` zachycují jemné odchylky v implementaci certifikátů, které se často vyskytují u phishingových nebo automaticky generovaných domén.

## Architektura neuronové sítě založené na TLS příznacích

Po návrhu a implementaci rozšířené sady TLS příznaků byl navržen klasifikační model, který tyto atributy zpracovává samostatně, bez využití dalších datových zdrojů (např. DNS nebo WHOIS). Model níže slouží jako experimentální klasifikátor využívající **pouze 24 TLS atributů**. Místo plného mechanismu *attention* používá jednodimenzionální *feature-wise gating*. Tedy malá sigmoid maska zvýrazňující relevantní rysy a potlačující šum ještě před hlubší projekcí.

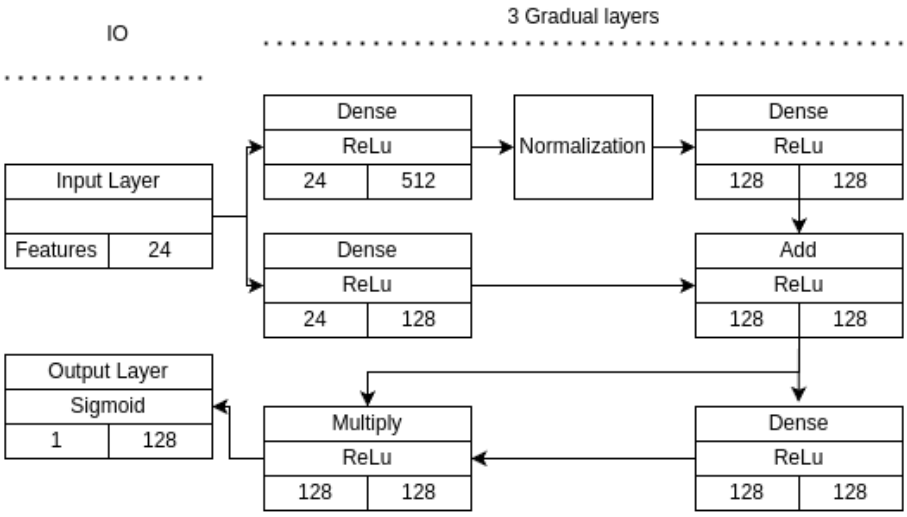
```
class TLSClassifier(nn.Module):
    def __init__(self, in_dim=24):
        super().__init__()
        self.gate = nn.Sequential(    # 1. Sigmoidová maska (feature gate)
            nn.Linear(in_dim, in_dim),
            nn.Sigmoid()
        )
        self.fc_in = nn.Sequential(   # 2. Vstupní projekce → 512 prvků
            nn.Linear(in_dim, 512),
            nn.BatchNorm1d(512),
            nn.ReLU()
        )

        self.fc1 = nn.Sequential(     # 3. Skrytý blok (512 -> 256 neuronů)
            nn.Linear(512, 256),
            nn.BatchNorm1d(256),
            nn.ReLU(),
            nn.Dropout(0.3)
        )
        self.fc2 = nn.Sequential(     # 3. Skrytý blok (256 -> 125 neuronů)
            nn.Linear(256, 128),
            nn.BatchNorm1d(128),
            nn.ReLU(),
            nn.Dropout(0.3)
        )

        self.skip = nn.Linear(in, 128)# 4) Reziduální větev vstupu
        self.out  = nn.Linear(128, 1) # 5) Pravděpodobnost malignity
```

- **Vstup (24 prvků)** – délka platnosti, typ validace, entropie CN aj. (normalizováno na $\langle 0, 1 \rangle$).
- **Gating** – maska $w \in (0, 1)^{24}$ z vrstvy Dense + sigmoid; vstup se po prvcích násobí $x \odot w$.
- **Projekční bloky** – Dense 512 $\rightarrow$ 256 $\rightarrow$ 128, vždy s `BatchNorm`, `ReLU`; po dvou blocích `Dropout 0.3`.
- **Reziduální větev** – vstup směrován do 128 neuronů a přičten (`skip connection`) pro lepší tok gradientu.
- **Výstup** – jeden neuron se `sigmoid` vrací pravděpodobnost, že doména je maligní.

Plný schématický náhled je na obr. E.1; textová podoba architektury je uvedena výše v blokovém výpisu.




Obrázek E.1: Architektura specializovaného TLS klasifikátoru s feature-wise gatingem.

## Výsledky klasifikace

Model `attention_tls` byl testován ve třetí fázi klasifikace samostatně pro phishingové a malware domény. Na validační datové sadě dosahoval velmi vysoké výkonnosti – přesnost, úplnost i F1 skóre se pohybovaly nad 95 %, jak shrnuje tabulka E.3.

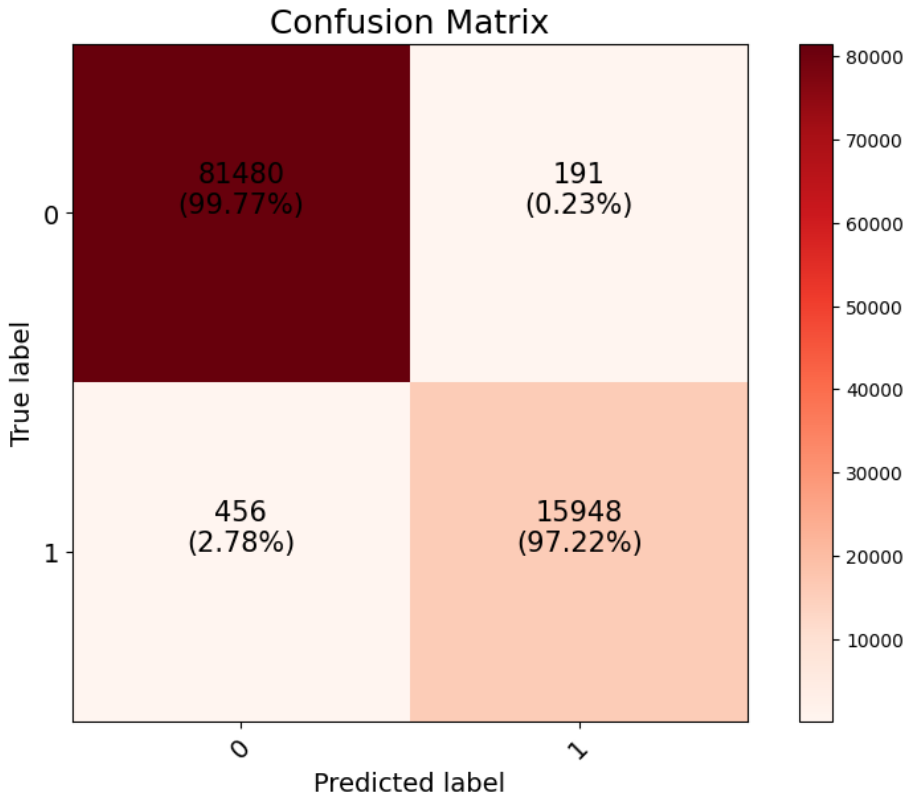
| **Metrika** | **Phishing** | **Malware** |
|---|---|---|
| Přesnost klasifikace (Accuracy) | 0,9934 ± 1,4e-04 | 0,9896 ± 7,8e-05 |
| Přesnost pozitivní třídy (Precision) | 0,9882 ± 8,8e-05 | 0,9660 ± 7,2e-05 |
| Úplnost (Recall) | 0,9722 ± 5,2e-05 | 0,9383 ± 5,7e-05 |
| F1 skóre | 0,9801 ± 6,4e-05 | 0,9519 ± 4,8e-05 |
| ROC AUC | 0,9849 ± 7,9e-05 | 0,9671 ± 6,1e-05 |

Tabulka E.3: Výsledky klasifikace modelu `tls` pro phishing a malware domény (10 běhů)

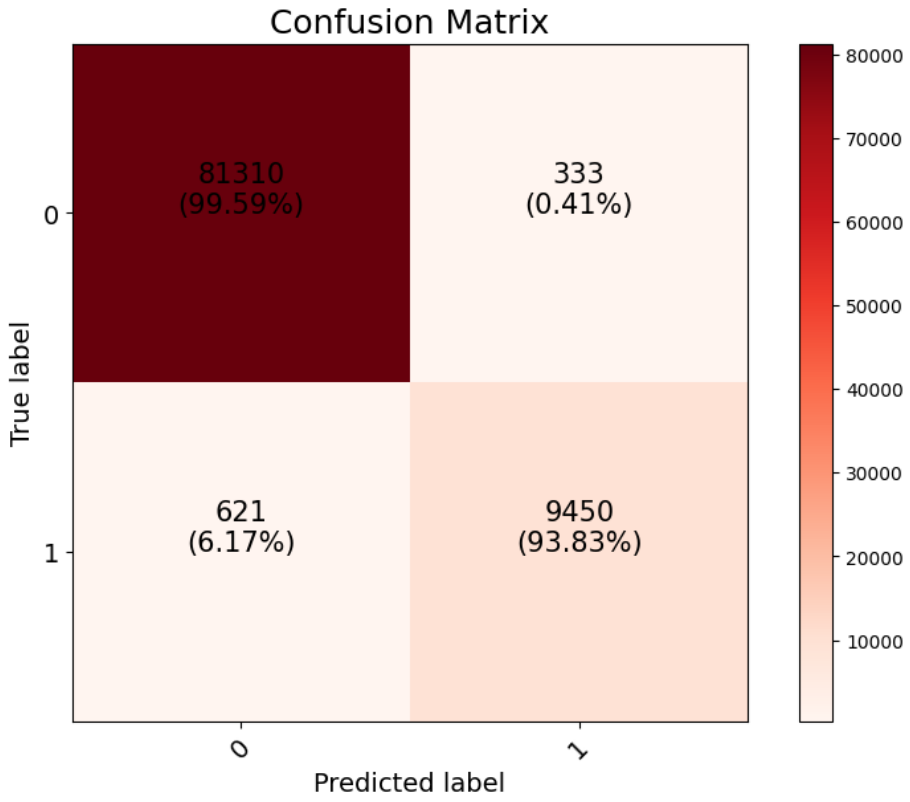
Z těchto výsledků vyplývá, že samotné TLS příznaky poskytují dostatečně bohatou informační hodnotu pro účinnou detekci škodlivých domén. Model vykazoval velmi vysoké hodnoty přesnosti, F1 skóre i AUC pro obě klasifikační úlohy a jeví se jako vhodný například pro nasazení v prostředích s omezeným přístupem k DNS nebo aplikačním datům. Výsledky byly dále ověřeny na oddělené verifikační datové sadě, viz sekce E.

## Analýza matic záměn

Kvalitu klasifikace potvrzuje i rozložení záměn zobrazené na obrázcích E.2 a E.3, kde je patrné minimum falešných pozitivních i negativních detekcí.




Obrázek E.2: Matice záměn pro phishingové domény (fáze 3).




Obrázek E.3: Matice záměn pro malware domény (fáze 3).

## Výsledky na verifikační datové sadě

Přestože model `attention_tls` dosáhl velmi dobrých výsledků na validační datové sadě (viz Tabulka E.3), jeho výkon na verifikačním vzorku byl výrazně slabší (viz Tabulka E.4) , zejména v případě detekce malware domén. Zatímco recall zůstal vysoký, přesnost (precision) se u obou úloh propadla, což naznačuje zvýšený počet falešně pozitivních klasifikací.

Z důvodu této slabší generalizace nebyl model založený výhradně na TLS příznacích zapojen do výsledné klasifikační pipeline. Přesto však považujeme tento přístup za zajímavý směr dalšího výzkumu – zejména s ohledem na nízké nároky na vstupní data a možnost jeho nasazení v prostředích s omezenými možnostmi hlubší inspekce.

| **Metrika** | **Malware** | **Phishing** |
|---|---|---|
| Přesnost klasifikace (Accuracy) | `0.8750 ± 6.2e-05` | `0.9278 ± 4.8e-05` |
| Přesnost pozitivní třídy (Precision) | `0.5814 ± 1.1e-04` | `0.7072 ± 9.1e-05` |
| Úplnost (Recall) | `0.8117 ± 9.0e-05` | `0.9481 ± 9.3e-05` |
| F1 skóre | `0.6767 ± 5.8e-05` | `0.8101 ± 4.1e-05` |
| ROC AUC | `0.8573 ± 7.2e-05` | `0.9360 ± 7.0e-05` |

Tabulka E.4: Srovnání metrik modelu `attention_tls` (Stage 3) pro malware a phishing (10 běhů).

## Závěr

Specializovaný model `attention_tls` ukázal, že je možné klasifikovat domény pouze na základě TLS metadat, bez nutnosti využití DNS, WHOIS nebo aplikačních příznaků. Při validaci dosáhl velmi dobrých výsledků a potvrdil, že TLS příznaky představují zajímavý, byť v praxi dosud málo využívaný, zdroj informací pro detekci škodlivých domén.

Při testování na verifikační datové sadě se však ukázalo, že model nedosahuje stejné úrovně generalizace. Zatímco úplnost zůstala vysoká, přesnost poklesla, zejména v případě malware domén, což vedlo ke zvýšenému výskytu falešně pozitivních klasifikací. Vzhledem k těmto výsledkům nebyl model `attention_tls` zařazen do finální klasifikační pipeline.

Přesto zůstává přístup založený na TLS a jeho samostatná klasifikace relevantní a slibnou oblastí pro další výzkum – zejména v kontextu pasivního monitoringu, analýzy šifrovaného provozu a nasazení v edge prostředích. Dále by bylo vhodné zkoumat možnosti rozšíření sady příznaků, robustnější trénovací přístupy a metody kombinace s jinými modalitami pro zvýšení odolnosti vůči rozdílům v distribuci dat mezi sadami.

# Příloha F

# Výsledky analýzy SHAP

Analýza Shapleyho hodnot (SHAP) poskytuje hlubší pohled na to, jak jednotlivé příznaky přispívají k rozhodování konkrétních modelů. Následující grafy znázorňují distribuci hodnot SHAP pro každý model zvlášť.

Model FFNN klade největší důraz na příznaky z oblasti RDAP `rdap_domain_age`) a lexikálních znaků domény (např. `lex_tld_abuse_score`). Dále je patrný vliv vybraných TLS atributů, i když jejich dopad je ve srovnání s ostatními kategoriemi menší.

U modelu LightGBM dominují atributy z RDAP oblasti a DNS záznamy, přičemž příznak `rdap_ip_v4_count` patří mezi nejvýznamnější. Rovněž se zde více uplatňuje informační entropie z IP a DNS zón.

Model XGBoost se opírá o podobné sady příznaků, nicméně více zvýrazňuje lexikální znaky druhé úrovně domény (např. `lex_sld_digit_count`) a specifické TLS vlastnosti jako `tls_CA_certs_in_chain_ratio`. Příznaky z RDAP oblasti zůstávají důležitým základem.

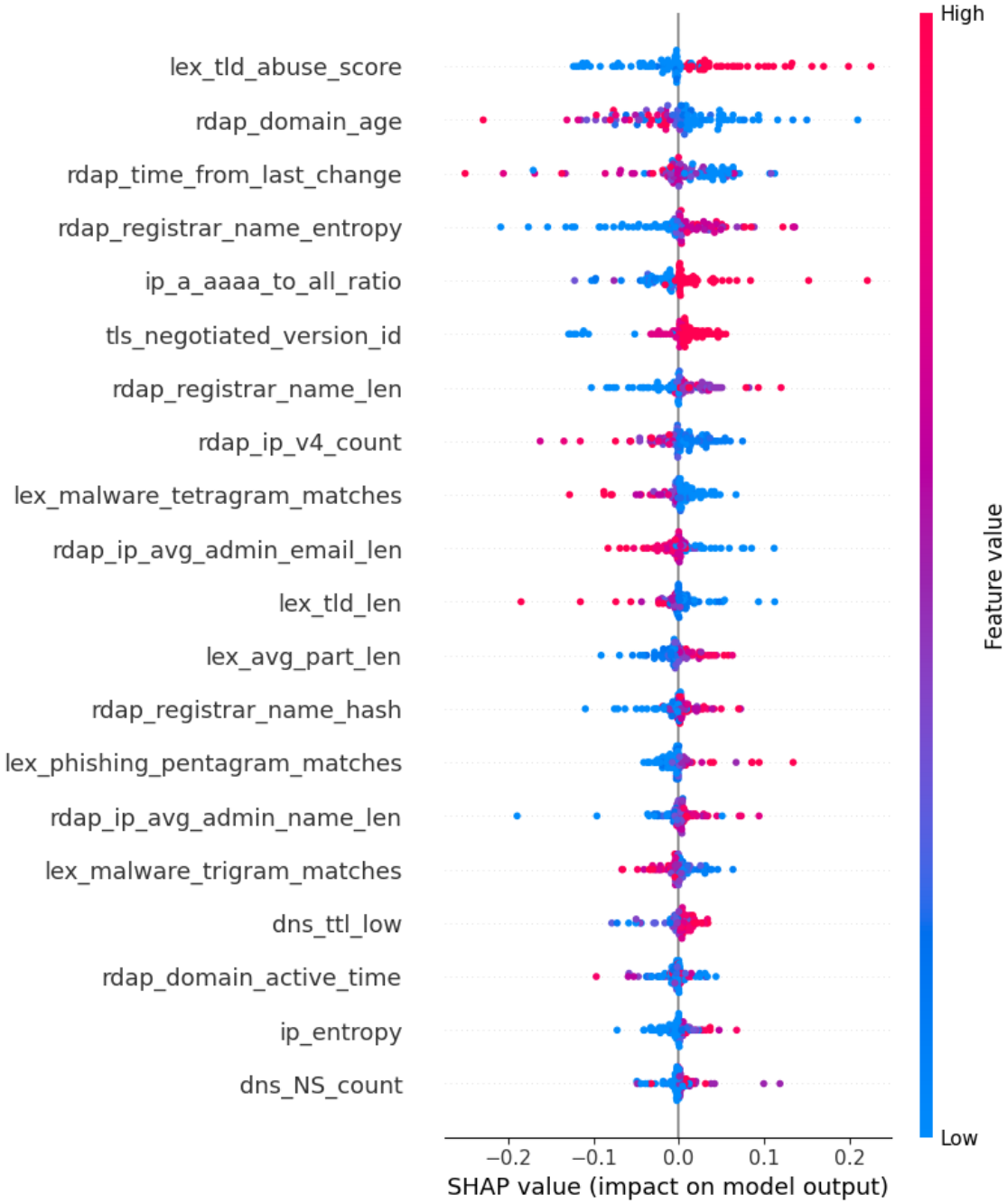
SVM model rovněž ukazuje silnou závislost na RDAP příznacích (věk domény, délka registrace), doplněnou o lexikální charakteristiky a síťové vlastnosti. Model reflektuje robustní schopnost klasifikace při kombinaci více typů příznaků.

## Přínos všech příznaků napříč modely

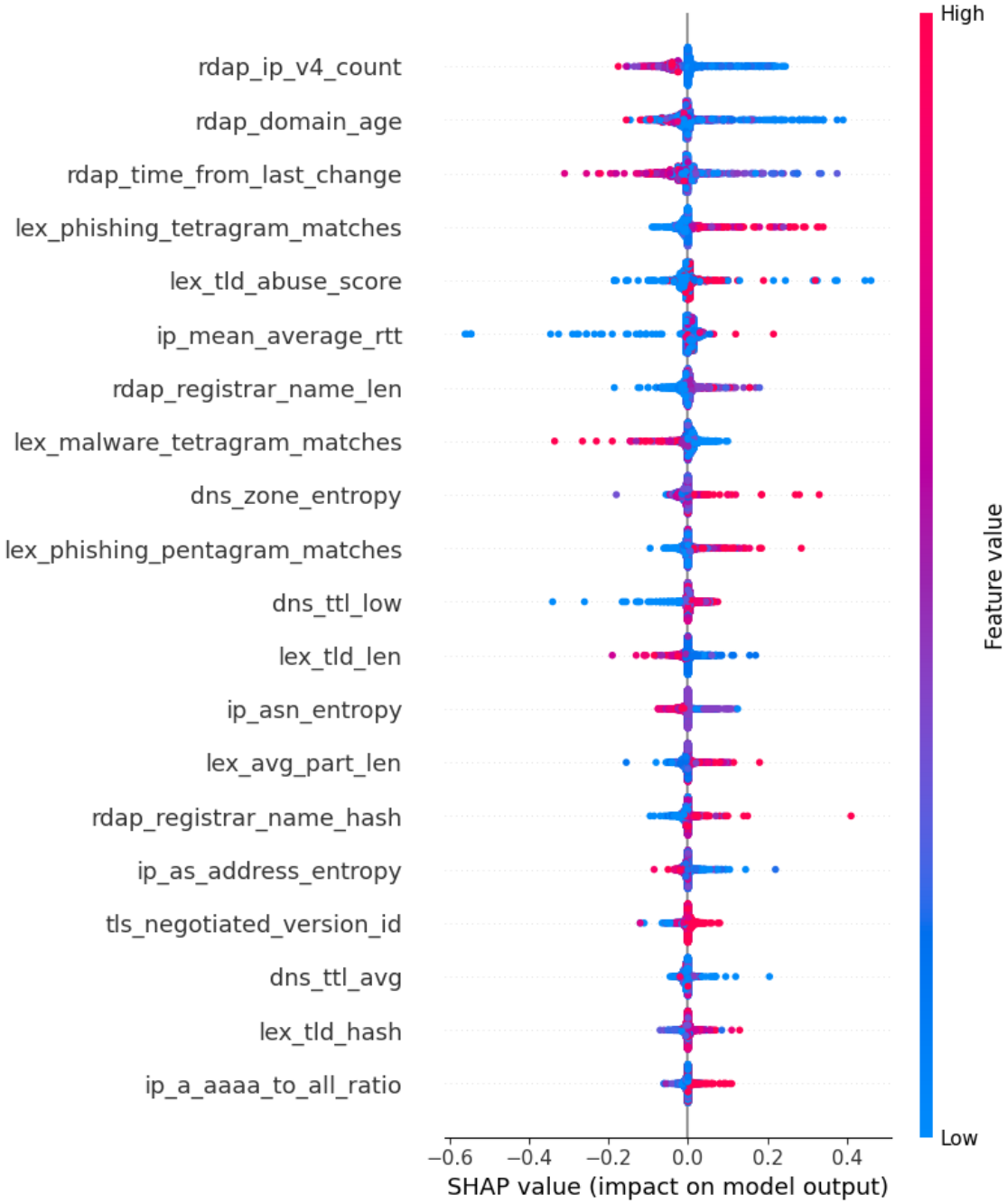
Kromě pohledu na jednotlivé modely byla provedena agregovaná analýza SHAP hodnot napříč celou klasifikační pipeline.

Na obrázku F.5 jsou jednotlivé příznaky seřazeny dle jejich průměrného přínosu k rozhodování. Dominují především příznaky z RDAP oblasti, následované IP a lexikálními znaky. Barevné označení umožňuje sledovat, které kategorie přispívají nejvíce, a zároveň ukazuje značný pokles důležitosti u DNS, GEO a TLS příznaků.

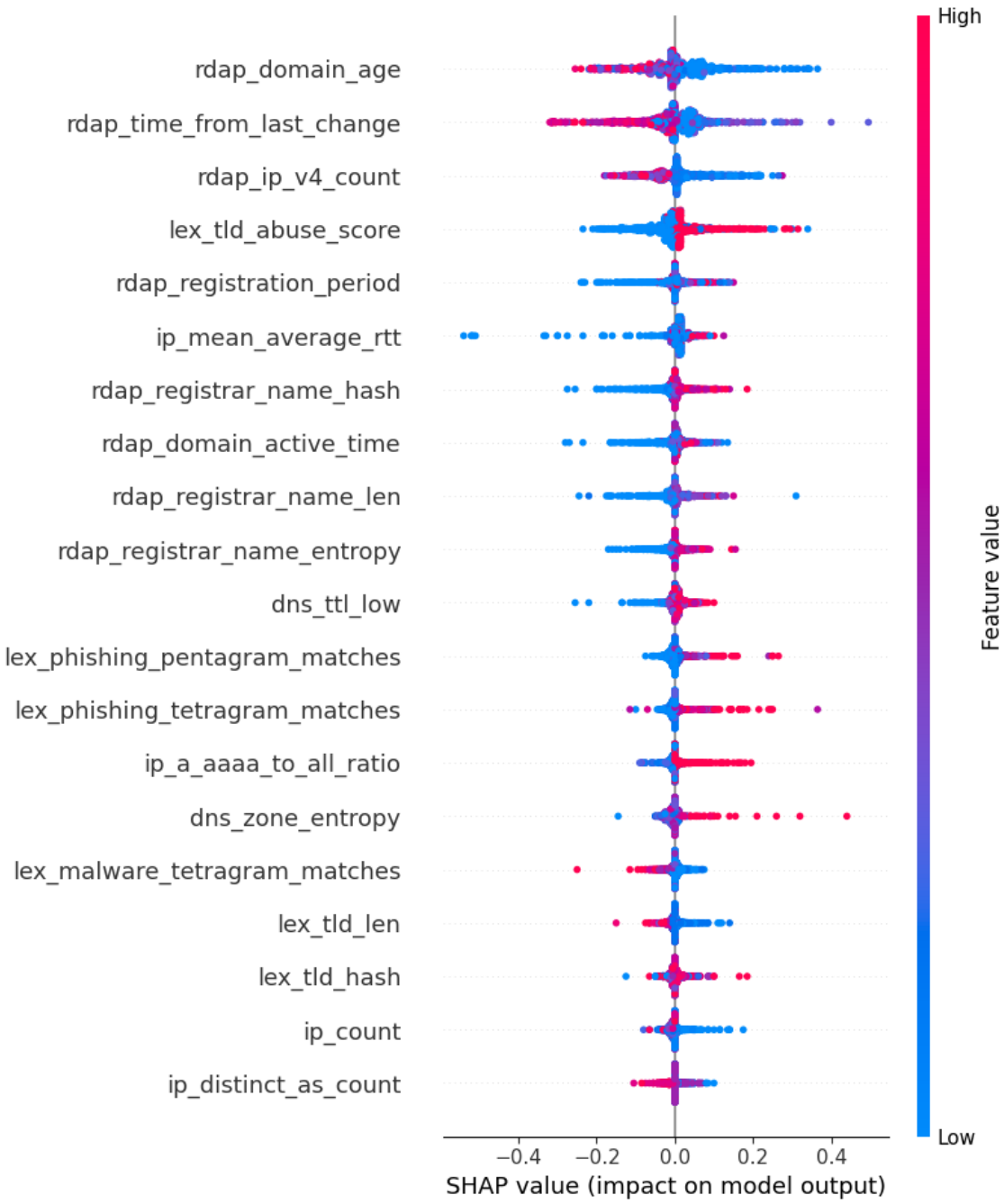
Konečný souhrn na obrázku F.6 kvantifikuje průměrný přínos jednotlivých kategorií. Nejvyšší přínos vykazují RDAP příznaky, následované IP a lexikálními znaky domény. Naopak TLS a GEO atributy měly relativně nízký vliv, což naznačuje jejich omezenou roli v celkové klasifikaci. Tyto poznatky mohou být vodítkem pro budoucí redukci dimenze nebo návrh specializovaných klasifikátorů pro jednotlivé kategorie.



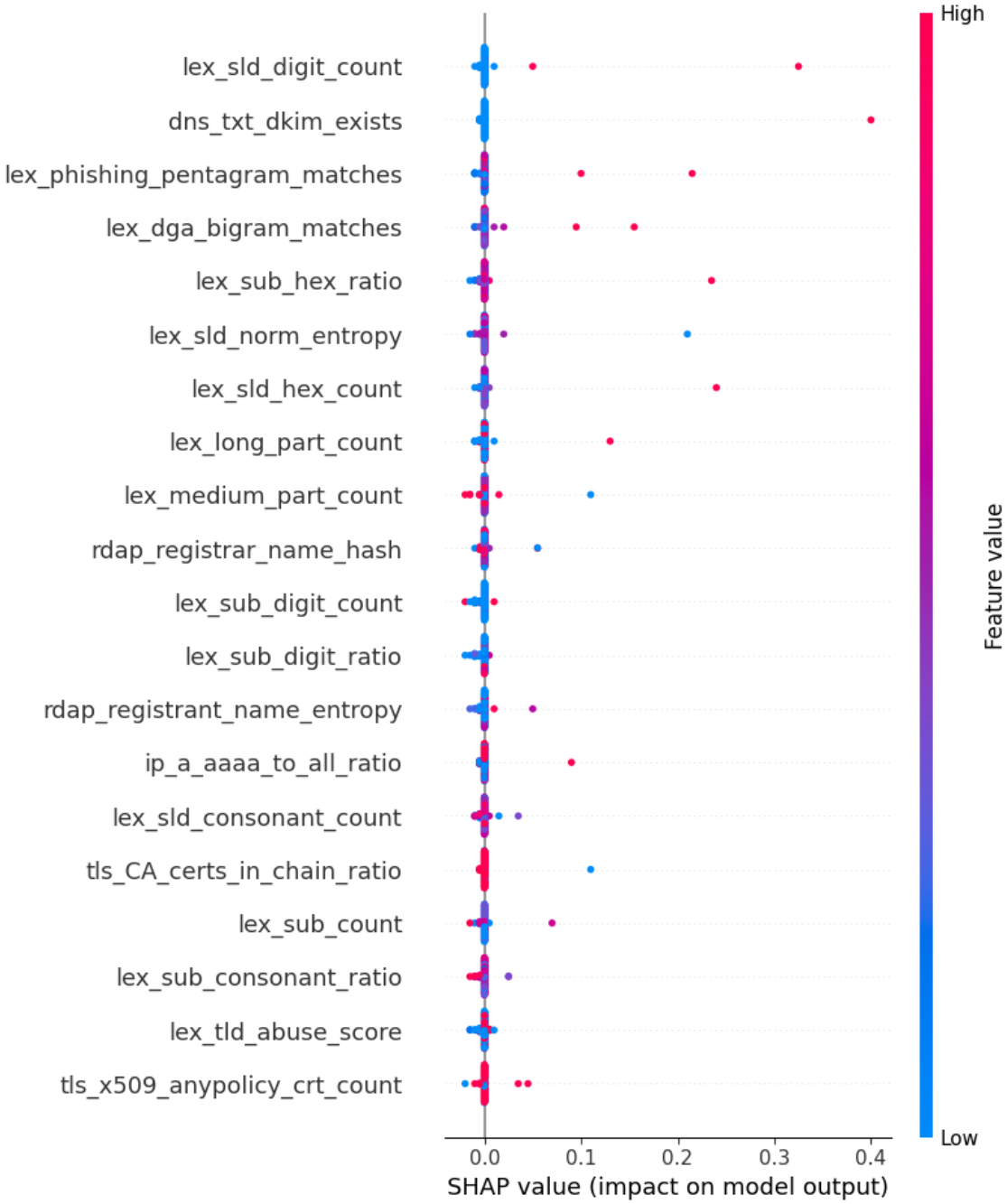

Obrázek F.1: Přínos jednotlivých příznaků pro model FFNN



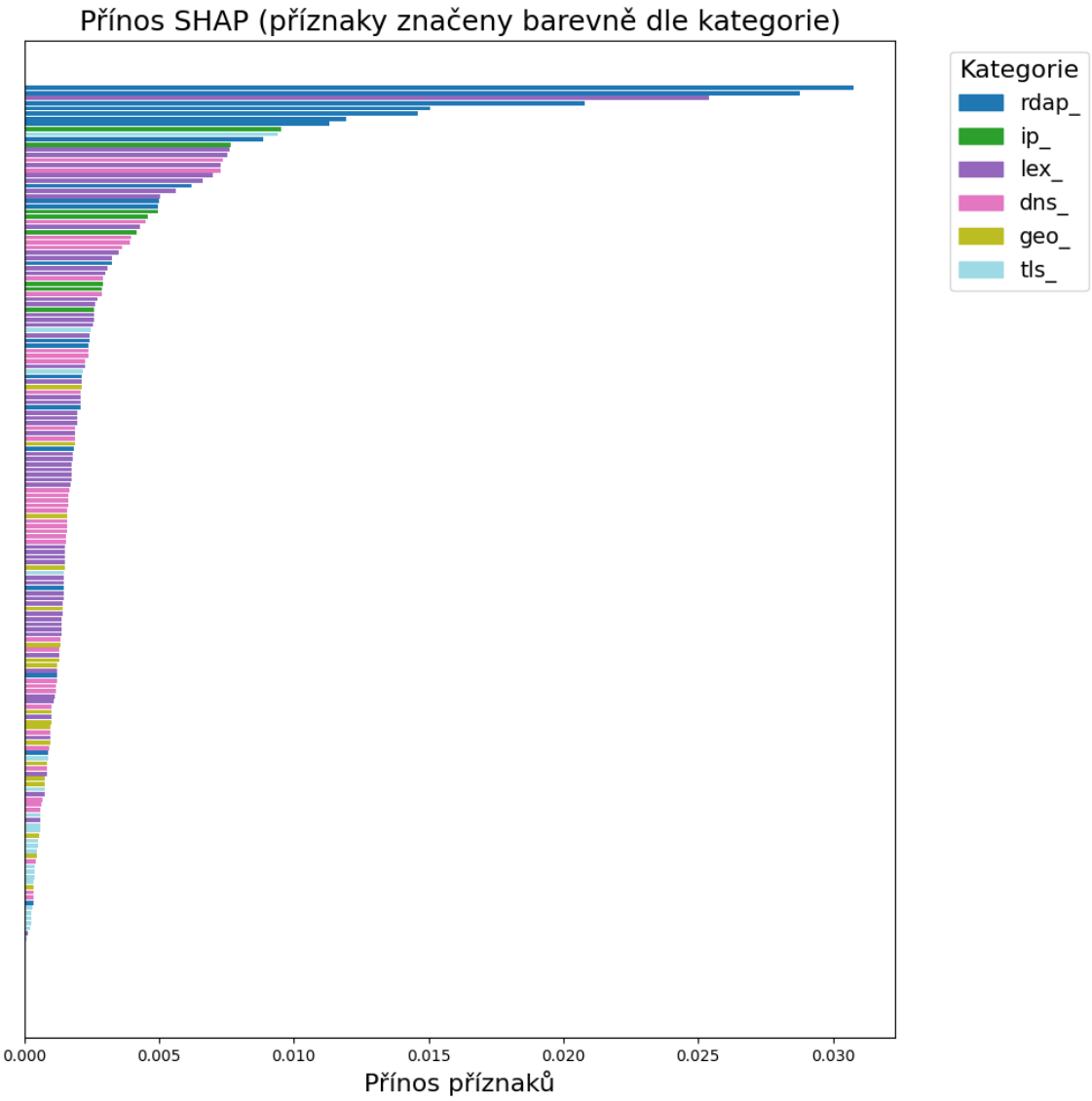

Obrázek F.2: Přínos jednotlivých příznaků pro model LightGBM



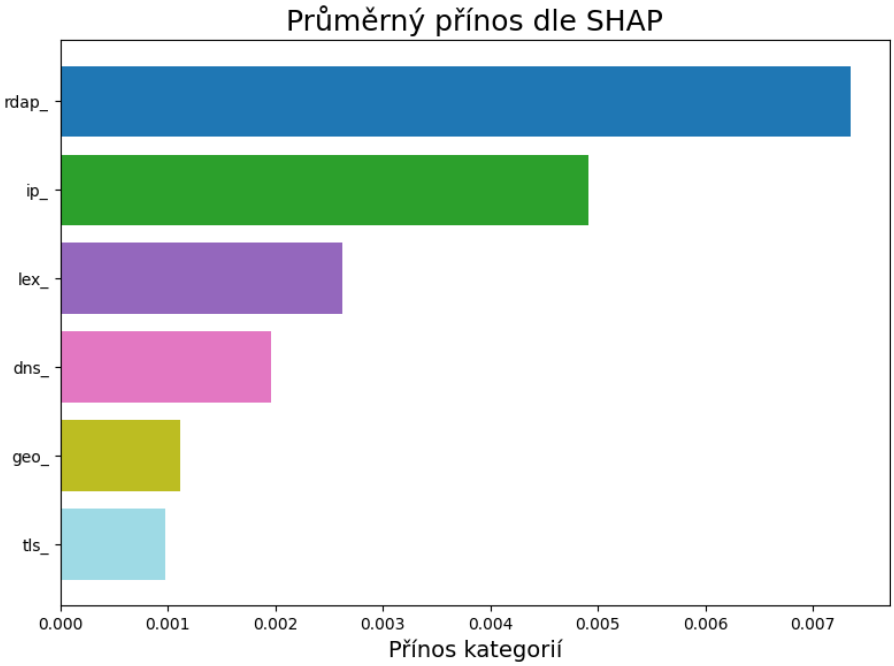

Obrázek F.3: Přínos jednotlivých příznaků pro model XGBoost




Obrázek F.4: Přínos jednotlivých příznaků pro model SVM




Obrázek F.5: Průměrný SHAP přínos všech příznaků, barevně rozlišen dle kategorií




Obrázek F.6: Průměrný přínos příznaků dle kategorií

# Příloha G

# Měření klasifikace dle podmnožin

Tato příloha obsahuje kompletní výsledky všech měření samostatných i agregovaných subsetů příznaků pro klasifikaci phishingových a malware domén.

Každý výstup zobrazuje přesnost jednotlivých klasifikačních algoritmů na dané kombinaci příznaků, zvlášť pro phishing a zvlášť pro malware. Měření byla provedena podle metodologie popsané v kapitole **Předběžná analýza podmnožin příznaků**, kde byl popsán postup automatizovaného trénování modelů pomocí knihovny `PyCaret`. Cílem bylo identifikovat optimální kombinace atributů a klasifikátorů pro detekci škodlivých domén.

### Použité klasifikátory

Následující tabulka obsahuje zkratky používané v jednotlivých výstupech a jejich odpovídající plné názvy modelů. Tyto zkratky byly zvoleny pro zajištění přehlednosti a úsporu místa ve výstupech.

Tabulka G.1: Zkratky použitých klasifikátorů

| Zkratka | Plný název | Poznámka |
|---|---|---|
| `RF` | Random Forest Classifier | Ensemble |
| `ADA` | Ada Boost Classifier | Ensemble |
| `DT` | Decision Tree Classifier | Interpretable |
| `Ridge` | Ridge Classifier | Linear model |
| `KNN` | K Neighbors Classifier | Instance-based |
| `SVM-L` | SVM - Linear Kernel | Linear SVM |
| `LR` | Logistic Regression | Linear model |
| `QDA` | Quadratic Discriminant Analysis | Probabilistic |
| `NB` | Naive Bayes | Probabilistic |
| `ET` | Extra Trees Classifier | Ensemble |
| `XGB` | Extreme Gradient Boosting | Boosted Trees |
| `LGBM` | Light Gradient Boosting Machine | Boosted Trees |
| `GBC` | Gradient Boosting Classifier | Boosted Trees |
| `LDA` | Linear Discriminant Analysis | Probabilistic |
| `Dummy` | Dummy Classifier | Referenční baseline |

### Komentář k výstupům

Každá tabulka představuje výsledky pro konkrétní subset příznaků a typ útoku (phishing nebo malware). Metody jsou porovnávány dle hlavních metrik klasifikace: přesnost (**Acc**), plocha pod ROC křivkou (**AUC**), **Recall**, **Precision**, **F1 skóre**, Cohenova **Kappa** a **Matthews Correlation Coefficient** (MCC).

Díky oddělení výsledků pro phishing a malware je možné detailně porovnat účinnost jednotlivých přístupů pro různé typy útoků a rozhodnout o vhodné strategii nasazení.

Tabulka G.2: Výsledky pro subset dns – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.9165 | 0.9120 | 0.9165 | 0.9135 | 0.9143 | 0.6861 | 0.6886 |
| ET | 0.9155 | 0.8910 | 0.9155 | 0.9125 | 0.9133 | 0.6829 | 0.6851 |
| KNN | 0.9142 | 0.8860 | 0.9142 | 0.9109 | 0.9116 | 0.6757 | 0.6787 |
| XGB | 0.9166 | 0.9175 | 0.9166 | 0.9129 | 0.9115 | 0.6681 | 0.6785 |
| LGBM | 0.9174 | 0.9176 | 0.9174 | 0.9144 | 0.9113 | 0.6647 | 0.6794 |
| DT | 0.8958 | 0.8503 | 0.8958 | 0.8957 | 0.8957 | 0.6268 | 0.6271 |
| GBC | 0.8911 | 0.8799 | 0.8911 | 0.8921 | 0.8728 | 0.4991 | 0.5538 |
| ADA | 0.8632 | 0.8417 | 0.8632 | 0.8536 | 0.8335 | 0.3352 | 0.3997 |
| LDA | 0.8491 | 0.7843 | 0.8491 | 0.8264 | 0.8136 | 0.2532 | 0.3100 |
| Ridge | 0.8346 | 0.7842 | 0.8346 | 0.7962 | 0.7724 | 0.0722 | 0.1401 |
| LR | 0.8242 | 0.6988 | 0.8242 | 0.7307 | 0.7582 | 0.0143 | 0.0262 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| SVM-L | 0.6466 | 0.6454 | 0.6466 | 0.7720 | 0.6756 | 0.1180 | 0.1475 |
| QDA | 0.5847 | 0.7880 | 0.5847 | 0.8468 | 0.6332 | 0.2208 | 0.3161 |
| NB | 0.3432 | 0.6313 | 0.3432 | 0.8202 | 0.3546 | 0.0640 | 0.1513 |

Tabulka G.3: Výsledky pro subset geo – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.8461 | 0.8194 | 0.8461 | 0.8210 | 0.8106 | 0.2416 | 0.2948 |
| ET | 0.8461 | 0.8173 | 0.8461 | 0.8210 | 0.8102 | 0.2400 | 0.2939 |
| LGBM | 0.8482 | 0.8291 | 0.8482 | 0.8272 | 0.8098 | 0.2352 | 0.2998 |
| DT | 0.8446 | 0.8161 | 0.8446 | 0.8176 | 0.8092 | 0.2367 | 0.2870 |
| XGB | 0.8474 | 0.8277 | 0.8474 | 0.8249 | 0.8092 | 0.2333 | 0.2953 |
| KNN | 0.8380 | 0.7701 | 0.8380 | 0.8056 | 0.8031 | 0.2147 | 0.2555 |
| GBC | 0.8446 | 0.8170 | 0.8446 | 0.8218 | 0.8000 | 0.1915 | 0.2634 |
| ADA | 0.8342 | 0.7980 | 0.8342 | 0.7815 | 0.7674 | 0.0511 | 0.1055 |
| Ridge | 0.8317 | 0.6927 | 0.8317 | 0.6917 | 0.7553 | -0.0001 | -0.0010 |
| LDA | 0.8317 | 0.6928 | 0.8317 | 0.6917 | 0.7553 | -0.0001 | -0.0010 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| LR | 0.8311 | 0.7075 | 0.8311 | 0.7085 | 0.7551 | -0.0008 | -0.0044 |
| SVM-L | 0.7870 | 0.5542 | 0.7870 | 0.7089 | 0.7393 | 0.0064 | 0.0019 |
| QDA | 0.4123 | 0.7363 | 0.4123 | 0.8469 | 0.4425 | 0.1110 | 0.2242 |
| NB | 0.3427 | 0.6407 | 0.3427 | 0.7833 | 0.3620 | 0.0453 | 0.1011 |

Tabulka G.4: Výsledky pro subset html – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.8410 | 0.7060 | 0.8410 | 0.8313 | 0.7824 | 0.1133 | 0.2110 |
| LGBM | 0.8414 | 0.7098 | 0.8414 | 0.8357 | 0.7823 | 0.1127 | 0.2145 |
| XGB | 0.8411 | 0.7090 | 0.8411 | 0.8327 | 0.7822 | 0.1125 | 0.2115 |
| ET | 0.8403 | 0.7009 | 0.8403 | 0.8260 | 0.7820 | 0.1117 | 0.2046 |
| KNN | 0.8409 | 0.6828 | 0.8409 | 0.8346 | 0.7811 | 0.1075 | 0.2087 |
| GBC | 0.8416 | 0.7078 | 0.8416 | 0.8485 | 0.7804 | 0.1041 | 0.2174 |
| ADA | 0.8411 | 0.6954 | 0.8411 | 0.8476 | 0.7793 | 0.0995 | 0.2111 |
| DT | 0.8285 | 0.6875 | 0.8285 | 0.7733 | 0.7745 | 0.0854 | 0.1272 |
| LR | 0.8313 | 0.6499 | 0.8313 | 0.7142 | 0.7553 | -0.0000 | 0.0011 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| Ridge | 0.8315 | 0.6633 | 0.8315 | 0.6917 | 0.7552 | -0.0004 | -0.0034 |
| LDA | 0.8314 | 0.6617 | 0.8314 | 0.6917 | 0.7551 | -0.0007 | -0.0063 |
| SVM-L | 0.7869 | 0.6562 | 0.7869 | 0.7278 | 0.7350 | 0.0172 | 0.0288 |
| NB | 0.4640 | 0.6623 | 0.4640 | 0.8337 | 0.5074 | 0.1309 | 0.2305 |
| QDA | 0.4459 | 0.6664 | 0.4459 | 0.8477 | 0.4837 | 0.1304 | 0.2437 |

Tabulka G.5: Výsledky pro subset ip – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.8669 | 0.8812 | 0.8669 | 0.8524 | 0.8511 | 0.4242 | 0.4478 |
| LGBM | 0.8669 | 0.8817 | 0.8669 | 0.8524 | 0.8490 | 0.4124 | 0.4414 |
| KNN | 0.8553 | 0.8210 | 0.8553 | 0.8452 | 0.8469 | 0.4289 | 0.4379 |
| RF | 0.8437 | 0.8314 | 0.8437 | 0.8312 | 0.8359 | 0.3888 | 0.3930 |
| ET | 0.8425 | 0.7755 | 0.8425 | 0.8296 | 0.8343 | 0.3824 | 0.3868 |
| GBC | 0.8589 | 0.8582 | 0.8589 | 0.8421 | 0.8326 | 0.3372 | 0.3833 |
| DT | 0.8406 | 0.7284 | 0.8406 | 0.8273 | 0.8322 | 0.3740 | 0.3786 |
| ADA | 0.8404 | 0.8359 | 0.8404 | 0.8126 | 0.7911 | 0.1533 | 0.2249 |
| LDA | 0.8317 | 0.7536 | 0.8317 | 0.7283 | 0.7556 | 0.0015 | 0.0107 |
| Ridge | 0.8317 | 0.7537 | 0.8317 | 0.6917 | 0.7553 | -0.0001 | -0.0010 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| LR | 0.8306 | 0.7597 | 0.8306 | 0.7085 | 0.7551 | -0.0007 | -0.0047 |
| SVM-L | 0.7730 | 0.6588 | 0.7730 | 0.7781 | 0.7467 | 0.1158 | 0.1512 |
| QDA | 0.5441 | 0.7503 | 0.5441 | 0.8516 | 0.5921 | 0.1961 | 0.3035 |
| NB | 0.5152 | 0.6785 | 0.5152 | 0.8361 | 0.5638 | 0.1630 | 0.2599 |

Tabulka G.6: Výsledky pro subset lex – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9371 | 0.9352 | 0.9371 | 0.9358 | 0.9338 | 0.7531 | 0.7618 |
| XGB | 0.9341 | 0.9303 | 0.9341 | 0.9321 | 0.9311 | 0.7441 | 0.7507 |
| GBC | 0.9255 | 0.9204 | 0.9255 | 0.9248 | 0.9194 | 0.6940 | 0.7125 |
| RF | 0.9238 | 0.9144 | 0.9238 | 0.9215 | 0.9188 | 0.6944 | 0.7068 |
| ET | 0.9213 | 0.8860 | 0.9213 | 0.9187 | 0.9158 | 0.6824 | 0.6959 |
| ADA | 0.9066 | 0.8994 | 0.9066 | 0.9018 | 0.8989 | 0.6161 | 0.6325 |
| LDA | 0.8987 | 0.8850 | 0.8987 | 0.8952 | 0.8864 | 0.5604 | 0.5920 |
| DT | 0.8761 | 0.7909 | 0.8761 | 0.8780 | 0.8769 | 0.5637 | 0.5639 |
| Ridge | 0.8929 | 0.8852 | 0.8929 | 0.8945 | 0.8751 | 0.5083 | 0.5627 |
| KNN | 0.8635 | 0.7860 | 0.8635 | 0.8479 | 0.8475 | 0.4106 | 0.4329 |
| LR | 0.8317 | 0.4768 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| NB | 0.8317 | 0.5302 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| SVM-L | 0.8317 | 0.4927 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| QDA | 0.3245 | 0.8671 | 0.3245 | 0.8487 | 0.3192 | 0.0676 | 0.1740 |

Tabulka G.7: Výsledky pro subset rdap – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9533 | 0.9590 | 0.9533 | 0.9524 | 0.9518 | 0.8228 | 0.8265 |
| XGB | 0.9522 | 0.9589 | 0.9522 | 0.9512 | 0.9509 | 0.8204 | 0.8230 |
| RF | 0.9502 | 0.9572 | 0.9502 | 0.9491 | 0.9487 | 0.8121 | 0.8151 |
| ET | 0.9467 | 0.9410 | 0.9467 | 0.9455 | 0.9450 | 0.7980 | 0.8016 |
| GBC | 0.9358 | 0.9420 | 0.9358 | 0.9342 | 0.9326 | 0.7489 | 0.7568 |
| DT | 0.9319 | 0.8981 | 0.9319 | 0.9313 | 0.9315 | 0.7541 | 0.7543 |
| KNN | 0.9296 | 0.8997 | 0.9296 | 0.9275 | 0.9281 | 0.7381 | 0.7396 |
| ADA | 0.9196 | 0.9229 | 0.9196 | 0.9160 | 0.9159 | 0.6877 | 0.6938 |
| LDA | 0.8576 | 0.8538 | 0.8576 | 0.8402 | 0.8421 | 0.3919 | 0.4102 |
| Ridge | 0.8476 | 0.8539 | 0.8476 | 0.8271 | 0.8067 | 0.2207 | 0.2905 |
| NB | 0.8317 | 0.7152 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| LR | 0.6944 | 0.4346 | 0.6944 | 0.6868 | 0.6857 | -0.0985 | -0.1081 |
| QDA | 0.6038 | 0.8827 | 0.6038 | 0.8596 | 0.6506 | 0.2492 | 0.3524 |
| SVM-L | 0.4393 | 0.4665 | 0.4393 | 0.6703 | 0.4658 | -0.1430 | -0.1609 |

Tabulka G.8: Výsledky pro subset tls – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| DT | 0.8370 | 0.6894 | 0.8370 | 0.8323 | 0.7706 | 0.0633 | 0.1591 |
| RF | 0.8373 | 0.6889 | 0.8373 | 0.8393 | 0.7705 | 0.0629 | 0.1633 |
| ET | 0.8371 | 0.6887 | 0.8371 | 0.8362 | 0.7705 | 0.0629 | 0.1612 |
| XGB | 0.8370 | 0.6919 | 0.8370 | 0.8360 | 0.7703 | 0.0620 | 0.1600 |
| LGBM | 0.8369 | 0.6885 | 0.8369 | 0.8346 | 0.7701 | 0.0613 | 0.1580 |
| GBC | 0.8366 | 0.6848 | 0.8366 | 0.8371 | 0.7689 | 0.0562 | 0.1531 |
| ADA | 0.8356 | 0.6326 | 0.8356 | 0.8327 | 0.7670 | 0.0482 | 0.1385 |
| Ridge | 0.8349 | 0.6280 | 0.8349 | 0.8172 | 0.7667 | 0.0473 | 0.1275 |
| LDA | 0.8330 | 0.6279 | 0.8330 | 0.7916 | 0.7665 | 0.0469 | 0.1093 |
| KNN | 0.8342 | 0.5444 | 0.8342 | 0.8192 | 0.7641 | 0.0364 | 0.1129 |
| NB | 0.8280 | 0.5668 | 0.8280 | 0.7525 | 0.7608 | 0.0239 | 0.0522 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| LR | 0.7481 | 0.5816 | 0.7481 | 0.7399 | 0.7438 | 0.0706 | 0.0707 |
| SVM-L | 0.6982 | 0.5314 | 0.6982 | 0.7315 | 0.6935 | 0.0594 | 0.0544 |
| QDA | 0.6663 | 0.5838 | 0.6663 | 0.7567 | 0.6503 | 0.0769 | 0.0889 |

Tabulka G.9: Výsledky pro subset dns+rdap+tls+geo+ip+html+lex – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9975 | 0.9999 | 0.9975 | 0.9975 | 0.9975 | 0.9874 | 0.9874 |
| XGB | 0.9974 | 0.9999 | 0.9974 | 0.9974 | 0.9974 | 0.9868 | 0.9868 |
| GBC | 0.9951 | 0.9997 | 0.9951 | 0.9951 | 0.9951 | 0.9752 | 0.9752 |
| ADA | 0.9945 | 0.9996 | 0.9945 | 0.9945 | 0.9945 | 0.9719 | 0.9719 |
| RF | 0.9942 | 0.9996 | 0.9942 | 0.9942 | 0.9941 | 0.9697 | 0.9699 |
| DT | 0.9930 | 0.9829 | 0.9930 | 0.9930 | 0.9930 | 0.9643 | 0.9644 |
| ET | 0.9888 | 0.9985 | 0.9888 | 0.9888 | 0.9885 | 0.9402 | 0.9415 |
| LDA | 0.9817 | 0.9958 | 0.9817 | 0.9817 | 0.9817 | 0.9063 | 0.9063 |
| Ridge | 0.9774 | 0.9958 | 0.9774 | 0.9770 | 0.9767 | 0.8775 | 0.8801 |
| KNN | 0.9530 | 0.9313 | 0.9530 | 0.9512 | 0.9517 | 0.7471 | 0.7488 |
| LR | 0.9265 | 0.9632 | 0.9265 | 0.9228 | 0.9242 | 0.6016 | 0.6035 |
| Dummy | 0.8901 | 0.5000 | 0.8901 | 0.7923 | 0.8384 | 0.0000 | 0.0000 |
| SVM-L | 0.8355 | 0.6441 | 0.8355 | 0.8080 | 0.8110 | 0.0235 | 0.0326 |
| QDA | 0.7671 | 0.9591 | 0.7671 | 0.9177 | 0.8090 | 0.3690 | 0.4622 |
| NB | 0.6445 | 0.9416 | 0.6445 | 0.9136 | 0.7097 | 0.2464 | 0.3704 |

Tabulka G.10: Výsledky pro subset dns+rdap – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9897 | 0.9989 | 0.9897 | 0.9898 | 0.9897 | 0.9478 | 0.9478 |
| XGB | 0.9895 | 0.9989 | 0.9895 | 0.9895 | 0.9895 | 0.9465 | 0.9465 |
| RF | 0.9878 | 0.9973 | 0.9878 | 0.9877 | 0.9878 | 0.9370 | 0.9371 |
| ET | 0.9847 | 0.9933 | 0.9847 | 0.9845 | 0.9844 | 0.9192 | 0.9200 |
| DT | 0.9826 | 0.9564 | 0.9826 | 0.9827 | 0.9827 | 0.9114 | 0.9114 |
| GBC | 0.9825 | 0.9971 | 0.9825 | 0.9826 | 0.9826 | 0.9109 | 0.9109 |
| ADA | 0.9800 | 0.9965 | 0.9800 | 0.9803 | 0.9801 | 0.8989 | 0.8990 |
| LDA | 0.9519 | 0.9794 | 0.9519 | 0.9520 | 0.9519 | 0.7547 | 0.7547 |
| KNN | 0.9423 | 0.9239 | 0.9423 | 0.9404 | 0.9411 | 0.6933 | 0.6943 |
| Ridge | 0.9407 | 0.9794 | 0.9407 | 0.9388 | 0.9334 | 0.6265 | 0.6562 |
| QDA | 0.8580 | 0.9716 | 0.8580 | 0.9339 | 0.8793 | 0.5287 | 0.5916 |
| LR | 0.8942 | 0.9420 | 0.8942 | 0.8710 | 0.8747 | 0.2767 | 0.3070 |
| Dummy | 0.8901 | 0.5000 | 0.8901 | 0.7923 | 0.8384 | 0.0000 | 0.0000 |
| SVM-L | 0.8407 | 0.5971 | 0.8407 | 0.8208 | 0.8149 | 0.0353 | 0.0527 |
| NB | 0.6356 | 0.9308 | 0.6356 | 0.9131 | 0.7022 | 0.2386 | 0.3638 |

Tabulka G.11: Výsledky pro subset dns – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9193 | 0.8869 | 0.9193 | 0.9110 | 0.9080 | 0.4945 | 0.5247 |
| RF | 0.9156 | 0.8698 | 0.9156 | 0.9059 | 0.9067 | 0.4961 | 0.5136 |
| KNN | 0.9133 | 0.8388 | 0.9133 | 0.9040 | 0.9060 | 0.4989 | 0.5103 |
| LGBM | 0.9197 | 0.8913 | 0.9197 | 0.9133 | 0.9059 | 0.4744 | 0.5193 |
| ET | 0.9139 | 0.8390 | 0.9139 | 0.9042 | 0.9058 | 0.4952 | 0.5089 |
| GBC | 0.9084 | 0.8640 | 0.9084 | 0.8977 | 0.8888 | 0.3693 | 0.4255 |
| DT | 0.8881 | 0.7566 | 0.8881 | 0.8838 | 0.8858 | 0.4206 | 0.4212 |
| ADA | 0.9037 | 0.8368 | 0.9037 | 0.8898 | 0.8824 | 0.3311 | 0.3864 |
| LDA | 0.8934 | 0.7625 | 0.8934 | 0.8738 | 0.8779 | 0.3266 | 0.3484 |
| Ridge | 0.8855 | 0.7627 | 0.8855 | 0.8305 | 0.8355 | 0.0159 | 0.0491 |
| Dummy | 0.8867 | 0.5000 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| LR | 0.8860 | 0.6164 | 0.8860 | 0.7951 | 0.8333 | -0.0003 | -0.0020 |
| SVM-L | 0.5407 | 0.4806 | 0.5407 | 0.8025 | 0.6217 | 0.0110 | 0.0139 |
| QDA | 0.4013 | 0.7845 | 0.4013 | 0.8798 | 0.4701 | 0.0807 | 0.1799 |
| NB | 0.1294 | 0.5148 | 0.1294 | 0.7511 | 0.0624 | -0.0038 | -0.0298 |

Tabulka G.12: Výsledky pro subset geo – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.8933 | 0.8167 | 0.8933 | 0.8739 | 0.8564 | 0.1548 | 0.2400 |
| ET | 0.8929 | 0.8133 | 0.8929 | 0.8727 | 0.8559 | 0.1515 | 0.2354 |
| XGB | 0.8943 | 0.8284 | 0.8943 | 0.8821 | 0.8556 | 0.1466 | 0.2458 |
| DT | 0.8914 | 0.8121 | 0.8914 | 0.8661 | 0.8551 | 0.1490 | 0.2227 |
| LGBM | 0.8941 | 0.8283 | 0.8941 | 0.8828 | 0.8549 | 0.1416 | 0.2428 |
| KNN | 0.8445 | 0.7442 | 0.8445 | 0.8627 | 0.8526 | 0.3103 | 0.3134 |
| GBC | 0.8929 | 0.8119 | 0.8929 | 0.8860 | 0.8505 | 0.1114 | 0.2192 |
| ADA | 0.8888 | 0.7937 | 0.8888 | 0.8908 | 0.8390 | 0.0360 | 0.1227 |
| Ridge | 0.8867 | 0.7215 | 0.8867 | 0.7863 | 0.8335 | -0.0001 | -0.0008 |
| Dummy | 0.8867 | 0.5000 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| LDA | 0.8866 | 0.7217 | 0.8866 | 0.7863 | 0.8334 | -0.0002 | -0.0014 |
| LR | 0.8863 | 0.7415 | 0.8863 | 0.7862 | 0.8333 | -0.0008 | -0.0065 |
| SVM-L | 0.7930 | 0.6306 | 0.7930 | 0.7947 | 0.7876 | -0.0099 | -0.0122 |
| QDA | 0.3741 | 0.7635 | 0.3741 | 0.8934 | 0.4352 | 0.0798 | 0.1935 |
| NB | 0.3263 | 0.6917 | 0.3263 | 0.8694 | 0.3786 | 0.0496 | 0.1318 |

Tabulka G.13: Výsledky pro subset html+lex – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9867 | 0.9981 | 0.9867 | 0.9865 | 0.9865 | 0.9306 | 0.9308 |
| XGB | 0.9860 | 0.9980 | 0.9860 | 0.9858 | 0.9859 | 0.9272 | 0.9274 |
| KNN | 0.9843 | 0.9816 | 0.9843 | 0.9842 | 0.9842 | 0.9184 | 0.9187 |
| GBC | 0.9818 | 0.9971 | 0.9818 | 0.9815 | 0.9815 | 0.9040 | 0.9047 |
| RF | 0.9806 | 0.9945 | 0.9806 | 0.9804 | 0.9800 | 0.8954 | 0.8975 |
| ADA | 0.9798 | 0.9964 | 0.9798 | 0.9795 | 0.9796 | 0.8945 | 0.8949 |
| DT | 0.9766 | 0.9425 | 0.9766 | 0.9767 | 0.9766 | 0.8809 | 0.8809 |
| ET | 0.9722 | 0.9883 | 0.9722 | 0.9724 | 0.9707 | 0.8430 | 0.8509 |
| LDA | 0.9643 | 0.9861 | 0.9643 | 0.9631 | 0.9633 | 0.8069 | 0.8092 |
| Ridge | 0.9548 | 0.9861 | 0.9548 | 0.9557 | 0.9500 | 0.7225 | 0.7475 |
| NB | 0.9083 | 0.9580 | 0.9083 | 0.9392 | 0.9176 | 0.6345 | 0.6613 |
| LR | 0.9165 | 0.9578 | 0.9165 | 0.9141 | 0.9152 | 0.5598 | 0.5604 |
| Dummy | 0.8901 | 0.5000 | 0.8901 | 0.7923 | 0.8384 | 0.0000 | 0.0000 |
| SVM-L | 0.7344 | 0.6485 | 0.7344 | 0.8143 | 0.6756 | 0.0001 | 0.0025 |
| QDA | 0.5878 | 0.9360 | 0.5878 | 0.9038 | 0.6609 | 0.1912 | 0.3116 |

Tabulka G.14: Výsledky pro subset html – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| ET | 0.8882 | 0.6627 | 0.8882 | 0.8554 | 0.8456 | 0.0846 | 0.1534 |
| XGB | 0.8890 | 0.6758 | 0.8890 | 0.8610 | 0.8455 | 0.0825 | 0.1587 |
| RF | 0.8881 | 0.6687 | 0.8881 | 0.8548 | 0.8451 | 0.0813 | 0.1496 |
| LGBM | 0.8897 | 0.6764 | 0.8897 | 0.8704 | 0.8441 | 0.0712 | 0.1582 |
| GBC | 0.8892 | 0.6748 | 0.8892 | 0.8811 | 0.8408 | 0.0480 | 0.1384 |
| ADA | 0.8889 | 0.6499 | 0.8889 | 0.8806 | 0.8400 | 0.0430 | 0.1302 |
| DT | 0.8732 | 0.6433 | 0.8732 | 0.8206 | 0.8385 | 0.0658 | 0.0839 |
| Ridge | 0.8873 | 0.6401 | 0.8873 | 0.8693 | 0.8356 | 0.0139 | 0.0683 |
| LDA | 0.8870 | 0.6403 | 0.8870 | 0.8558 | 0.8354 | 0.0132 | 0.0598 |
| LR | 0.8868 | 0.6233 | 0.8868 | 0.8461 | 0.8353 | 0.0128 | 0.0542 |
| Dummy | 0.8867 | 0.5000 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| SVM-L | 0.8800 | 0.6228 | 0.8800 | 0.8017 | 0.8329 | 0.0093 | 0.0146 |
| KNN | 0.8490 | 0.6286 | 0.8490 | 0.8550 | 0.8168 | 0.0709 | 0.1400 |
| NB | 0.3675 | 0.6230 | 0.3675 | 0.8597 | 0.4337 | 0.0548 | 0.1285 |
| QDA | 0.2869 | 0.6387 | 0.2869 | 0.8696 | 0.3237 | 0.0391 | 0.1170 |

Tabulka G.15: Výsledky pro subset ip – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| KNN | 0.8775 | 0.8336 | 0.8775 | 0.8844 | 0.8780 | 0.4027 | 0.4130 |
| XGB | 0.9020 | 0.8958 | 0.9020 | 0.8895 | 0.8758 | 0.2829 | 0.3564 |
| LGBM | 0.9019 | 0.8956 | 0.9019 | 0.8914 | 0.8739 | 0.2685 | 0.3509 |
| RF | 0.8833 | 0.8440 | 0.8833 | 0.8609 | 0.8679 | 0.2767 | 0.2917 |
| ET | 0.8832 | 0.8026 | 0.8832 | 0.8608 | 0.8678 | 0.2758 | 0.2909 |
| GBC | 0.8993 | 0.8866 | 0.8993 | 0.8920 | 0.8664 | 0.2164 | 0.3159 |
| DT | 0.8809 | 0.7700 | 0.8809 | 0.8577 | 0.8653 | 0.2625 | 0.2767 |
| ADA | 0.8901 | 0.8722 | 0.8901 | 0.8781 | 0.8438 | 0.0682 | 0.1628 |
| LR | 0.8854 | 0.8289 | 0.8854 | 0.8214 | 0.8346 | 0.0100 | 0.0332 |
| LDA | 0.8867 | 0.8080 | 0.8867 | 0.8203 | 0.8337 | 0.0014 | 0.0129 |
| Ridge | 0.8867 | 0.8080 | 0.8867 | 0.7863 | 0.8335 | -0.0001 | -0.0006 |
| Dummy | 0.8867 | 0.5000 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| SVM-L | 0.8547 | 0.6997 | 0.8547 | 0.8144 | 0.8206 | 0.0277 | 0.0389 |
| QDA | 0.6927 | 0.8047 | 0.6927 | 0.8918 | 0.7488 | 0.2573 | 0.3445 |
| NB | 0.5588 | 0.7420 | 0.5588 | 0.8603 | 0.6361 | 0.1216 | 0.1924 |

Tabulka G.16: Výsledky pro subset lex+dns+ip+geo+rdap+tls+html – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9860 | 0.9971 | 0.9860 | 0.9859 | 0.9859 | 0.9266 | 0.9271 |
| LGBM | 0.9856 | 0.9970 | 0.9856 | 0.9855 | 0.9854 | 0.9244 | 0.9250 |
| RF | 0.9777 | 0.9938 | 0.9777 | 0.9776 | 0.9768 | 0.8768 | 0.8810 |
| ET | 0.9746 | 0.9917 | 0.9746 | 0.9745 | 0.9735 | 0.8587 | 0.8639 |
| GBC | 0.9733 | 0.9914 | 0.9733 | 0.9729 | 0.9721 | 0.8517 | 0.8565 |
| DT | 0.9670 | 0.9180 | 0.9670 | 0.9672 | 0.9671 | 0.8318 | 0.8319 |
| ADA | 0.9615 | 0.9814 | 0.9615 | 0.9602 | 0.9605 | 0.7924 | 0.7944 |
| LDA | 0.9429 | 0.9556 | 0.9429 | 0.9422 | 0.9425 | 0.7033 | 0.7035 |
| KNN | 0.9337 | 0.9040 | 0.9337 | 0.9311 | 0.9321 | 0.6446 | 0.6458 |
| Ridge | 0.9345 | 0.9557 | 0.9345 | 0.9304 | 0.9262 | 0.5849 | 0.6135 |
| LR | 0.8842 | 0.7243 | 0.8842 | 0.8261 | 0.8425 | 0.0405 | 0.0662 |
| Dummy | 0.8904 | 0.5000 | 0.8904 | 0.7928 | 0.8388 | 0.0000 | 0.0000 |
| SVM-L | 0.7482 | 0.6698 | 0.7482 | 0.8309 | 0.7599 | 0.0895 | 0.1068 |
| QDA | 0.6229 | 0.9246 | 0.6229 | 0.9050 | 0.6920 | 0.2160 | 0.3330 |
| NB | 0.1505 | 0.7315 | 0.1505 | 0.8506 | 0.1003 | 0.0067 | 0.0356 |

Tabulka G.17: Výsledky pro subset lex+dns+ip+geo+rdap+tls – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9857 | 0.9970 | 0.9857 | 0.9855 | 0.9855 | 0.9248 | 0.9253 |
| LGBM | 0.9857 | 0.9968 | 0.9857 | 0.9856 | 0.9855 | 0.9247 | 0.9254 |
| RF | 0.9796 | 0.9941 | 0.9796 | 0.9794 | 0.9789 | 0.8884 | 0.8915 |
| ET | 0.9766 | 0.9927 | 0.9766 | 0.9764 | 0.9757 | 0.8711 | 0.8752 |
| GBC | 0.9733 | 0.9911 | 0.9733 | 0.9730 | 0.9722 | 0.8521 | 0.8568 |
| DT | 0.9669 | 0.9167 | 0.9669 | 0.9670 | 0.9669 | 0.8307 | 0.8308 |
| ADA | 0.9616 | 0.9811 | 0.9616 | 0.9603 | 0.9606 | 0.7934 | 0.7950 |
| LDA | 0.9413 | 0.9535 | 0.9413 | 0.9405 | 0.9408 | 0.6946 | 0.6949 |
| KNN | 0.9342 | 0.9043 | 0.9342 | 0.9315 | 0.9325 | 0.6462 | 0.6475 |
| Ridge | 0.9324 | 0.9536 | 0.9324 | 0.9279 | 0.9234 | 0.5682 | 0.5987 |
| LR | 0.8840 | 0.7218 | 0.8840 | 0.8253 | 0.8422 | 0.0384 | 0.0632 |
| Dummy | 0.8904 | 0.5000 | 0.8904 | 0.7928 | 0.8388 | 0.0000 | 0.0000 |
| SVM-L | 0.7570 | 0.6526 | 0.7570 | 0.8240 | 0.7611 | 0.0644 | 0.0794 |
| QDA | 0.6554 | 0.9471 | 0.6554 | 0.9113 | 0.7195 | 0.2498 | 0.3691 |
| NB | 0.1505 | 0.7313 | 0.1505 | 0.8508 | 0.1003 | 0.0067 | 0.0357 |

Tabulka G.18: Výsledky pro subset lex+dns+ip+geo+rdap – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9856 | 0.9966 | 0.9856 | 0.9855 | 0.9854 | 0.9244 | 0.9250 |
| LGBM | 0.9853 | 0.9967 | 0.9853 | 0.9852 | 0.9851 | 0.9228 | 0.9235 |
| RF | 0.9792 | 0.9943 | 0.9792 | 0.9790 | 0.9785 | 0.8862 | 0.8894 |
| ET | 0.9773 | 0.9928 | 0.9773 | 0.9772 | 0.9765 | 0.8753 | 0.8791 |
| GBC | 0.9740 | 0.9910 | 0.9740 | 0.9736 | 0.9729 | 0.8563 | 0.8605 |
| DT | 0.9673 | 0.9187 | 0.9673 | 0.9675 | 0.9674 | 0.8332 | 0.8333 |
| ADA | 0.9614 | 0.9809 | 0.9614 | 0.9601 | 0.9603 | 0.7918 | 0.7936 |
| LDA | 0.9403 | 0.9485 | 0.9403 | 0.9389 | 0.9395 | 0.6860 | 0.6864 |
| KNN | 0.9338 | 0.9044 | 0.9338 | 0.9317 | 0.9325 | 0.6484 | 0.6492 |
| Ridge | 0.9321 | 0.9486 | 0.9321 | 0.9278 | 0.9229 | 0.5645 | 0.5963 |
| Dummy | 0.8904 | 0.5000 | 0.8904 | 0.7928 | 0.8388 | 0.0000 | 0.0000 |
| LR | 0.8872 | 0.6219 | 0.8872 | 0.8104 | 0.8387 | 0.0053 | 0.0123 |
| SVM-L | 0.8553 | 0.5332 | 0.8553 | 0.8103 | 0.8181 | 0.0133 | 0.0263 |
| QDA | 0.6509 | 0.9467 | 0.6509 | 0.9116 | 0.7157 | 0.2466 | 0.3673 |
| NB | 0.1434 | 0.6702 | 0.1434 | 0.8418 | 0.0875 | 0.0048 | 0.0265 |

Tabulka G.19: Výsledky pro subset lex+dns+ip+geo – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9778 | 0.9932 | 0.9778 | 0.9774 | 0.9774 | 0.8828 | 0.8835 |
| LGBM | 0.9762 | 0.9925 | 0.9762 | 0.9757 | 0.9758 | 0.8740 | 0.8748 |
| RF | 0.9712 | 0.9891 | 0.9712 | 0.9706 | 0.9700 | 0.8409 | 0.8452 |
| ET | 0.9699 | 0.9871 | 0.9699 | 0.9693 | 0.9685 | 0.8319 | 0.8373 |
| DT | 0.9595 | 0.9005 | 0.9595 | 0.9599 | 0.9597 | 0.7944 | 0.7945 |
| GBC | 0.9611 | 0.9827 | 0.9611 | 0.9598 | 0.9591 | 0.7805 | 0.7872 |
| ADA | 0.9514 | 0.9723 | 0.9514 | 0.9491 | 0.9495 | 0.7321 | 0.7356 |
| KNN | 0.9326 | 0.9049 | 0.9326 | 0.9319 | 0.9320 | 0.6487 | 0.6498 |
| LDA | 0.9303 | 0.9320 | 0.9303 | 0.9276 | 0.9287 | 0.6264 | 0.6277 |
| Ridge | 0.9187 | 0.9321 | 0.9187 | 0.9116 | 0.9033 | 0.4394 | 0.4903 |
| SVM-L | 0.8904 | 0.4509 | 0.8904 | 0.7928 | 0.8388 | 0.0000 | 0.0000 |
| Dummy | 0.8904 | 0.5000 | 0.8904 | 0.7928 | 0.8388 | 0.0000 | 0.0000 |
| LR | 0.8900 | 0.5718 | 0.8900 | 0.7964 | 0.8386 | -0.0004 | -0.0029 |
| QDA | 0.6509 | 0.9407 | 0.6509 | 0.9115 | 0.7157 | 0.2465 | 0.3671 |
| NB | 0.1174 | 0.5649 | 0.1174 | 0.7482 | 0.0401 | -0.0011 | -0.0177 |

Tabulka G.20: Výsledky pro subset lex+dns+ip+tls+geo+rdap+html – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9820 | 0.9945 | 0.9820 | 0.9818 | 0.9817 | 0.9060 | 0.9070 |
| LGBM | 0.9817 | 0.9951 | 0.9817 | 0.9815 | 0.9813 | 0.9035 | 0.9050 |
| RF | 0.9707 | 0.9889 | 0.9707 | 0.9709 | 0.9690 | 0.8358 | 0.8445 |
| GBC | 0.9704 | 0.9896 | 0.9704 | 0.9701 | 0.9689 | 0.8363 | 0.8427 |
| ET | 0.9654 | 0.9864 | 0.9654 | 0.9655 | 0.9630 | 0.8026 | 0.8142 |
| DT | 0.9597 | 0.9010 | 0.9597 | 0.9600 | 0.9598 | 0.7977 | 0.7979 |
| ADA | 0.9587 | 0.9787 | 0.9587 | 0.9572 | 0.9574 | 0.7789 | 0.7815 |
| LDA | 0.9382 | 0.9544 | 0.9382 | 0.9370 | 0.9375 | 0.6818 | 0.6822 |
| Ridge | 0.9351 | 0.9546 | 0.9351 | 0.9310 | 0.9274 | 0.5999 | 0.6254 |
| KNN | 0.9259 | 0.8794 | 0.9259 | 0.9223 | 0.9236 | 0.6041 | 0.6061 |
| LR | 0.8820 | 0.7088 | 0.8820 | 0.8175 | 0.8378 | 0.0258 | 0.0452 |
| Dummy | 0.8884 | 0.5000 | 0.8884 | 0.7892 | 0.8359 | 0.0000 | 0.0000 |
| SVM-L | 0.7912 | 0.6510 | 0.7912 | 0.8110 | 0.7600 | 0.0159 | 0.0327 |
| QDA | 0.6440 | 0.9088 | 0.6440 | 0.9004 | 0.7092 | 0.2295 | 0.3379 |
| NB | 0.3234 | 0.7360 | 0.3234 | 0.8628 | 0.3772 | 0.0433 | 0.1152 |

Tabulka G.21: Výsledky pro subset lex+dns+ip+tls+geo+rdap – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9821 | 0.9943 | 0.9821 | 0.9819 | 0.9818 | 0.9062 | 0.9076 |
| LGBM | 0.9818 | 0.9945 | 0.9818 | 0.9816 | 0.9814 | 0.9045 | 0.9058 |
| RF | 0.9734 | 0.9892 | 0.9734 | 0.9734 | 0.9720 | 0.8525 | 0.8593 |
| GBC | 0.9709 | 0.9896 | 0.9709 | 0.9706 | 0.9695 | 0.8395 | 0.8457 |
| ET | 0.9706 | 0.9879 | 0.9706 | 0.9706 | 0.9691 | 0.8365 | 0.8441 |
| DT | 0.9605 | 0.8996 | 0.9605 | 0.9606 | 0.9605 | 0.8009 | 0.8012 |
| ADA | 0.9597 | 0.9789 | 0.9597 | 0.9583 | 0.9586 | 0.7857 | 0.7878 |
| LDA | 0.9366 | 0.9523 | 0.9366 | 0.9353 | 0.9358 | 0.6728 | 0.6732 |
| Ridge | 0.9329 | 0.9525 | 0.9329 | 0.9284 | 0.9246 | 0.5834 | 0.6106 |
| KNN | 0.9257 | 0.8794 | 0.9257 | 0.9221 | 0.9235 | 0.6031 | 0.6051 |
| LR | 0.8820 | 0.7110 | 0.8820 | 0.8181 | 0.8378 | 0.0259 | 0.0462 |
| Dummy | 0.8884 | 0.5000 | 0.8884 | 0.7892 | 0.8359 | 0.0000 | 0.0000 |
| SVM-L | 0.7912 | 0.6510 | 0.7912 | 0.8110 | 0.7600 | 0.0159 | 0.0327 |
| QDA | 0.6430 | 0.9405 | 0.6430 | 0.9074 | 0.7082 | 0.2392 | 0.3569 |
| NB | 0.3240 | 0.7359 | 0.3240 | 0.8624 | 0.3780 | 0.0432 | 0.1147 |

Tabulka G.22: Výsledky pro subset lex+dns+ip+tls+geo – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9736 | 0.9897 | 0.9736 | 0.9731 | 0.9731 | 0.8620 | 0.8631 |
| XGB | 0.9733 | 0.9889 | 0.9733 | 0.9728 | 0.9728 | 0.8605 | 0.8616 |
| RF | 0.9656 | 0.9826 | 0.9656 | 0.9650 | 0.9637 | 0.8079 | 0.8157 |
| ET | 0.9632 | 0.9807 | 0.9632 | 0.9626 | 0.9610 | 0.7926 | 0.8020 |
| GBC | 0.9582 | 0.9827 | 0.9582 | 0.9568 | 0.9558 | 0.7659 | 0.7739 |
| DT | 0.9488 | 0.8722 | 0.9488 | 0.9491 | 0.9489 | 0.7428 | 0.7430 |
| ADA | 0.9483 | 0.9717 | 0.9483 | 0.9457 | 0.9461 | 0.7176 | 0.7220 |
| LDA | 0.9284 | 0.9392 | 0.9284 | 0.9254 | 0.9266 | 0.6211 | 0.6226 |
| KNN | 0.9255 | 0.8835 | 0.9255 | 0.9238 | 0.9245 | 0.6147 | 0.6153 |
| Ridge | 0.9187 | 0.9394 | 0.9187 | 0.9108 | 0.9048 | 0.4616 | 0.5038 |
| Dummy | 0.8884 | 0.5000 | 0.8884 | 0.7892 | 0.8359 | 0.0000 | 0.0000 |
| LR | 0.8871 | 0.6485 | 0.8871 | 0.8125 | 0.8357 | 0.0012 | 0.0083 |
| QDA | 0.6291 | 0.9376 | 0.6291 | 0.9065 | 0.6962 | 0.2274 | 0.3467 |
| SVM-L | 0.6647 | 0.5957 | 0.6647 | 0.8042 | 0.6380 | 0.0427 | 0.0529 |
| NB | 0.2024 | 0.6722 | 0.2024 | 0.8302 | 0.1979 | 0.0092 | 0.0358 |

Tabulka G.23: Výsledky pro subset lex+dns+ip – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9721 | 0.9881 | 0.9721 | 0.9715 | 0.9715 | 0.8534 | 0.8549 |
| LGBM | 0.9714 | 0.9887 | 0.9714 | 0.9708 | 0.9708 | 0.8498 | 0.8512 |
| RF | 0.9640 | 0.9807 | 0.9640 | 0.9633 | 0.9619 | 0.7980 | 0.8064 |
| ET | 0.9627 | 0.9792 | 0.9627 | 0.9621 | 0.9604 | 0.7897 | 0.7992 |
| GBC | 0.9605 | 0.9808 | 0.9605 | 0.9593 | 0.9585 | 0.7808 | 0.7877 |
| ADA | 0.9498 | 0.9705 | 0.9498 | 0.9474 | 0.9478 | 0.7271 | 0.7309 |
| DT | 0.9461 | 0.8696 | 0.9461 | 0.9468 | 0.9464 | 0.7312 | 0.7314 |
| LDA | 0.9269 | 0.9290 | 0.9269 | 0.9231 | 0.9245 | 0.6076 | 0.6098 |
| KNN | 0.9205 | 0.8598 | 0.9205 | 0.9127 | 0.9139 | 0.5355 | 0.5476 |
| Ridge | 0.9175 | 0.9295 | 0.9175 | 0.9102 | 0.9021 | 0.4426 | 0.4919 |
| Dummy | 0.8884 | 0.5000 | 0.8884 | 0.7892 | 0.8359 | 0.0000 | 0.0000 |
| LR | 0.8882 | 0.4478 | 0.8882 | 0.7892 | 0.8358 | -0.0003 | -0.0019 |
| QDA | 0.7226 | 0.9367 | 0.7226 | 0.9116 | 0.7735 | 0.3158 | 0.4182 |
| SVM-L | 0.8107 | 0.4429 | 0.8107 | 0.7116 | 0.7545 | 0.0000 | 0.0000 |
| NB | 0.1595 | 0.5829 | 0.1595 | 0.8016 | 0.1246 | -0.0004 | -0.0011 |

Tabulka G.24: Výsledky pro subset lex – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9143 | 0.8911 | 0.9143 | 0.9039 | 0.9021 | 0.4605 | 0.4901 |
| XGB | 0.9123 | 0.8852 | 0.9123 | 0.9012 | 0.9017 | 0.4649 | 0.4861 |
| GBC | 0.9086 | 0.8753 | 0.9086 | 0.8965 | 0.8913 | 0.3884 | 0.4339 |
| RF | 0.9019 | 0.8543 | 0.9019 | 0.8866 | 0.8888 | 0.3900 | 0.4121 |
| ET | 0.8982 | 0.8156 | 0.8982 | 0.8810 | 0.8838 | 0.3601 | 0.3831 |
| ADA | 0.8993 | 0.8542 | 0.8993 | 0.8810 | 0.8761 | 0.2927 | 0.3454 |
| DT | 0.8656 | 0.6856 | 0.8656 | 0.8688 | 0.8671 | 0.3466 | 0.3468 |
| LDA | 0.8882 | 0.8111 | 0.8882 | 0.8615 | 0.8659 | 0.2429 | 0.2747 |
| KNN | 0.8759 | 0.6237 | 0.8759 | 0.8668 | 0.8544 | 0.2099 | 0.2657 |
| Ridge | 0.8892 | 0.8111 | 0.8892 | 0.8671 | 0.8434 | 0.0666 | 0.1497 |
| LR | 0.8867 | 0.4891 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| NB | 0.8867 | 0.5540 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| Dummy | 0.8867 | 0.5000 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| SVM-L | 0.7320 | 0.5312 | 0.7320 | 0.6316 | 0.6714 | 0.0000 | 0.0000 |
| QDA | 0.2948 | 0.8104 | 0.2948 | 0.8831 | 0.3319 | 0.0473 | 0.1404 |

Tabulka G.25: Výsledky pro subset lex – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9235 | 0.9181 | 0.9235 | 0.9156 | 0.9156 | 0.5314 | 0.5491 |
| LGBM | 0.9233 | 0.9205 | 0.9233 | 0.9153 | 0.9141 | 0.5185 | 0.5417 |
| RF | 0.9161 | 0.8957 | 0.9161 | 0.9054 | 0.9044 | 0.4588 | 0.4863 |
| GBC | 0.9140 | 0.9003 | 0.9140 | 0.9030 | 0.8990 | 0.4185 | 0.4588 |
| ET | 0.9124 | 0.8877 | 0.9124 | 0.9002 | 0.8989 | 0.4234 | 0.4548 |
| ADA | 0.9043 | 0.8770 | 0.9043 | 0.8877 | 0.8856 | 0.3356 | 0.3774 |
| DT | 0.8827 | 0.7182 | 0.8827 | 0.8871 | 0.8848 | 0.4207 | 0.4212 |
| LDA | 0.8923 | 0.8321 | 0.8923 | 0.8685 | 0.8734 | 0.2709 | 0.2973 |
| KNN | 0.8983 | 0.7603 | 0.8983 | 0.8780 | 0.8686 | 0.2123 | 0.2816 |
| Ridge | 0.8935 | 0.8320 | 0.8935 | 0.8730 | 0.8504 | 0.0812 | 0.1681 |
| LR | 0.8904 | 0.4764 | 0.8904 | 0.7928 | 0.8388 | 0.0000 | 0.0000 |
| NB | 0.8904 | 0.5573 | 0.8904 | 0.7928 | 0.8388 | 0.0000 | 0.0000 |
| Dummy | 0.8904 | 0.5000 | 0.8904 | 0.7928 | 0.8388 | 0.0000 | 0.0000 |
| SVM-L | 0.8123 | 0.5191 | 0.8123 | 0.7147 | 0.7571 | 0.0000 | 0.0000 |
| QDA | 0.2811 | 0.8301 | 0.2811 | 0.8933 | 0.3153 | 0.0453 | 0.1432 |

Tabulka G.26: Výsledky pro subset rdap – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9343 | 0.9373 | 0.9343 | 0.9297 | 0.9278 | 0.6115 | 0.6306 |
| LGBM | 0.9341 | 0.9369 | 0.9341 | 0.9298 | 0.9269 | 0.6046 | 0.6272 |
| RF | 0.9281 | 0.9300 | 0.9281 | 0.9219 | 0.9206 | 0.5718 | 0.5911 |
| ET | 0.9256 | 0.9025 | 0.9256 | 0.9188 | 0.9173 | 0.5519 | 0.5734 |
| GBC | 0.9236 | 0.9188 | 0.9236 | 0.9177 | 0.9120 | 0.5133 | 0.5504 |
| DT | 0.9095 | 0.8239 | 0.9095 | 0.9015 | 0.9042 | 0.4982 | 0.5038 |
| ADA | 0.9071 | 0.8953 | 0.9071 | 0.8937 | 0.8907 | 0.3882 | 0.4274 |
| KNN | 0.8849 | 0.8387 | 0.8849 | 0.8944 | 0.8891 | 0.4697 | 0.4720 |
| LDA | 0.8811 | 0.8438 | 0.8811 | 0.8463 | 0.8548 | 0.1740 | 0.2022 |
| LR | 0.8640 | 0.5839 | 0.8640 | 0.8401 | 0.8499 | 0.1906 | 0.1972 |
| Ridge | 0.8882 | 0.8438 | 0.8882 | 0.8661 | 0.8396 | 0.0412 | 0.1169 |
| NB | 0.8867 | 0.7349 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| Dummy | 0.8867 | 0.5000 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| QDA | 0.6925 | 0.8442 | 0.6925 | 0.8932 | 0.7486 | 0.2598 | 0.3489 |
| SVM-L | 0.5716 | 0.5686 | 0.5716 | 0.7825 | 0.6484 | -0.0518 | -0.0657 |

Tabulka G.27: Výsledky pro subset tls+geo+ip – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9873 | 0.9983 | 0.9873 | 0.9875 | 0.9874 | 0.9358 | 0.9359 |
| LGBM | 0.9873 | 0.9982 | 0.9873 | 0.9875 | 0.9874 | 0.9359 | 0.9361 |
| RF | 0.9865 | 0.9978 | 0.9865 | 0.9865 | 0.9865 | 0.9310 | 0.9310 |
| ET | 0.9848 | 0.9968 | 0.9848 | 0.9847 | 0.9848 | 0.9217 | 0.9218 |
| DT | 0.9825 | 0.9562 | 0.9825 | 0.9825 | 0.9825 | 0.9105 | 0.9105 |
| GBC | 0.9787 | 0.9962 | 0.9787 | 0.9797 | 0.9791 | 0.8947 | 0.8954 |
| ADA | 0.9760 | 0.9953 | 0.9760 | 0.9767 | 0.9763 | 0.8802 | 0.8806 |
| LDA | 0.9456 | 0.9794 | 0.9456 | 0.9505 | 0.9474 | 0.7410 | 0.7439 |
| KNN | 0.9303 | 0.9563 | 0.9303 | 0.9325 | 0.9313 | 0.6540 | 0.6545 |
| NB | 0.9072 | 0.9556 | 0.9072 | 0.9188 | 0.9119 | 0.5747 | 0.5798 |
| QDA | 0.8901 | 0.9726 | 0.8901 | 0.9417 | 0.9044 | 0.6031 | 0.6507 |
| Ridge | 0.9203 | 0.9794 | 0.9203 | 0.9169 | 0.9033 | 0.4360 | 0.5007 |
| LR | 0.8927 | 0.9526 | 0.8927 | 0.8600 | 0.8604 | 0.1631 | 0.2141 |
| Dummy | 0.8901 | 0.5000 | 0.8901 | 0.7923 | 0.8384 | 0.0000 | 0.0000 |
| SVM-L | 0.7799 | 0.8371 | 0.7799 | 0.7826 | 0.7512 | -0.0052 | -0.0205 |

Tabulka G.28: Výsledky pro subset tls – malware

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.8877 | 0.7209 | 0.8877 | 0.8606 | 0.8383 | 0.0330 | 0.0990 |
| ET | 0.8878 | 0.7206 | 0.8878 | 0.8635 | 0.8383 | 0.0326 | 0.1002 |
| DT | 0.8876 | 0.7194 | 0.8876 | 0.8559 | 0.8382 | 0.0324 | 0.0942 |
| XGB | 0.8876 | 0.7194 | 0.8876 | 0.8608 | 0.8381 | 0.0314 | 0.0966 |
| LGBM | 0.8873 | 0.7208 | 0.8873 | 0.8546 | 0.8377 | 0.0291 | 0.0880 |
| GBC | 0.8876 | 0.7138 | 0.8876 | 0.8603 | 0.8376 | 0.0276 | 0.0910 |
| LDA | 0.8860 | 0.6626 | 0.8860 | 0.8352 | 0.8366 | 0.0232 | 0.0633 |
| ADA | 0.8866 | 0.6902 | 0.8866 | 0.8442 | 0.8351 | 0.0116 | 0.0475 |
| Ridge | 0.8869 | 0.6590 | 0.8869 | 0.8432 | 0.8342 | 0.0048 | 0.0336 |
| LR | 0.8866 | 0.6336 | 0.8866 | 0.7976 | 0.8335 | 0.0001 | 0.0018 |
| Dummy | 0.8867 | 0.5000 | 0.8867 | 0.7863 | 0.8335 | 0.0000 | 0.0000 |
| QDA | 0.8788 | 0.5174 | 0.8788 | 0.8035 | 0.8329 | 0.0244 | 0.0330 |
| NB | 0.8001 | 0.6260 | 0.8001 | 0.8442 | 0.8190 | 0.2087 | 0.2165 |
| SVM-L | 0.7544 | 0.6082 | 0.7544 | 0.8233 | 0.7662 | 0.1665 | 0.1578 |
| KNN | 0.8134 | 0.6126 | 0.8134 | 0.8627 | 0.7619 | 0.0153 | 0.0679 |

Tabulka G.29: Výsledky pro subset dns – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.9165 | 0.9120 | 0.9165 | 0.9135 | 0.9143 | 0.6861 | 0.6886 |
| ET | 0.9155 | 0.8910 | 0.9155 | 0.9125 | 0.9133 | 0.6829 | 0.6851 |
| KNN | 0.9142 | 0.8860 | 0.9142 | 0.9109 | 0.9116 | 0.6757 | 0.6787 |
| XGB | 0.9166 | 0.9175 | 0.9166 | 0.9129 | 0.9115 | 0.6681 | 0.6785 |
| LGBM | 0.9175 | 0.9176 | 0.9175 | 0.9145 | 0.9114 | 0.6649 | 0.6796 |
| DT | 0.8958 | 0.8503 | 0.8958 | 0.8957 | 0.8957 | 0.6268 | 0.6271 |
| GBC | 0.8911 | 0.8799 | 0.8911 | 0.8921 | 0.8728 | 0.4991 | 0.5538 |
| ADA | 0.8632 | 0.8417 | 0.8632 | 0.8536 | 0.8335 | 0.3352 | 0.3997 |
| LDA | 0.8491 | 0.7843 | 0.8491 | 0.8264 | 0.8136 | 0.2532 | 0.3100 |
| Ridge | 0.8346 | 0.7842 | 0.8346 | 0.7962 | 0.7724 | 0.0722 | 0.1401 |
| LR | 0.8242 | 0.6988 | 0.8242 | 0.7307 | 0.7582 | 0.0143 | 0.0262 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| SVM-L | 0.6466 | 0.6454 | 0.6466 | 0.7720 | 0.6756 | 0.1180 | 0.1475 |
| QDA | 0.5847 | 0.7880 | 0.5847 | 0.8468 | 0.6332 | 0.2208 | 0.3161 |
| NB | 0.3432 | 0.6313 | 0.3432 | 0.8202 | 0.3546 | 0.0640 | 0.1513 |

Tabulka G.30: Výsledky pro subset geo – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.8461 | 0.8194 | 0.8461 | 0.8210 | 0.8106 | 0.2416 | 0.2948 |
| ET | 0.8461 | 0.8173 | 0.8461 | 0.8210 | 0.8102 | 0.2400 | 0.2939 |
| LGBM | 0.8483 | 0.8292 | 0.8483 | 0.8274 | 0.8098 | 0.2351 | 0.2999 |
| DT | 0.8446 | 0.8161 | 0.8446 | 0.8176 | 0.8092 | 0.2367 | 0.2870 |
| XGB | 0.8474 | 0.8277 | 0.8474 | 0.8249 | 0.8092 | 0.2333 | 0.2953 |
| KNN | 0.8380 | 0.7701 | 0.8380 | 0.8056 | 0.8031 | 0.2147 | 0.2555 |
| GBC | 0.8446 | 0.8170 | 0.8446 | 0.8218 | 0.8000 | 0.1915 | 0.2634 |
| ADA | 0.8342 | 0.7980 | 0.8342 | 0.7815 | 0.7674 | 0.0511 | 0.1055 |
| Ridge | 0.8317 | 0.6927 | 0.8317 | 0.6917 | 0.7553 | -0.0001 | -0.0010 |
| LDA | 0.8317 | 0.6928 | 0.8317 | 0.6917 | 0.7553 | -0.0001 | -0.0010 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| LR | 0.8311 | 0.7075 | 0.8311 | 0.7085 | 0.7551 | -0.0008 | -0.0044 |
| SVM-L | 0.7870 | 0.5542 | 0.7870 | 0.7089 | 0.7393 | 0.0064 | 0.0019 |
| QDA | 0.4123 | 0.7363 | 0.4123 | 0.8469 | 0.4425 | 0.1110 | 0.2242 |
| NB | 0.3427 | 0.6407 | 0.3427 | 0.7833 | 0.3620 | 0.0453 | 0.1011 |

Tabulka G.31: Výsledky pro subset html – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.8410 | 0.7060 | 0.8410 | 0.8313 | 0.7824 | 0.1133 | 0.2110 |
| LGBM | 0.8414 | 0.7099 | 0.8414 | 0.8357 | 0.7823 | 0.1127 | 0.2145 |
| XGB | 0.8411 | 0.7090 | 0.8411 | 0.8327 | 0.7822 | 0.1125 | 0.2115 |
| ET | 0.8403 | 0.7009 | 0.8403 | 0.8260 | 0.7820 | 0.1117 | 0.2046 |
| KNN | 0.8409 | 0.6828 | 0.8409 | 0.8346 | 0.7811 | 0.1075 | 0.2087 |
| GBC | 0.8416 | 0.7078 | 0.8416 | 0.8485 | 0.7804 | 0.1041 | 0.2174 |
| ADA | 0.8411 | 0.6954 | 0.8411 | 0.8476 | 0.7793 | 0.0995 | 0.2111 |
| DT | 0.8285 | 0.6875 | 0.8285 | 0.7733 | 0.7745 | 0.0854 | 0.1272 |
| LR | 0.8313 | 0.6499 | 0.8313 | 0.7142 | 0.7553 | -0.0000 | 0.0011 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| Ridge | 0.8315 | 0.6633 | 0.8315 | 0.6917 | 0.7552 | -0.0004 | -0.0034 |
| LDA | 0.8314 | 0.6617 | 0.8314 | 0.6917 | 0.7551 | -0.0007 | -0.0063 |
| SVM-L | 0.7869 | 0.6562 | 0.7869 | 0.7278 | 0.7350 | 0.0172 | 0.0288 |
| NB | 0.4640 | 0.6623 | 0.4640 | 0.8337 | 0.5074 | 0.1309 | 0.2305 |
| QDA | 0.4459 | 0.6664 | 0.4459 | 0.8477 | 0.4837 | 0.1304 | 0.2437 |

Tabulka G.32: Výsledky pro subset ip – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.8669 | 0.8812 | 0.8669 | 0.8524 | 0.8511 | 0.4242 | 0.4478 |
| LGBM | 0.8668 | 0.8817 | 0.8668 | 0.8522 | 0.8488 | 0.4117 | 0.4407 |
| KNN | 0.8553 | 0.8210 | 0.8553 | 0.8452 | 0.8469 | 0.4289 | 0.4379 |
| RF | 0.8437 | 0.8314 | 0.8437 | 0.8312 | 0.8359 | 0.3888 | 0.3930 |
| ET | 0.8425 | 0.7755 | 0.8425 | 0.8296 | 0.8343 | 0.3824 | 0.3868 |
| GBC | 0.8589 | 0.8582 | 0.8589 | 0.8421 | 0.8326 | 0.3372 | 0.3833 |
| DT | 0.8406 | 0.7284 | 0.8406 | 0.8273 | 0.8322 | 0.3740 | 0.3786 |
| ADA | 0.8404 | 0.8359 | 0.8404 | 0.8126 | 0.7911 | 0.1533 | 0.2249 |
| LDA | 0.8317 | 0.7536 | 0.8317 | 0.7283 | 0.7556 | 0.0015 | 0.0107 |
| Ridge | 0.8317 | 0.7537 | 0.8317 | 0.6917 | 0.7553 | -0.0001 | -0.0010 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| LR | 0.8306 | 0.7597 | 0.8306 | 0.7085 | 0.7551 | -0.0007 | -0.0047 |
| SVM-L | 0.7730 | 0.6588 | 0.7730 | 0.7781 | 0.7467 | 0.1158 | 0.1512 |
| QDA | 0.5441 | 0.7503 | 0.5441 | 0.8516 | 0.5921 | 0.1961 | 0.3035 |
| NB | 0.5152 | 0.6785 | 0.5152 | 0.8361 | 0.5638 | 0.1630 | 0.2599 |

Tabulka G.33: Výsledky pro subset lex – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9370 | 0.9352 | 0.9370 | 0.9356 | 0.9337 | 0.7527 | 0.7613 |
| XGB | 0.9341 | 0.9303 | 0.9341 | 0.9321 | 0.9311 | 0.7441 | 0.7507 |
| GBC | 0.9255 | 0.9204 | 0.9255 | 0.9248 | 0.9194 | 0.6940 | 0.7125 |
| RF | 0.9238 | 0.9144 | 0.9238 | 0.9215 | 0.9188 | 0.6944 | 0.7068 |
| ET | 0.9213 | 0.8860 | 0.9213 | 0.9187 | 0.9158 | 0.6824 | 0.6959 |
| ADA | 0.9066 | 0.8994 | 0.9066 | 0.9018 | 0.8989 | 0.6161 | 0.6325 |
| LDA | 0.8987 | 0.8850 | 0.8987 | 0.8952 | 0.8864 | 0.5604 | 0.5920 |
| DT | 0.8761 | 0.7909 | 0.8761 | 0.8780 | 0.8769 | 0.5637 | 0.5639 |
| Ridge | 0.8929 | 0.8852 | 0.8929 | 0.8945 | 0.8751 | 0.5083 | 0.5627 |
| KNN | 0.8635 | 0.7860 | 0.8635 | 0.8479 | 0.8475 | 0.4106 | 0.4329 |
| LR | 0.8317 | 0.4768 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| NB | 0.8317 | 0.5302 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| SVM-L | 0.8317 | 0.4927 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| QDA | 0.3245 | 0.8671 | 0.3245 | 0.8487 | 0.3192 | 0.0676 | 0.1740 |

Tabulka G.34: Výsledky pro subset rdap – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9533 | 0.9590 | 0.9533 | 0.9525 | 0.9518 | 0.8227 | 0.8266 |
| XGB | 0.9522 | 0.9589 | 0.9522 | 0.9512 | 0.9509 | 0.8204 | 0.8230 |
| RF | 0.9502 | 0.9572 | 0.9502 | 0.9491 | 0.9487 | 0.8121 | 0.8151 |
| ET | 0.9467 | 0.9410 | 0.9467 | 0.9455 | 0.9450 | 0.7980 | 0.8016 |
| GBC | 0.9358 | 0.9420 | 0.9358 | 0.9342 | 0.9326 | 0.7489 | 0.7568 |
| DT | 0.9319 | 0.8981 | 0.9319 | 0.9313 | 0.9315 | 0.7541 | 0.7543 |
| KNN | 0.9296 | 0.8997 | 0.9296 | 0.9275 | 0.9281 | 0.7381 | 0.7396 |
| ADA | 0.9196 | 0.9229 | 0.9196 | 0.9160 | 0.9159 | 0.6877 | 0.6938 |
| LDA | 0.8576 | 0.8538 | 0.8576 | 0.8402 | 0.8421 | 0.3919 | 0.4102 |
| Ridge | 0.8476 | 0.8539 | 0.8476 | 0.8271 | 0.8067 | 0.2207 | 0.2905 |
| NB | 0.8317 | 0.7152 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| LR | 0.6944 | 0.4346 | 0.6944 | 0.6868 | 0.6857 | -0.0985 | -0.1081 |
| QDA | 0.6038 | 0.8827 | 0.6038 | 0.8596 | 0.6506 | 0.2492 | 0.3524 |
| SVM-L | 0.4393 | 0.4665 | 0.4393 | 0.6703 | 0.4658 | -0.1430 | -0.1609 |

Tabulka G.35: Výsledky pro subset tls – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| DT | 0.8370 | 0.6894 | 0.8370 | 0.8323 | 0.7706 | 0.0633 | 0.1591 |
| RF | 0.8373 | 0.6889 | 0.8373 | 0.8393 | 0.7705 | 0.0629 | 0.1633 |
| ET | 0.8371 | 0.6887 | 0.8371 | 0.8362 | 0.7705 | 0.0629 | 0.1612 |
| XGB | 0.8370 | 0.6919 | 0.8370 | 0.8360 | 0.7703 | 0.0620 | 0.1600 |
| LGBM | 0.8369 | 0.6885 | 0.8369 | 0.8346 | 0.7701 | 0.0613 | 0.1580 |
| GBC | 0.8366 | 0.6848 | 0.8366 | 0.8371 | 0.7689 | 0.0562 | 0.1531 |
| ADA | 0.8356 | 0.6326 | 0.8356 | 0.8327 | 0.7670 | 0.0482 | 0.1385 |
| Ridge | 0.8349 | 0.6280 | 0.8349 | 0.8172 | 0.7667 | 0.0473 | 0.1275 |
| LDA | 0.8330 | 0.6279 | 0.8330 | 0.7916 | 0.7665 | 0.0469 | 0.1093 |
| KNN | 0.8342 | 0.5444 | 0.8342 | 0.8192 | 0.7641 | 0.0364 | 0.1129 |
| NB | 0.8280 | 0.5668 | 0.8280 | 0.7525 | 0.7608 | 0.0239 | 0.0522 |
| Dummy | 0.8317 | 0.5000 | 0.8317 | 0.6917 | 0.7553 | 0.0000 | 0.0000 |
| LR | 0.7481 | 0.5816 | 0.7481 | 0.7399 | 0.7438 | 0.0706 | 0.0707 |
| SVM-L | 0.6982 | 0.5314 | 0.6982 | 0.7315 | 0.6935 | 0.0594 | 0.0544 |
| QDA | 0.6663 | 0.5838 | 0.6663 | 0.7567 | 0.6503 | 0.0769 | 0.0889 |

Tabulka G.36: Výsledky pro subset dns+rdap+tls+geo+ip+html+lex – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9970 | 0.9999 | 0.9970 | 0.9970 | 0.9970 | 0.9891 | 0.9891 |
| LGBM | 0.9970 | 0.9999 | 0.9970 | 0.9970 | 0.9970 | 0.9893 | 0.9893 |
| RF | 0.9916 | 0.9995 | 0.9916 | 0.9917 | 0.9916 | 0.9695 | 0.9698 |
| GBC | 0.9899 | 0.9992 | 0.9899 | 0.9899 | 0.9899 | 0.9638 | 0.9638 |
| ADA | 0.9896 | 0.9991 | 0.9896 | 0.9896 | 0.9896 | 0.9627 | 0.9627 |
| DT | 0.9887 | 0.9808 | 0.9887 | 0.9887 | 0.9887 | 0.9595 | 0.9595 |
| ET | 0.9863 | 0.9990 | 0.9863 | 0.9865 | 0.9861 | 0.9495 | 0.9505 |
| LDA | 0.9758 | 0.9956 | 0.9758 | 0.9757 | 0.9757 | 0.9125 | 0.9126 |
| Ridge | 0.9743 | 0.9956 | 0.9743 | 0.9741 | 0.9738 | 0.9047 | 0.9058 |
| KNN | 0.9630 | 0.9646 | 0.9630 | 0.9626 | 0.9627 | 0.8650 | 0.8654 |
| LR | 0.8985 | 0.9488 | 0.8985 | 0.8972 | 0.8978 | 0.6310 | 0.6312 |
| Dummy | 0.8326 | 0.5000 | 0.8326 | 0.6932 | 0.7565 | 0.0000 | 0.0000 |
| SVM-L | 0.7884 | 0.6171 | 0.7884 | 0.7609 | 0.7536 | 0.0864 | 0.1039 |
| QDA | 0.6856 | 0.9565 | 0.6856 | 0.8789 | 0.7253 | 0.3430 | 0.4404 |
| NB | 0.6163 | 0.9389 | 0.6163 | 0.8782 | 0.6614 | 0.2775 | 0.3951 |

Tabulka G.37: Výsledky pro subset dns+rdap – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9878 | 0.9989 | 0.9878 | 0.9878 | 0.9878 | 0.9562 | 0.9562 |
| LGBM | 0.9867 | 0.9987 | 0.9867 | 0.9867 | 0.9867 | 0.9522 | 0.9522 |
| RF | 0.9865 | 0.9975 | 0.9865 | 0.9865 | 0.9864 | 0.9509 | 0.9512 |
| ET | 0.9825 | 0.9950 | 0.9825 | 0.9824 | 0.9823 | 0.9356 | 0.9363 |
| DT | 0.9766 | 0.9596 | 0.9766 | 0.9767 | 0.9767 | 0.9164 | 0.9165 |
| GBC | 0.9690 | 0.9948 | 0.9690 | 0.9693 | 0.9691 | 0.8896 | 0.8897 |
| ADA | 0.9636 | 0.9931 | 0.9636 | 0.9638 | 0.9636 | 0.8699 | 0.8700 |
| KNN | 0.9553 | 0.9618 | 0.9553 | 0.9547 | 0.9549 | 0.8369 | 0.8372 |
| LDA | 0.9307 | 0.9723 | 0.9307 | 0.9347 | 0.9322 | 0.7624 | 0.7641 |
| Ridge | 0.9268 | 0.9723 | 0.9268 | 0.9242 | 0.9247 | 0.7228 | 0.7255 |
| LR | 0.8813 | 0.9358 | 0.8813 | 0.8731 | 0.8753 | 0.5336 | 0.5392 |
| QDA | 0.7910 | 0.9506 | 0.7910 | 0.8941 | 0.8154 | 0.4843 | 0.5480 |
| Dummy | 0.8326 | 0.5000 | 0.8326 | 0.6932 | 0.7565 | 0.0000 | 0.0000 |
| SVM-L | 0.7914 | 0.5236 | 0.7914 | 0.7300 | 0.7467 | 0.0741 | 0.0859 |
| NB | 0.6122 | 0.9360 | 0.6122 | 0.8778 | 0.6575 | 0.2735 | 0.3917 |

Tabulka G.38: Výsledky pro subset dns – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.9147 | 0.9131 | 0.9147 | 0.9113 | 0.9121 | 0.6822 | 0.6851 |
| XGB | 0.9167 | 0.9197 | 0.9167 | 0.9133 | 0.9116 | 0.6734 | 0.6841 |
| ET | 0.9135 | 0.8937 | 0.9135 | 0.9101 | 0.9110 | 0.6785 | 0.6811 |
| KNN | 0.9130 | 0.8833 | 0.9130 | 0.9099 | 0.9108 | 0.6793 | 0.6812 |
| LGBM | 0.9156 | 0.9190 | 0.9156 | 0.9124 | 0.9095 | 0.6639 | 0.6778 |
| DT | 0.8955 | 0.8544 | 0.8955 | 0.8947 | 0.8951 | 0.6293 | 0.6295 |
| GBC | 0.8865 | 0.8788 | 0.8865 | 0.8859 | 0.8678 | 0.4883 | 0.5406 |
| ADA | 0.8589 | 0.8436 | 0.8589 | 0.8482 | 0.8276 | 0.3228 | 0.3878 |
| LDA | 0.8377 | 0.7780 | 0.8377 | 0.8076 | 0.7922 | 0.1753 | 0.2379 |
| Ridge | 0.8303 | 0.7779 | 0.8303 | 0.7890 | 0.7646 | 0.0570 | 0.1211 |
| SVM-L | 0.7879 | 0.6197 | 0.7879 | 0.7446 | 0.7574 | 0.0754 | 0.0868 |
| LR | 0.8245 | 0.6877 | 0.8245 | 0.7162 | 0.7531 | 0.0110 | 0.0211 |
| Dummy | 0.8284 | 0.5000 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| QDA | 0.5479 | 0.7888 | 0.5479 | 0.8435 | 0.5946 | 0.1957 | 0.2962 |
| NB | 0.2225 | 0.6425 | 0.2225 | 0.7940 | 0.1542 | 0.0143 | 0.0617 |

Tabulka G.39: Výsledky pro subset geo – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.8439 | 0.8181 | 0.8439 | 0.8185 | 0.8096 | 0.2522 | 0.3019 |
| XGB | 0.8458 | 0.8266 | 0.8458 | 0.8236 | 0.8091 | 0.2476 | 0.3065 |
| ET | 0.8436 | 0.8159 | 0.8436 | 0.8180 | 0.8090 | 0.2497 | 0.2996 |
| DT | 0.8420 | 0.8137 | 0.8420 | 0.8148 | 0.8079 | 0.2462 | 0.2929 |
| LGBM | 0.8458 | 0.8281 | 0.8458 | 0.8242 | 0.8077 | 0.2408 | 0.3031 |
| GBC | 0.8438 | 0.8135 | 0.8438 | 0.8236 | 0.8004 | 0.2079 | 0.2806 |
| KNN | 0.8409 | 0.7643 | 0.8409 | 0.8169 | 0.7966 | 0.1929 | 0.2615 |
| ADA | 0.8330 | 0.7953 | 0.8330 | 0.8039 | 0.7707 | 0.0824 | 0.1579 |
| Ridge | 0.8283 | 0.6889 | 0.8283 | 0.6862 | 0.7506 | -0.0001 | -0.0010 |
| LDA | 0.8283 | 0.6890 | 0.8283 | 0.6862 | 0.7506 | -0.0001 | -0.0010 |
| Dummy | 0.8284 | 0.5000 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| LR | 0.8278 | 0.7014 | 0.8278 | 0.6861 | 0.7503 | -0.0013 | -0.0111 |
| SVM-L | 0.7767 | 0.5732 | 0.7767 | 0.7127 | 0.7364 | 0.0094 | 0.0024 |
| QDA | 0.4146 | 0.7299 | 0.4146 | 0.8480 | 0.4426 | 0.1148 | 0.2312 |
| NB | 0.3421 | 0.6363 | 0.3421 | 0.7805 | 0.3584 | 0.0455 | 0.1018 |

Tabulka G.40: Výsledky pro subset html+lex – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9881 | 0.9984 | 0.9881 | 0.9880 | 0.9880 | 0.9567 | 0.9569 |
| LGBM | 0.9877 | 0.9983 | 0.9877 | 0.9877 | 0.9877 | 0.9555 | 0.9556 |
| RF | 0.9822 | 0.9957 | 0.9822 | 0.9822 | 0.9820 | 0.9345 | 0.9353 |
| GBC | 0.9806 | 0.9967 | 0.9806 | 0.9805 | 0.9805 | 0.9296 | 0.9298 |
| KNN | 0.9804 | 0.9842 | 0.9804 | 0.9802 | 0.9802 | 0.9285 | 0.9288 |
| ADA | 0.9774 | 0.9960 | 0.9774 | 0.9774 | 0.9774 | 0.9188 | 0.9188 |
| DT | 0.9754 | 0.9569 | 0.9754 | 0.9754 | 0.9754 | 0.9118 | 0.9118 |
| ET | 0.9752 | 0.9918 | 0.9752 | 0.9754 | 0.9746 | 0.9066 | 0.9092 |
| LDA | 0.9589 | 0.9876 | 0.9589 | 0.9581 | 0.9580 | 0.8467 | 0.8482 |
| Ridge | 0.9523 | 0.9876 | 0.9523 | 0.9521 | 0.9502 | 0.8143 | 0.8213 |
| LR | 0.8910 | 0.9430 | 0.8910 | 0.8897 | 0.8903 | 0.6039 | 0.6040 |
| NB | 0.8503 | 0.9263 | 0.8503 | 0.8887 | 0.8619 | 0.5583 | 0.5792 |
| SVM-L | 0.8326 | 0.5714 | 0.8326 | 0.6932 | 0.7565 | 0.0000 | 0.0000 |
| Dummy | 0.8326 | 0.5000 | 0.8326 | 0.6932 | 0.7565 | 0.0000 | 0.0000 |
| QDA | 0.5143 | 0.9328 | 0.5143 | 0.8551 | 0.5581 | 0.1775 | 0.2901 |

Tabulka G.41: Výsledky pro subset html – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.8381 | 0.7041 | 0.8381 | 0.8233 | 0.7807 | 0.1231 | 0.2156 |
| XGB | 0.8390 | 0.7062 | 0.8390 | 0.8311 | 0.7807 | 0.1226 | 0.2228 |
| ET | 0.8378 | 0.6982 | 0.8378 | 0.8215 | 0.7805 | 0.1222 | 0.2130 |
| LGBM | 0.8395 | 0.7060 | 0.8395 | 0.8416 | 0.7796 | 0.1177 | 0.2278 |
| KNN | 0.8387 | 0.6789 | 0.8387 | 0.8331 | 0.7795 | 0.1175 | 0.2198 |
| GBC | 0.8400 | 0.7057 | 0.8400 | 0.8542 | 0.7785 | 0.1128 | 0.2339 |
| ADA | 0.8382 | 0.6934 | 0.8382 | 0.8457 | 0.7755 | 0.1009 | 0.2142 |
| DT | 0.8275 | 0.6851 | 0.8275 | 0.7779 | 0.7746 | 0.1023 | 0.1495 |
| LR | 0.8283 | 0.6357 | 0.8283 | 0.7398 | 0.7510 | 0.0013 | 0.0119 |
| Dummy | 0.8284 | 0.5000 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| Ridge | 0.8281 | 0.6544 | 0.8281 | 0.6862 | 0.7505 | -0.0006 | -0.0062 |
| LDA | 0.8280 | 0.6549 | 0.8280 | 0.6862 | 0.7504 | -0.0008 | -0.0080 |
| SVM-L | 0.7315 | 0.6485 | 0.7315 | 0.7294 | 0.6955 | 0.0425 | 0.0586 |
| NB | 0.4572 | 0.6523 | 0.4572 | 0.8284 | 0.4981 | 0.1262 | 0.2236 |
| QDA | 0.3864 | 0.6566 | 0.3864 | 0.8380 | 0.4078 | 0.0944 | 0.2015 |

Tabulka G.42: Výsledky pro subset ip – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.8647 | 0.8757 | 0.8647 | 0.8501 | 0.8488 | 0.4252 | 0.4483 |
| LGBM | 0.8636 | 0.8761 | 0.8636 | 0.8489 | 0.8457 | 0.4097 | 0.4380 |
| KNN | 0.8486 | 0.8166 | 0.8486 | 0.8374 | 0.8395 | 0.4099 | 0.4188 |
| RF | 0.8367 | 0.8196 | 0.8367 | 0.8233 | 0.8283 | 0.3702 | 0.3743 |
| GBC | 0.8556 | 0.8549 | 0.8556 | 0.8387 | 0.8280 | 0.3292 | 0.3772 |
| ET | 0.8362 | 0.7580 | 0.8362 | 0.8220 | 0.8273 | 0.3649 | 0.3695 |
| DT | 0.8316 | 0.7130 | 0.8316 | 0.8162 | 0.8219 | 0.3439 | 0.3487 |
| ADA | 0.8344 | 0.8308 | 0.8344 | 0.8026 | 0.7826 | 0.1341 | 0.2009 |
| SVM-L | 0.8161 | 0.6753 | 0.8161 | 0.7817 | 0.7745 | 0.1334 | 0.1739 |
| LDA | 0.8285 | 0.7467 | 0.8285 | 0.7862 | 0.7515 | 0.0033 | 0.0270 |
| LR | 0.8280 | 0.7549 | 0.8280 | 0.7410 | 0.7513 | 0.0026 | 0.0151 |
| Ridge | 0.8284 | 0.7466 | 0.8284 | 0.7034 | 0.7507 | 0.0002 | 0.0028 |
| Dummy | 0.8284 | 0.5000 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| QDA | 0.5397 | 0.7390 | 0.5397 | 0.8467 | 0.5864 | 0.1925 | 0.2978 |
| NB | 0.5092 | 0.6737 | 0.5092 | 0.8299 | 0.5563 | 0.1576 | 0.2517 |

Tabulka G.43: Výsledky pro subset lex – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| LGBM | 0.9334 | 0.9336 | 0.9334 | 0.9318 | 0.9298 | 0.7418 | 0.7510 |
| XGB | 0.9323 | 0.9312 | 0.9323 | 0.9302 | 0.9291 | 0.7408 | 0.7476 |
| RF | 0.9186 | 0.9144 | 0.9186 | 0.9156 | 0.9132 | 0.6784 | 0.6907 |
| GBC | 0.9198 | 0.9142 | 0.9198 | 0.9189 | 0.9128 | 0.6731 | 0.6935 |
| ET | 0.9167 | 0.8756 | 0.9167 | 0.9136 | 0.9109 | 0.6693 | 0.6825 |
| ADA | 0.9020 | 0.8944 | 0.9020 | 0.8970 | 0.8934 | 0.6001 | 0.6187 |
| LDA | 0.8935 | 0.8790 | 0.8935 | 0.8901 | 0.8796 | 0.5399 | 0.5755 |
| DT | 0.8753 | 0.7957 | 0.8753 | 0.8765 | 0.8758 | 0.5650 | 0.5653 |
| Ridge | 0.8865 | 0.8790 | 0.8865 | 0.8873 | 0.8670 | 0.4838 | 0.5403 |
| KNN | 0.7235 | 0.7656 | 0.7235 | 0.8190 | 0.7508 | 0.2907 | 0.3250 |
| LR | 0.8284 | 0.4703 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| NB | 0.8284 | 0.5369 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| Dummy | 0.8284 | 0.5000 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| SVM-L | 0.5000 | 0.5103 | 0.5000 | 0.3579 | 0.4005 | 0.0000 | 0.0000 |
| QDA | 0.2269 | 0.8585 | 0.2269 | 0.8483 | 0.1575 | 0.0228 | 0.1029 |

Tabulka G.44: Výsledky pro subset rdap – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| XGB | 0.9538 | 0.9588 | 0.9538 | 0.9529 | 0.9527 | 0.8299 | 0.8321 |
| LGBM | 0.9527 | 0.9570 | 0.9527 | 0.9518 | 0.9513 | 0.8240 | 0.8272 |
| RF | 0.9503 | 0.9550 | 0.9503 | 0.9492 | 0.9491 | 0.8172 | 0.8192 |
| ET | 0.9477 | 0.9352 | 0.9477 | 0.9465 | 0.9463 | 0.8067 | 0.8092 |
| KNN | 0.9355 | 0.9156 | 0.9355 | 0.9337 | 0.9341 | 0.7633 | 0.7650 |
| DT | 0.9344 | 0.9016 | 0.9344 | 0.9337 | 0.9340 | 0.7665 | 0.7667 |
| GBC | 0.9356 | 0.9368 | 0.9356 | 0.9340 | 0.9323 | 0.7518 | 0.7598 |
| ADA | 0.9175 | 0.9150 | 0.9175 | 0.9140 | 0.9129 | 0.6793 | 0.6884 |
| LDA | 0.8599 | 0.8505 | 0.8599 | 0.8445 | 0.8460 | 0.4195 | 0.4359 |
| Ridge | 0.8441 | 0.8504 | 0.8441 | 0.8228 | 0.8030 | 0.2199 | 0.2877 |
| NB | 0.8284 | 0.7081 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| Dummy | 0.8284 | 0.5000 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| LR | 0.7214 | 0.4535 | 0.7214 | 0.6905 | 0.7022 | -0.0843 | -0.0866 |
| QDA | 0.6033 | 0.8785 | 0.6033 | 0.8562 | 0.6490 | 0.2496 | 0.3511 |
| SVM-L | 0.4524 | 0.5007 | 0.4524 | 0.6719 | 0.4729 | -0.1222 | -0.1351 |

Tabulka G.45: Výsledky pro subset tls+geo+ip – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.9788 | 0.9960 | 0.9788 | 0.9788 | 0.9788 | 0.9238 | 0.9238 |
| XGB | 0.9786 | 0.9966 | 0.9786 | 0.9790 | 0.9788 | 0.9244 | 0.9246 |
| LGBM | 0.9783 | 0.9965 | 0.9783 | 0.9788 | 0.9785 | 0.9236 | 0.9238 |
| ET | 0.9775 | 0.9943 | 0.9775 | 0.9774 | 0.9774 | 0.9188 | 0.9188 |
| DT | 0.9713 | 0.9498 | 0.9713 | 0.9714 | 0.9713 | 0.8972 | 0.8972 |
| GBC | 0.9649 | 0.9920 | 0.9649 | 0.9665 | 0.9654 | 0.8780 | 0.8789 |
| ADA | 0.9620 | 0.9897 | 0.9620 | 0.9640 | 0.9627 | 0.8684 | 0.8697 |
| LDA | 0.9292 | 0.9703 | 0.9292 | 0.9377 | 0.9319 | 0.7660 | 0.7713 |
| Ridge | 0.9226 | 0.9703 | 0.9226 | 0.9206 | 0.9214 | 0.7137 | 0.7145 |
| KNN | 0.8963 | 0.9366 | 0.8963 | 0.9012 | 0.8983 | 0.6432 | 0.6444 |
| NB | 0.8244 | 0.9137 | 0.8244 | 0.9108 | 0.8438 | 0.5511 | 0.6121 |
| LR | 0.8570 | 0.9237 | 0.8570 | 0.8397 | 0.8430 | 0.3966 | 0.4106 |
| QDA | 0.8219 | 0.9576 | 0.8219 | 0.9079 | 0.8417 | 0.5451 | 0.6042 |
| Dummy | 0.8326 | 0.5000 | 0.8326 | 0.6932 | 0.7565 | 0.0000 | 0.0000 |
| SVM-L | 0.6495 | 0.7851 | 0.6495 | 0.7238 | 0.6123 | 0.0223 | 0.0296 |

Tabulka G.46: Výsledky pro subset tls – phishing

| Model | Acc | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| RF | 0.8354 | 0.6913 | 0.8354 | 0.8430 | 0.7689 | 0.0736 | 0.1813 |
| XGB | 0.8353 | 0.6921 | 0.8353 | 0.8421 | 0.7689 | 0.0737 | 0.1807 |
| ET | 0.8353 | 0.6914 | 0.8353 | 0.8420 | 0.7688 | 0.0735 | 0.1803 |
| LGBM | 0.8353 | 0.6924 | 0.8353 | 0.8414 | 0.7688 | 0.0734 | 0.1799 |
| DT | 0.8349 | 0.6911 | 0.8349 | 0.8371 | 0.7686 | 0.0727 | 0.1758 |
| GBC | 0.8345 | 0.6863 | 0.8345 | 0.8411 | 0.7667 | 0.0649 | 0.1688 |
| KNN | 0.8326 | 0.5477 | 0.8326 | 0.8181 | 0.7646 | 0.0565 | 0.1422 |
| ADA | 0.8331 | 0.6371 | 0.8331 | 0.8328 | 0.7637 | 0.0529 | 0.1473 |
| Ridge | 0.8315 | 0.6255 | 0.8315 | 0.8064 | 0.7629 | 0.0496 | 0.1254 |
| LDA | 0.8296 | 0.6255 | 0.8296 | 0.7852 | 0.7623 | 0.0476 | 0.1076 |
| Dummy | 0.8284 | 0.5000 | 0.8284 | 0.6862 | 0.7506 | 0.0000 | 0.0000 |
| LR | 0.7706 | 0.5758 | 0.7706 | 0.7311 | 0.7436 | 0.0472 | 0.0488 |
| NB | 0.7492 | 0.5492 | 0.7492 | 0.7381 | 0.7406 | 0.0693 | 0.0725 |
| QDA | 0.7728 | 0.4459 | 0.7728 | 0.7519 | 0.7039 | -0.0021 | 0.0093 |
| SVM-L | 0.6409 | 0.5159 | 0.6409 | 0.7132 | 0.6343 | 0.0384 | 0.0259 |