UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA
PRÍRODOVEDECKÁ FAKULTA

KLASIFIKÁCIA MALVÉRU POMOCOU NEURÓNOVÝCH SIETÍ

2022 **František KURIMSKÝ**

UNIVERZITA PAVLA JOZEFA ŠAFÁRIKA
PRÍRODOVEDECKÁ FAKULTA

KLASIFIKÁCIA MALVÉRU POMOCOU NEURÓNOVÝCH SIETÍ

BAKALÁRSKA PRÁCA

Študijný program:	Informatika
Pracovisko (katedra/ústav):	Ústav informatiky
Vedúci bakalárskej práce:	RNDr. JUDr. Pavol Sokol, PhD.
Konzultant bakalárskej práce: (nepovinný)	Mgr. Richard Staňa

Košice 2022 **František KURIMSKÝ**

Univerzita P. J. Šafárika v
Košiciach Prírodovedecká fakulta

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	František Kurimský
Študijný program:	informatika (jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor:	Informatika
Typ záverečnej práce:	Bakalárska práca
Jazyk záverečnej práce:	slovenský
Sekundárny jazyk:	anglický

Názov:	Klasifikácia malvéru pomocou neurónových sietí
Názov EN:	Classification of malware using neural networks
Cieľ:	1) Analyzovať prístupy k analýze malvéru a vizuálnej reprezentácii malvéru. 2) Porovnať dostupné prístupy ku klasifikácii malvéru prostredníctvom neurónových sietí. 3) Navrhnúť, implementovať a vyhodnotiť systém na klasifikáciu malvéru prostredníctvom neurónových sietí.
Literatúra:	1) Stamp M, Alazab M, Shalaginov A. Malware Analysis Using Artificial Intelligence and Deep Learning. Springer International Publishing AG;2021. 2) Saxe J, Sanders H. Malware Data Science: Attack Detection and Attribution. No Starch Press; 2018. 3) Le Q, Boydell O, Mac Namee B, Scanlon M. Deep learning at the shallow end: Malware classification for non-domain experts. Digital Investigation. 2018 Jul 1;26:S118-26.

Vedúci:	RNDr. JUDr. Pavol Sokol, PhD.
Konzultant:	Mgr. Richard Staňa
Ústav :	ÚINF - Ústav informatiky
Riaditeľ ústavu:	doc. RNDr. Ondrej Krídlo, PhD.

Dátum schválenia: 16.05.2022

Abstrakt v štátnom jazyku

V dnešnom svete je malvér považovaný za jednu z najväčších bezpečnostných hrozieb na internete. Každým rokom sa objavuje čoraz viac útokov s rôznymi druhmi malvéru. Existujú viaceré prístupy, ako tento problém riešiť pomocou neurónových sietí. Hlavným cieľom tejto práce je porovnať rôzne možnosti reprezentácie súborov do vizuálnej podoby a takisto použitie rôznych možnosti navrhnutia modelu neurónovej siete. Naším cieľom je zo získaných poznatkov navrhnúť vlastné riešenie klasifikácie malvéru pomocou neurónových sietí. Pre trénovanie modelu potrebujeme vhodný dataset. Pre našu prácu používame Malimg dataset a MalwareBazaar dataset, ktorý obsahuje 490716 vzoriek malvéru detekovaných v období od 24.02.2020 do 11.03.2022. V závere je spracovaná implementácia riešenia klasifikácie malvéru pomocou neurónových sietí.

Abstrakt v cudzom jazyku

In today's world, malware is considered as one of the biggest security threats on the internet. More and more attacks with different types of malware appear every year. There are several approaches for solving this problem using neural networks. The main goal of this thesis is to compare different visual representations of files and use different models of neural networks. With the acquired knowledge, our other goal is to design our own solution for malware classification using neural networks. We need a suitable dataset for training the model. In this thesis we use a "malware daily dataset", which contains 490716 malware samples detected in the period from 24.02.2020 to 11.3.2021. Finally, we implemented our solution for classification of malware files by using neural network.

Obsah

ZADANIE ZÁVEREČNEJ PRÁCE **2**
Obsah **5**
Zoznam ilustrácií **7**
Zoznam tabuliek **8**
Úvod **9**
1 Malvér a jeho analýza **11**
1.1 Microsoft Windows spustiteľné súbory 11
1.1.1 Hlavičky PE formátu 12
1.2 Obfuskovaný a zabalený malvér 14
1.3 Techniky analýzy malvéru 14
1.4 Základná statická analýza 14
1.4.1 Pokročilá statická analýza 15
1.4.2 Dynamická analýza 15
1.5 Klasifikácia malvéru 15
1.5.1 Základné rodiny malvéru 16
2 Vizuálna reprezentácia malvéru **18**
2.1 Greyscale 18
2.2 RGB 18
2.3 SimHash 20
3 Dataset **21**
3.1 Malimg dataset 21
3.2 MalwareBazaar dataset 22
3.2.1 Ohodnotenie pomocou HybridAnalysis 22
3.2.2 Ohodnotenie pomocou MalwareBazaar 23
4 Neurónové siete pre klasifikáciu malvéru **25**
4.1 Konvolučná neurónová sieť a jej vrstvy 25
4.2 ResNet50 26
4.3 Metriky úspešnosti neurónových sietí 27
4.4 Popis modelu 28
5 Dosiahnuté výsledky v klasifikácii malvéru **30**
5.1 Malimg dataset a konvolučná neurónová sieť 30
5.2 Malimg dataset a ResNet50 33

5.3 MalwareBaazar dataset a konvolučná sieť 35
Záver 37
Zoznam použitej literatúry 42
Prílohy 45

Zoznam ilustrácií

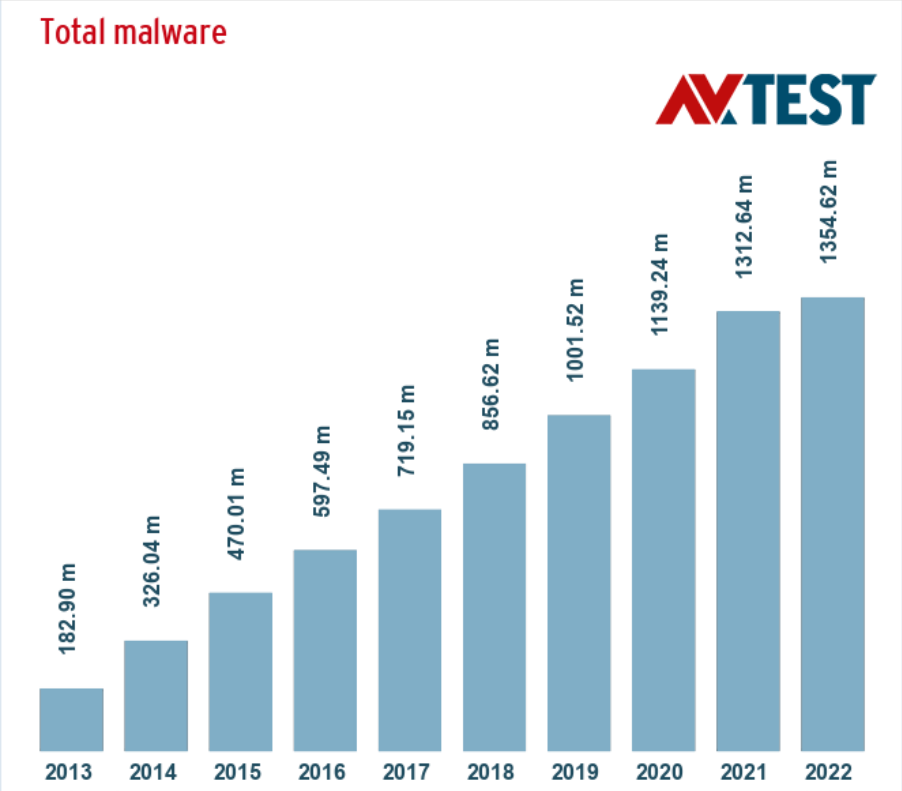
Obr. 1 Štatistika počtu detekovaného malvéru [1] 9
Obr. 2 Štruktúra spustiteľného súboru 12
Obr. 3 Prevod súboru do greyscale obrázka 18
Obr. 4 Greyscale vizuálna reprezentácia troch vzoriek malvéru 18
Obr. 5 Prevod súboru do RGB obrázka verzia 1 19
Obr. 6 Prevod súboru do RGB obrázka verzia 2 19
Obr. 7 RGB vizuálna reprezentácia troch vzoriek malvéru 19
Obr. 8 Vizuálna reprezentácia štyroch vzoriek malvéru pomocou Simhash algoritmu . 20
Obr. 9 Ukážka výpočtu dot produktu na konvolučnej vrstve 26
Obr. 10 Architektúra modelu konvolučnej neurónovej siete pre klasifikáciu malvéru .. 28
Obr. 11 Architektúra modelu konvolučnej neurónovej siete použitím ResNet architektúry 29
Obr. 12 Prehľad hodnoty validačnej presnosti počas trénovania modelu konvolučnej siete na Malimg datasete 30
Obr. 13 Hodnoty Precision, Recall a F1 skóre pre triedy Malimg datasetu na modeli konvolučnej siete 31
Obr. 14 Confusion matica modelu konvolučnej siete na Malimg datasete 32
Obr. 15 Hodnoty presnosti, validačnej presnosti a validačnej loss funkcie na Malimg datasete a ResNet50 modeli 33
Obr. 16 Hodnoty Precision, Recall a F1 skóre pre triedy Malimg datasetu na modeli Resnet50 33
Obr. 17 Confusion matica modelu ResNet50 siete na Malimg datasete 34
Obr. 18 Hodnoty presnosti, validačnej presnosti a validačnej loss funkcie na modeli konvolučnej siete natrénovanej na datasete aj ohodnotení od MalwareBaazar 35
Obr. 19 Hodnoty Precision, Recall a F1 skóre pre triedy MalwareBazaar datasetu na modeli 36
Obr. 20 Confusion matica modelu konvolučnej siete na MalwareBazaar datasete 36
Obr. 21 Hodnoty presnosti, validačnej presnosti a validačnej loss funkcie na ResNet50 modeli natrénovanej na datasete aj ohodnotení od MalwareBaazar 37
Obr. 22 Hodnoty Precision, Recall a F1 skóre pre triedy MalwareBazaar datasetu na ResNet50 modeli 37
Obr. 23 Confusion matica modelu ResNet50 na MalwareBazaar datasete 38

Zoznam tabuliek

Tab. 1 Počet vzoriek malvéru pre rodiny v Malimg datasete 21
Tab. 2 Počet vzoriek malvéru pre rodiny v MalwareBazaar datasete s ohodnotením pomocou Hybrid-analysis 22
Tab. 3 Počet vzoriek malvéru pre rodiny v MalwareBazaar dataset po ohodnotení službou MalwareBazaar 23
Tab. 4 hodnoty úspešnosti, presnosti a recall začiatok [30] 39
Tab. 5 hodnoty úspešnosti, presnosti a pokračovanie [30] 39

Úvod

Malvér je veľkým rizikom na internete. V poslednej dobe sledujeme rapídne narastajúce množstvo detekovaného malvéru. Spoločnosť AV-TEST denne registruje okolo 450000 nových vzoriek nebezpečných súborov. Podľa ich štatistiky, od roku 2013 narástlo množstvo detekovaných vzoriek malvéru o približne 1 miliardu. Obr. 1 ukazuje štatistiku počtu detekovaného malvéru od roku 2013 do dnes vytvorenú spoločnosťou AV-TEST [1].



Obr. 1 Štatistika počtu detekovaného malvéru [1]

Jedným z hlavných dôvodov tohto rapídneho nárastu je použitie obfuskačných metód a takisto praktík zabalenia súboru. Malvér developeri využívaju tieto techniky na modifikovanie malvéru tak, aby nebol jasný zámer a správanie daného malvéru. Tieto techniky umožňujú rýchlu úpravu už existujúceho malvéru, a teda vytváranie nového malvéru [2].

Vzorky malvéru vieme kategorizovať do takzvaných rodín. V každej rodine sa nachádzajú vzorky malvéru, ktorá sú si podobné správaním, a teda škodami alebo účinkami, ktoré spôsobujú napadaným programom, alebo operačným systémom. Kvôli veľkému množstvu nového detekovaného malvéru, a spoločným črtám v rodinách

malvéru chceme overiť nový prístup klasifikácie malvéru, a to použitím neurónových sietí.

Naším cieľom je porovnať rôzne prístupy k analýze malvéru, a teda bližšie popísať metódy statickej a dynamickej analýzy nebezpečných súborov. V druhom rade porovnávame rôzne metódy vizualizácie malvéru do obrázkovej formy. Porovnávame takisto rôzne spôsoby využitia neurónových sietí, a viaceré kombinácie implementácie modelu neurónovej siete s vizualizáciou. V poslednom kroku je naším cieľom navrhnúť a implementovať vlastný systém klasifikácie malvéru prostredníctvom neurónových sietí.

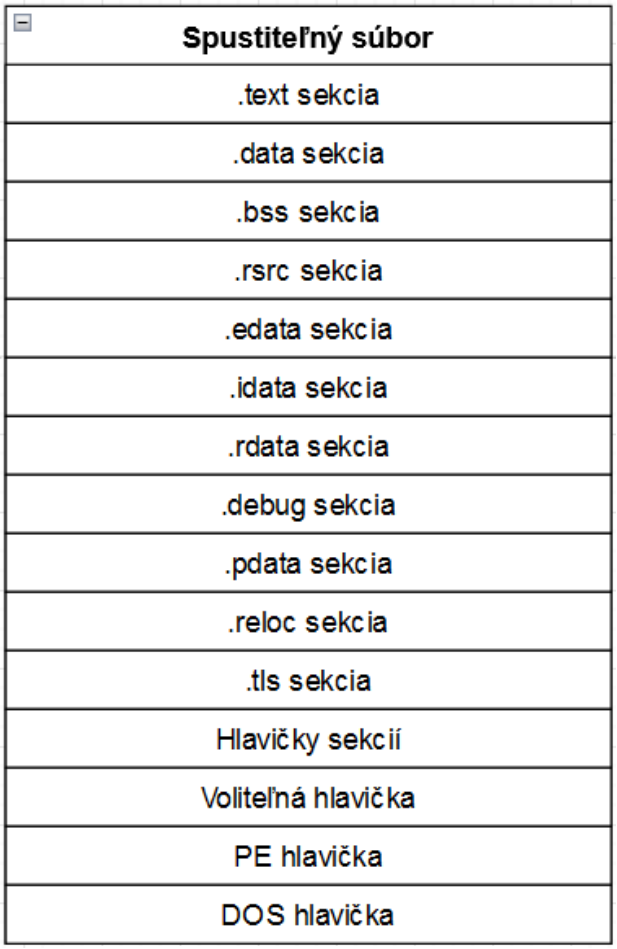
1 Malvér a jeho analýza

Slovo malvér vzniklo spojením slov malicious a software [2]. Všeobecne malvér definujeme ako pridaný, odstránený alebo zmenený kód v systéme tak, aby znemožnil správnu a očakávanú funkcionalitu napádaného programu (napr. aj operačného systému) [3].

Analýza malvéru je veľmi potrebná v dnešnom svete. Jej cieľom je pochopiť, ako funguje malvér, ako ho je možné identifikovať, poraziť a odstrániť. To zahŕňa zistenie, čo konkrétna vzorka malvéru spôsobuje, ako ho vieme odhaliť, a odhalenie rozsahu škôd ktoré napáchal. Vzhľadom na veľké množstvo malvéru je v dnešnom svete veľký dopyt po malvér analytikoch. Analýza malvéru je dôležitá pre každého, kto pracuje s bezpečnostnými incidentami počítačov [4]. V tejto časti je bližšie popísaný formát Windows spustiteľných súborov, metódy obfuskácie a zabalenia súboru a konkrétne typy analýz malvéru.

1.1 Microsoft Windows spustiteľné súbory

Pre statickú analýzu je potrebné poznať format spustiteľného súboru. Spustiteľný súbor obsahuje informácie potrebné na to, aby operačný súbor vedel ako spracovať súbor a ako ho namapovať do pamäte [4]. Takisto dodáva média alebo zdroje, ktoré sú použité pri behu program a poskytuje bezpečnostné údaje, ako sú podpisy digitálneho kódu. Tieto podpisy Windows využíva ako overenie toho, že kód pochádza z dôveryhodného zdroja. Takýto typ súboru obsahuje viaceré hlavičky, ktoré zabezpečujú správne namapovanie do pamäte. Takisto obsahuje viaceré sekcie, ktoré obsahujú dáta pre spustenie programu [5]. Obr. 2 popisuje štruktúru spustiteľného súboru.



Obr. 2 Štruktúra spustiteľného súboru

1.1.1 Hlavičky PE formátu

PE hlavička nám hovorí o tom, či je program určený pre 32-bitovú architektúru, alebo 64-bitovú architektúru. Takisto definuje všeobecné atribúty programu, ako napríklad binárny kód, obrázky a komprimované údaje. Hlavička poskytuje základné informácie, inak povedané metadáta o časovej pečiatke a o počte sekcií [5], [6].

Voliteľná hlavička je v skutočnosti prítomná v dnešných spustiteľných súboroch. Táto hlavička definuje umiestnenie vstupného bodu program, ktorý odkazuje na prvý príkaz vykonaný pri behu programu. Takisto definuje veľkosť údajov, ktoré operačný systém načíta do pamäte. Informácia o vstupnom bode programu je veľmi užitočná pre analytikov malvéru [5].

Hlavičky sekcií sú umiestnené hneď za voliteľnou hlavičkou. Každá jedna hlavička konkrétnej sekcie pozostáva z 40 bajtov [7]. Spustiteľný súbor je rozdelený na sekcie. Sekcia v spustiteľnom súbore predstavuje dáta, ktoré budú namapované do pamäte, počas načítania súboru operačným systémom, alebo obsahuje inštrukcie o tom, ako má byť spustiteľný súbor načítaný do pamäte. Hlavičky sekcií takisto obsahujú informáciu o tom, aké práva sú udelené sekciám, a teda či konkrátnu sekciu je možná čítať, upravovať, alebo či sekcia má byť spustená počas behu programu [5].

V nasledujúcej časti bližšie popíšeme najdôležitejšie sekcie spustiteľných súborov. Zvyčajne sa jedná o .text, .data, .bss, .rsrc, .edata, .idata, .rdata, .debug, .pdata, .reloc a .tls [5]–[7].

- .text sekcia obsahuje spustiteľný kód programu, a takisto obsahuje vstupný bod programu.
- .data sekcia obsahuje globálne dáta potrebné pre beh programu.
- .bss sekcia je použitá na deklarovanie premenných a obsahuje neinicializované údaje.
- .rsrc sekcia obsahuje všetky zdroje programu, ako napríklad ikony, obrázky a stringy.
- .edata sekcia obsahuje tabuľku exportovaných dát. Exportované dáta sú napríklad funkcie programu, ktoré môžu byť použité iným programom.
- .idata sekcia obsahuje tabuľku importov. Program môže využívať funkcie z externých zdrojov, a preto je potrebné tieto funkcie po spustení program importovať.
- .rdata sekcia obsahuje informácie iba na čítanie, ako sú stringy, konštanty a informácie o debugovacích adresároch.
- .debug sekcia obsahuje debugovacie dáta. Ako sme vyššie spomenuli, debugovacie adresáre sa nachádzajú v .rdata sekcii. Každý debugovací adresár odkazuje na debug informácie v .debug sekcii.
- .pdata sekcia pozostáva z poľa funkcií používaných pri výskyte výnimiek v programe.
- .reloc sekcia umožňuje zmenu miesta kódu v pamäti. Táto sekcia pozostáva z tabuľky, ktorá operačnému systému poskytuje informácie o tom, na ktoré

miesto v pamäti bola presunutá daná informácia, a to použitím hodnoty offset, teda posun z pôvodného miesta v pamäti.

- .tls sekcia poskytuje podporu pre lokálne ukladanie statického vlákna. Umožňuje každému vláknu ukladať vlastné lokálne hodnoty pre definované premenné.

1.2 Obfuskovaný a zabalený malvér

Tvorcovia malvéru často pre tvorenie nového malvéru používajú obfuskáciu a zabalenie malvéru. Obfuskácia je proces, kedy sa autor snaží skryť dôležité časti kódu, kritické slová, s cieľom, aby daný súbor bol ťažko pochopiteľný. Zabalený malvér je komprimovaný súbor. Pri spustený takého malvéru sa na začiatku spustí menší program, ktorý dekomprimuje zabalený malvér, a následne sa spustí škodlivý súbor. Obfuskované aj zabalené vzorky malvéru často obmedzujú vykonávanie statickej analýzy nebezpečných súborov.

1.3 Techniky analýzy malvéru

Väčšina vzoriek malvéru je vo forme spustiteľného súboru. Analýzu rozdeľujeme na statickú a dynamickú. Základným rozdielom týchto typov je, že pri statickej analýze vzorku malvéru nespúšťame, ale analyzujeme jeho kód a štruktúru s cieľom odhaliť jeho zámer. Pri dynamickej analýze vzorku malvéru spúšťame [4].

1.4 Základná statická analýza

Pri analýze potenciálneho malvéru je na začiatok dobré overiť, či tento súbor už nebol rozpoznaný. Služba VirusTotal alebo HybridAnalysis umožňuje nahranie súboru, a generuje súhrn výsledkov z viacerých antivírusových programov [8], [9]. Pre unikátne označenie malvéru sa používa hashovací algoritmus. Najčastejšie sa jedná o MD5 algoritmus, ale aj SHA-1 alebo SHA-256. Na overenie, či potenciálny malvér už bol identifikovaný, môžeme použiť nasledujúci postup. Vygeneruje sa hash z podozrivého súboru a následne sa nahrá na stránku VirusTotal alebo HybridAnalysis.

Hľadanie stringov v súbore je jednou z metód statickej analýzy. Program obsahuje string napríklad vtedy, keď vypisuje informáciu do konzoly, pripája sa na URL adresu, alebo kopíruje súbor na miesto v napádanom zariadení [4]. Napríklad stringy IP adries je možné zo súboru získať nasledujúcim príkazom :

```
$ strings example.exe | sed -n 's/.*\([0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\).*/\1/1p' .
```

Ďalším zo spôsobov statickej analýzy spustiteľného súboru je použitie python knižnice pefile, ktorej autorom je Ero Carerra [10]. Táto knižnica umožňuje zo spustiteľných súborov získať informácie o súbore, ako napríklad namapovanie konkrétnych sekcií v pamäti. Takisto pomocou nej je možné získať informácie o použitých funkciách v kóde súboru [5].

1.4.1 Pokročilá statická analýza

Pre analýzu škodlivého súboru často nepostačuje základná statická analýza, teda informácie o stringoch, sekciách, importovaných a použitých funkciách nie sú dostatočné. Preto sa ďalej využíva metóda dissasmeblovania, t.j. rozobrania súboru a prekladu jeho binárneho kódu do jazyka assembler. Kód v jazyku assembler je ľudsky čitateľná reprezentácia binárneho kódu nebezpečného súboru. Pomocou tejto metódy je oveľa jednoduchšie pochopiť, čo sa malvér snaží vykonať [5].

1.4.2 Dynamická analýza

Statická analýza poskytuje užitočné informácie o analyzovanej vzorke malvéru, avšak neumožňuje sledovanie jeho správanie. Dynamická analýza spočíva na princípe spustenia malvéru v bezpečnom prostredí, ktoré sa nazýva sandbox, so zámerom sledovať jeho správanie. Príkladom takýchto sandboxov je napríklad Cuckoo sandbox, alebo ANY.RUN [11], [12]. Týmto spôsobom je možné sa vyhnúť problémom, ktorým sa čelí pri statickej analýze, ak je súbor obfuskovaný alebo zabalený. Dynamická analýza takto umožňuje sledovať zmeny, ktoré spustený malvér vykonáva v operačnom systéme a na jeho disku. Vďaka tomu je možné jednoduchšie kategorizovať malvér vzorky do kategórií, podľa spoločných čŕt v správaní. Najdôležitejším prínosom dynamickej analýzy je možnosť vytvorenia detektorov malvéru na základe strojového učenia. Trénovanie takýchto detektorov spočíva napríklad v použití systémových logov zo vzoriek malvéru, ale aj z bezpečných vzoriek, aby model vedel rozlíšiť medzi týmito typmi súborov [5].

1.5 Klasifikácia malvéru

Pri analýze malvéru je dôležitou časťou jeho klasifikácia. Klasifikácia malvéru umožňuje pozorovať, či konkrétna vzorka malvéru má podobné charakteristické črty s už rozpoznanými vzorkami malvéru. Kryptografiké hashovacie algoritmy, ako sú MD5, SHA-1 alebo SHA-256 umožňujú porovnať identické vzorky malvéru, avšak malvér

developeri môžu zmeniť nepodstatné časti malvéru, a tým úplne zmenia hodnoty týchto hashovacích algoritmov [13]. V nasledujúcej časti sú uvedené metódy klasifikácie pomocou fuzzy hashu, import hashu, hashovania sekcií a pomocou YARA pravidiel.

Metóda fuzzy hashovania umožňuje porovnávať podobnosti v súboroch. Ssdeep je nástroj na generovanie fuzzy hashov pre súbory, a takisto umožňuje určiť percentuálnu zhodu fuzzy hashov medzi jednotlivými vzorkami súborov [14]. V pythone je možné získať fuzzy hashe súborov a takisto porovnať ich percentuálnu zhodu pomocou knižnice python-ssdeep [15].

Ďalšou metódou klasifikácie súborov je pomocou import hashu. Hodnota import hashu je vypočítaná na základe názvov importovaných funkcií a knižníc. Súbory, ktoré boli skompilované z rovnakého zdroja a na rovnakom princípe majú tendenciu mať rovnakú hodnotu import hashu. To znamená, že tabuľka importovaných adries v porovnávaných súboroch je rovnaká [13]. V pythone je možné získať hodnotu import hashu pomocou knižnice pefile [10].

Pri hľadaní podobností vo vzorkách malvéru sa zvykne takisto používať princíp, kedy sa pre každú sekciu spustiteľného súboru vytvorí MD5 hash. Prehľad týchto hashov v jazyku python sa získava pomocou knižnice pefile [10], [13].

Konkrétne rodiny malvéru sú odlíšiteľné na základe unikátnych stringov alebo binárnych ukazovateľov. YARA je flexibilný nástroj na klasifikáciu malvéru, ktorý umožňuje definovať pravidlá, ktoré detekujú reťazce, sekvencie pokynov a regulárne výrazy [16]. Pomocou týchto pravidiel vieme skenovať vzorku malvéru, a klasifikovať jej rodinu [17].

1.5.1 Základné rodiny malvéru

Informácie o rodinách sú interpretované zo zdroja [4].

Backdoor je škodlivý kód, ktorý sa sám inštaluje na počítač aby umožnil prístup útočníkovi. Tento typ malvéru umožňuje útočníkovi pripojiť sa na napádaný počítač, a vykonávať príkazy.

Botnet je podobný typu backdoor tým, že umožňuje útočníkovi sa pripojiť na napádaný systém, avšak všetky počítače napadnuté rovnakou vzorkou botnetu dostávajú inštrukcie z jedného servera.

Downloader je malvér, ktorého úlohou je stiahnuť na napádaný počítač ďalší malvér. Útočníci zvyčajne inštalujú tento typ malvéru hneď po tom, čo prvý krát získajú prístup do systému.

Rootkit je kód navrhnutý tak, aby zakrýval existenciu iného malvéru. Tento typ malvéru je zvyčajne používaný s malvérom z rodiny backdoor a zabezpečiť tak prístup do systému útočnikovi s cieľom sťažiť jeho odhalenie.

Ďalším typom malvéru je **information-stealing malvér.** Cieľom tohto malvéru je zbierať informácie na napadnutom systéme, ako napríklad prihlasovacie údaje, a posielať ich útočníkovi.

Launcher je škodlivý program, ktorého cieľom je spustenie iných malvérov. Na spustenie zvyčajne využíva netradičné spôsoby, aby sa zabezpečilo nenápadne spustenie a lepší prístup k systému.

Scareware je typ malvéru, ktorý sa snaží zastrašiť majiteľa napadnutého systému. Zvyčajne používa grafické rozhranie systému a oznamuje používateľovi, že jeho počítač bol napadnutý a jediný spôsob, ako odstrániť malvér je zakúpením ich softvéru.

Worm or virus je typ malvéru, ktorý sa dokáže sám nakopírovať na ďalšie počítače.

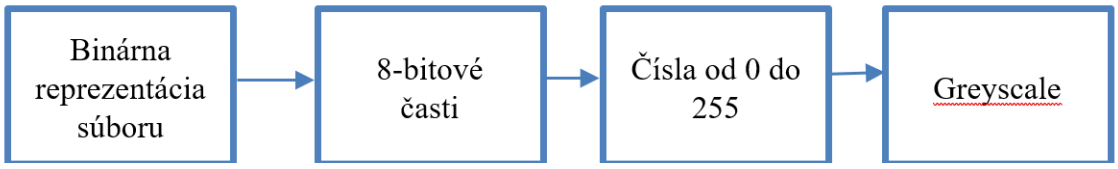
Cieľom **spam-sending** malvéru je infikovať počítač a následne ho použiť na rozposielanie spamových správ. Týmto spôsobom si útočníci zarábajú poskytovaním služieb rozposielania spamu.

2 Vizuálna reprezentácia malvéru

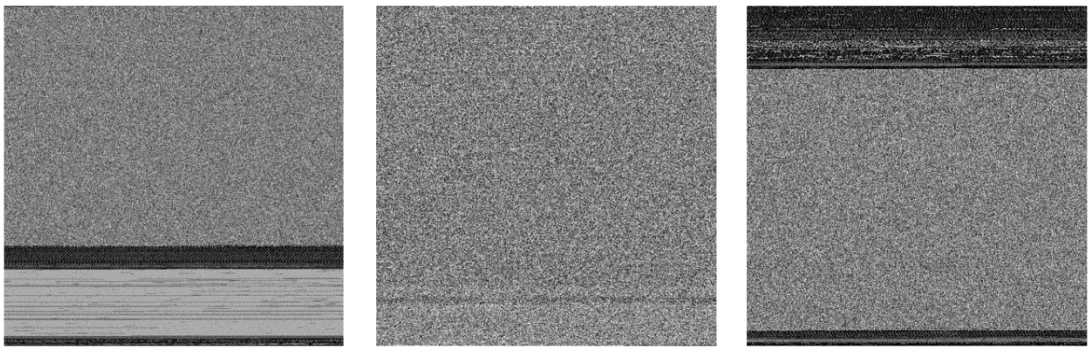
Pre použitie neurónových sietí je dobré prezentovať dáta do formátu, na ktorom sa model vie naučiť klasifikovať jednotlivé vzorky súborov do príslušných rodín alebo inak povedané do tried. Klasifikácia obrázkov do tried je overená technológia. Na prezentáciu súborov v obrázkovej forme môžeme použiť viaceré techniky.

2.1 Greyscale

Jedným z typov vizualizácie je prevod súboru do greyscale obrázku, inak povedané do obrázku v odtieňoch sivej. Túto metódu použil v svojej práci Kalash. Binárnu formu súboru je rozdelíme na 8-bitové časti. Každá z týchto častí reprezentuje číslo od 0 do 255. Následne tieto čísla môžeme považovať za reprezentantov jednotlivých pixelov. Pole takýchto čísel teda prevedieme na obrázok, ktorý je v odtieňoch sivej [18]. Obr. 3 ukazuje postup prevedenia súboru do greyscale obrázka a Obr. 4 ukazuje konkrétny príklad vizuálnej reprezentácie malvéru do greyscale formy.



Obr. 3 Prevod súboru do greyscale obrázka

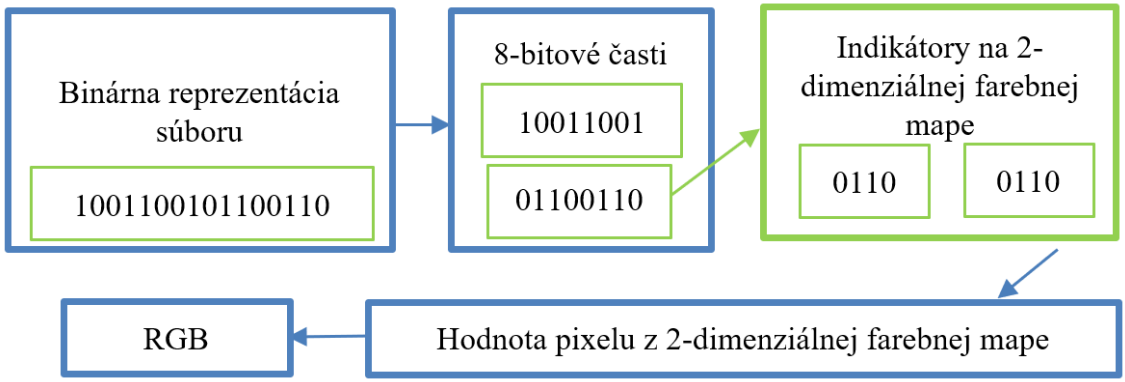


Obr. 4 Greyscale vizuálna reprezentácia troch vzoriek malvéru

2.2 RGB

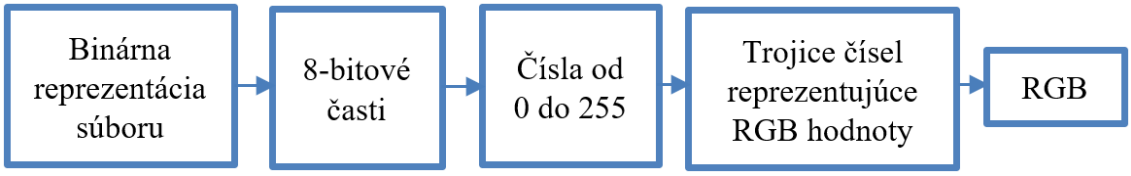
Reprezentácia súborov do RGB obrázkov, inak povedané do farebných obrázkov, je podobná logike prevodu na greyscale. Súbor prevedieme na jeho binárnu formu. Následne ho rozdelíme do častí po 8 bitoch. Tieto časti následne rozdelíme na dve

podskupiny, ktoré budú indikátormi farebnej hodnoty v 2-dimenziálnej farebnej mape [19]. Obr. 5 načrtáva postup prevodu súboru do RGB obrázka.

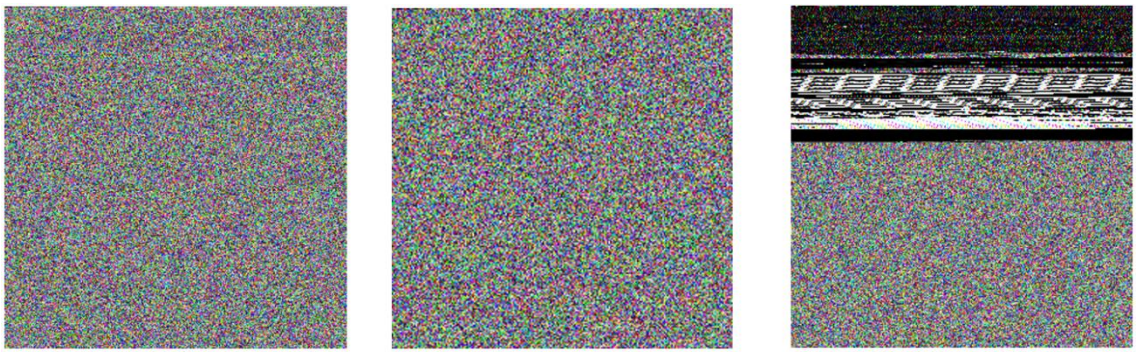


Obr. 5 Prevod súboru do RGB obrázka verzia 1

Poznáme aj iný prístup k reprezentácií súborov do RGB obrázka. Binárnu formu súboru rozdelíme do častí po 8 bitoch. Tieto bity vieme prezentovať do čísel v intervale od 0 po 255. Následne vezmeme trojice týchto čísel, kde každá prezentuje jednu časť RGB hodnoty pre pixel. Obr. 6 ukazuje postup tohto prístupu a Obr. 7 ukazuje konkrétny príklad vizuálnej reprezentácie malvéru do RGB formy.



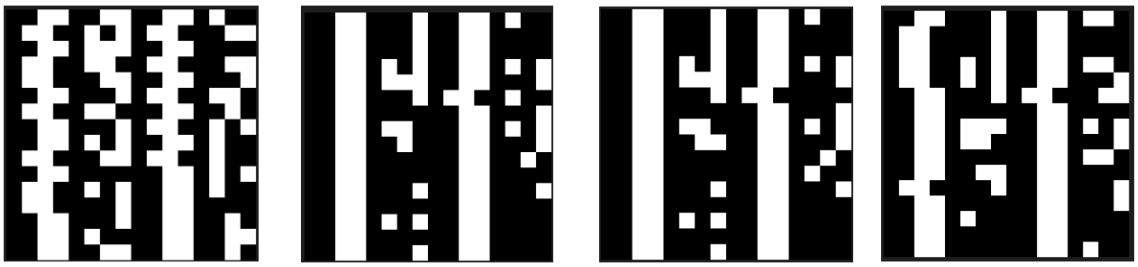
Obr. 6 Prevod súboru do RGB obrázka verzia 2



Obr. 7 RGB vizuálna reprezentácia troch vzoriek malvéru

2.3 SimHash

Hashovací algoritmus Simhash slúži na porovnávanie podobností v súboroch. Táto metóda spočíva na získaní inštrukcií jazyka assembler zo spustiteľného súboru, napríklad použitím príkazu objdump v operačnom systéme Linux [20]. Následne pre každú inštrukciu je potrebné vytvoriť hash, napríklad použitím hashovacích algoritmov MD5 alebo SHA256, a tieto hashe si uložiť do matice. Hodnoty hashov v ďalšom kroku prevediem na ich binárnu reprezentáciu. Následne každú hodnotu týchto binárnych reprezentácií nahradíme tak, že hodnotu 1 prevedieme na 1, a hodnotu 0 na -1. Ďalej spočítame tieto hodnoty v matici po stĺpcoch, čím dostaneme výsledný vektor. V poslednom kroku, každú hodnotu, ktorá je väčšia alebo rovná 0, nahradíme hodnotou 255 a každú menšiu ako 0 nahradíme 0. Takto získané pole vieme následne previesť do obrázka [21]. Ukazuje príklad reprezentácie malvéru do obrázka použitím hashovacieho algoritmu Simhash a algoritmu MD5.



Obr. 8 Vizuálna reprezentácia štyroch vzoriek malvéru pomocou Simhash algoritmu

3 Dataset

Pre trénovanie neurónovej siete je potrebné nájsť dataset. Pri hľadaní vhodného datasetu sme našli viaceré. Jedná sa o Microsoft Kaggle competition dataset [22], Malimg dataset [23], a MalwareBazaar dataset [24]. Microsoft Kaggle competition dataset je dataset pripravený spoločnosťou Microsoft v roku 2015. V tomto roku otvorila súťaž zameranú na riešenie problému klasifikácia malvéru pomocou neurónových sietí.

3.1 Malimg dataset

Malimg dataset obsahuje vzorky malvéru reprezentované v greyscale forme. Tento dataset obsahuje malvér z 25 rodín. Celkový počet vzoriek v tomto datasete je 9339. Malimg dataset nebolo potrebné ohodnocovať, keďže je optimalizovaný pre riešenie problému klasifikácie malvéru pomocou neurónových sietí, a teda už má ohodnotené jednotlivé vzorky. Tab. 1 Ukazuje názvy rodín a takisto počty vzoriek patriacich do konkrétnych rodín v Malimg datasete.

Tab. 1 Počet vzoriek malvéru pre rodiny v Malimg datasete

Rodina	Počet	Rodina	Počet
Yuner.A	800	Adialer.C	122
Lolyda.AA3	123	Swizzor.gen!I	132
Dialplatform.B	177	Rbot!gen	158
Swizzor.gen!E	128	Dontovo.A	162
Lolyda.AT	159	Alueron.gen!J	198
VB.AT	408	Obfuscator.AD	142
Lolyda.AA2	184	Lolyda.AA1	213
Malex.gen!J	136	Fakerean	381
C2LOP.P	146	Skintrim.N	80
Autorun.K	106	Allaple.L	1591
Wintrim.BX	97	Agent.FYI	116
Instantaccess	431	C2LOP.gen!g	200
Allaple.A	2949		

3.2 MalwareBazaar dataset

MalwareBazaar dataset vznikol z potreby malvér analytikov, a to kvôli obmedzeným službám na iných platformách, kde je napríklad obmedzené množstvo stiahnutí, alebo za používanie služby sú veľmi drahé poplatky. MalwareBazaar nie je len dataset, ale miesto, kde bezpečnostní analytici môžu zdieľať najnovšie hrozby, a to bez obmedzenia sťahovania súborov a bez poplatkov [24]. V čase stiahnutia tohto datasetu sme získali 490662 vzoriek malvéru. Tento dataset predstavuje najnovšie hrozby na internet a vzorky, ktoré sa v ňom nachádzajú boli detekované v období od 24.02.2020 do doby stiahnutia, teda 11.03.2022. Tento dataset bolo potrebné ohodnotiť, na čo sme využili službu HybridAnalysis [9] a takisto sme jednotlivé vzorky ohodnotili pomocou API služby, ktorú poskytuje samotný MalwareBazaar.

3.2.1 Ohodnotenie pomocou HybridAnalysis

Hybrid-analysis je internetová platforma, ktorá umožňuje nahrať súbor, ktorý potrebujeme analyzovať, a následne nám vráti zhrnutie výsledkov o správaní konkretného súboru. Táto služba má voľné prístupné API volania, ktoré umožňujú napríklad nahrať súbor. Pre nás bolo podstatné použiť API volanie, ktorým sme získali ohodnotenie a teda triedu konkrétnej vzorky na základe informácie SHA-256 hashu našej vzorky. Takto sme získali ohodnotenie nášho datasetu. Veľa vzoriek však Hybrid-analysis nepoznal, a niektoré triedy boli málo zastúpené v tomto datasete. Preto sme si vybrali najviac relevantné triedy. Takto sme dostali 48934 ohodnotených vzoriek malvéru rozdelených do 38 tried. Tab. 2 popisuje zastúpenie jednotlivých tried a počty vzoriek po ohodnotení službou Hybrid-analysis.

Tab. 2 Počet vzoriek malvéru pre rodiny v MalwareBazaar datasete s ohodnotením pomocou Hybrid-analysis

Rodina	Počet	Rodina	Počet
Trojan.Agent	11925	Trojan.DOC.Agent	773
XLM.Trojan.Abracadabra.28	4769	Fragtor.Generic	757
Razy.Generic	3659	Trojan.Injuke	858
Trojan.Emotet	3353	Valyria	1691

Zusy.Generic	2088	W97m.Downloader	561
X97M/Downloader.fk	2090	Ursu.Generic	553
Bulz.Generic	1862	Agent	545
Trojan.Delf.FareIt.Generic	653	MemScan:Trojan.Agent	535
Trojan.Linux.Mirai	1630	Graftor.Generic	520
VBA.Agent	1186	MSILHeracles.Generic	517
Trojan.Linux.Gafgyt.Generic	1011	MSILPerseus.Generic	493
Trojan.EmotetU.Generic	663	PonyStealer.Generic	480
Trojan.Zenpak	463	Virlock.Generic	362
Lazy.Generic	445	Trojan.NSISX.Spy.Generic	348
Trojan.Yakes	424	MSIL.Bladabindi.Generic	344
Trojan.MSOffice.Emotet	407	Trojan.Downloader	332
XF.Coeus	376	Jaik.Generic	329
Mal_GENISO	373	Mikey.Generic	309
IL:Trojan.MSILZilla	784	Strictor.Generic	466

3.2.2 Ohodnotenie pomocou MalwareBazaar

MalwareBazaar takisto poskytuje API volanie, ktoré nám umožňuje nahrať SHA-256 hash nášho súboru a získať zhrnutý report. Keďže dataset pochádza od tej istej služby, tak všetky vzorky boli službou rozpoznané, ale niektoré nemali ohodnotenie. Po tomto ohodnotení sme takisto mali niektoré triedy, ktorých zastúpenie bolo veľmi malé, preto sme vybrali prvých 20 najviac zastúpených tried a celkový počet vzoriek v týchto triedach bol 363562. Tab. 3 popisuje zastúpenie jednotlivých tried a počty vzoriek malvéru pre vybrané triedy po ohodnotením službou MalwareBazaar.

Tab. 3 Počet vzoriek malvéru pre rodiny v MalwareBazaar dataset po ohodnotení službou MalwareBazaar

Rodina	**Počet**	**Rodina**	**Počet**
Heodo	89724	NanoCore	5844
Quakbot	65169	Gozi	5829
AgentTesla	49181	GuLoader	5708

Dridex	36694	SnakeKeylogger	5538
Formbook	18333	RemcosRAT	5384
Mirai	14379	TrickBot	5113
Loki	10969	AveMariaRAT	5056
RedLineStealer	10759	IcedID	4250
SilentBuilder	9210	RaccoonStealer	4229
CobaltStrike	8179	Gafgyt	4014

4 Neurónové siete pre klasifikáciu malvéru

V tejto kapitole sú uvedené typy neurónových sietí, ktoré sme použili a aké metriky úspešnosti sme zvolili.

4.1 Konvolučná neurónová sieť a jej vrstvy

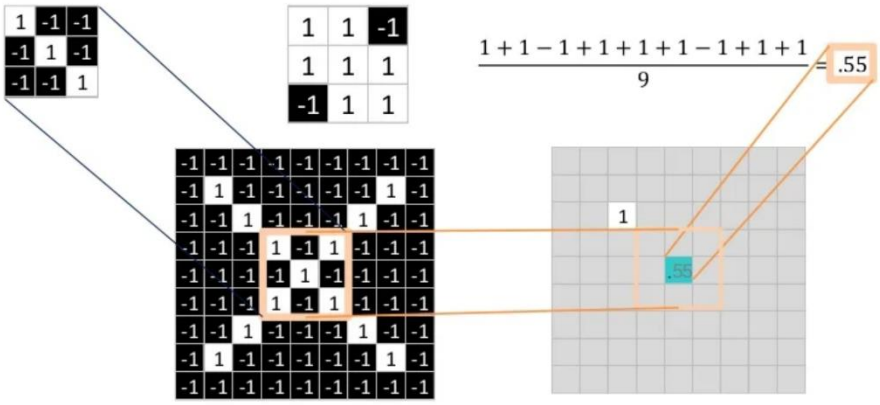
Konvolučné neurónové siete patria k metódam hĺbkového. Konvolučné neurónové siete pozostávajú z neurónov, ktoré majú váhy, ktoré sa v procese učenia prispôsobujú trénovacej množine. Tieto neurónové siete predpokladajú, že na vstupe majú obrázok, či už v greyscale forme alebo RGB forme. Na rozdiel základných neurónových sietí majú konvolučné neurónové siete usporiadané neuróny v konvolučnej vrstve v 3 rozmeroch, výška , šírka a hĺbka [26].

Pri tvorení konvolučných neurónových sietí sa používaju nasledujúce typy vrstiev. Vstupná vrstva je vrstva neurónov, ktoré obsahuje hodnoty získané z pixelov obrázka.

Konvolučná vrstva je vrstva, ktorá pozostáva z viacerých učenia schopných filtrov. Každý filter je definovaný výškou a šírkou, plus má takú istú hĺbku, ako aj vstupný obrázok. Počas dopredného chodu sa každý filter posúva cez celú výšku a šírku vstupného objemu, a vypočítava takzvaný dot produkt. Na Obr. 9 je znázornený výpočet dot produktu pre konkrétny filter [26]. Vzorec výpočtu na jednej vrstve je nasledujúci:

$$c_k^{(1)} = ReLU(\sum_j \psi_{j,k}^{(1)} * x_j + b_k^{(1)})$$

kde x je vstupná matica, ψ je filter, k predstavuje poradové číslo filtra, j je index hĺbky vstupnej matice, c_k predstavuje výstupné hodnoty pre filter s poradovým číslom k, b_k je váha prahového neurónu pre filter k a (1) vyjadruje, že ide o prvú konvolučnú vrstvu.



Obr. 9 Ukážka výpočtu dot produktu na konvolučnej vrstve

Funkciou pooling vrstvy je rázne zmenšovať priestorovú veľkosť reprezentácie, s cieľom zmenšiť veľké množstvo parametrov a potrebných výpočtov v sieti. Najčastejšie sa používa pooling vrstva s definovaným filtrom o veľkosti 2x2 a funkciou max. Výpočet na tejto vrstve je vykonávaný tak, že max funkcia vypočíta hodnotu pre malé plochy o veľkosti 2x2 pozdĺž výšky a šírky, a pre každú vrstvu hĺbky osobitne. Týmto spôsobom sa zahodí 75% aktivácií [26].

V rámci konvolučných neurónových sietí sú známe aj normalizačné a plne prepojené vrstvy. Úlohou normalizácie je získať výstupné hodnoty v rozsahu od 0 do 1.

Plne prepojené vrstvy predstavujú neuróny, ktoré majú plné prepojenie na aktivácie predchádzajúcej vrstvy [26].

Ďalšia vrstva ktorá sa využíva v konvolučných neurónových sietach je vrstva sploštenia. Úlohou tejto vrstvy je previesť matice ľubovoľnej hĺbky na vektor.

4.2 ResNet50

Pri tvorení väčších konvolučných neurónových sietí sa prišlo na to, že aj keď hlbšie neurónové siete môžu predstavovať komplexnejšie funkcie, a tým zvýšiť výkon modelu, tak bol odhalený problém znižúcej sa presnosti. Vinníkom tohto javu je efekt takzvaného miznúceho gradientu (angl. vanishing gradient). Pri spätnom šírení sa určujú hodnoty gradientu, posielajú sa vyššej vrstve a prepočítava sa hodnota váhy. Avšak pri hlbokých neurónových sieťach sa gradient zmenší natoľko, že aktualizácia vrchných vrstiev modelu je veľmi pomalá, ak vôbec ku zmene váhy dôjde.

ResNet siete ako prvé predstavili myšlienku prepojení preskočenia. Architektúra ResNet siete spočíva na 50 vrstvách, z toho 48 konvolučných vrsteiv a 2 pooling vrstvy, jedna s použitím funkcie max a druhá s použitím funkcie avg. Konvolučné vrstvy sú rozdelené do trojíc zvyškových blokov [27].

4.3 Metriky úspešnosti neurónových sietí

Pre získanie zobrazenie validácie natrénovaného modelu používame takzvanú confusion maticu. Táto matica zobrazuje to, ako model predikoval jednotlivé vzorky na testovacích dátach. Ukazuje počty vzoriek, ktoré boli predikované správne, a ktoré nesprávne. V rámci tejto matice rozpoznávame nižšie popísané 4 pojmy [28]. Pre správne pochopenie týchto pojmov si zaveďme kategóriu, ktorú označme napr. **K**.

- *TP* (true positiv): táto hodnota predstavuje pre kategóriu **K** počet vzoriek, ktoré patria tejto kategórii a boli správne predikované.
- *TN* (true negative): hodnota tohto termínu predstavuje počet vzoriek, ktoré nepatria do kategórie **K** a boli správne predikované.
- *FP* (false positiv): hodnote predstavuje počet vzoriek, ktoré nepatria do kategórie K, ale boli predikované ako členmi kategórie **K**.
- *FN* (false negative): hodnota predstavuje počet vzoriek, ktoré patria do kategórie **K** a boli predikované ako nejaká iná kategória.

Pomocou týchto hodnôt ďalej vieme definovať nasledujúce termíny, *Precision*, *Recall*, a *F1 skóre*. Hodnoty týchto termínov sú definované nasledujúcimi vzorcami.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ skóre = 2 * \frac{(Precision * Recall)}{Precision + Recall}$$

Tieto metriky vieme získať z predikcií natrénovaného modelu v pythone pomocou knižnice sklearn, a to nasledujúcimi príkazmi [29].

```
>>> from sklearn.metrics import classification_report
>>> print(classification_report(y_test,y_pred,target_names=labels))
```

4.4 Popis modelu

Model, na ktorom sme sa rozhodli trénovať klasifikáciu malvéru je založený na postupe tvorenia modelu použitom Kalashom v článku [18]. Rozhodli sme sa inšpirovať týmto modelom, aby sme overili jeho úspešnosť na Malimg datasete [23]. Model sa skladá z viacerých konvolučných vrstiev, pooling vrstiev založených na funkcii max, flatten vrstvy a v závere plne prepojených vrstiev. Na vstupe model očakáva maticu o rozmeroch 256x256x1. Prvou vrstvou modelu je konvolučná vrstva, ktorá obsahuje 64 filtrov, a na výstupe je matica o rozmeroch 254x254x64. Ďalšou vrstvou je pooling vrstva, ktorá pomocou max funkcie na výstupe dáva maticu o rozmeroch 127x127x64. Takto pokračuje stavba modelu, až na zmenu počtu filtrov v konvolučných vrstvách. Obr. 10 ukazuje architektúru vytvoreného modelu.

Layer (type)	Output Shape
conv2d_6 (Conv2D)	(None, 254, 254, 64)
max_pooling2d_5 (MaxPooling2	(None, 127, 127, 64)
conv2d_7 (Conv2D)	(None, 125, 125, 128)
max_pooling2d_6 (MaxPooling2	(None, 62, 62, 128)
conv2d_8 (Conv2D)	(None, 60, 60, 256)
max_pooling2d_7 (MaxPooling2	(None, 30, 30, 256)
conv2d_9 (Conv2D)	(None, 28, 28, 512)
max_pooling2d_8 (MaxPooling2	(None, 14, 14, 512)
conv2d_10 (Conv2D)	(None, 12, 12, 512)
max_pooling2d_9 (MaxPooling2	(None, 6, 6, 512)
conv2d_11 (Conv2D)	(None, 4, 4, 512)
flatten_1 (Flatten)	(None, 8192)
dense_2 (Dense)	(None, 4096)
dense_3 (Dense)	(None, 4096)
dense_4 (Dense)	(None, 20)

Obr. 10 Architektúra modelu konvolučnej neurónovej siete pre klasifikáciu malvéru

Ako je poukázané vo výsledkovej časti bakalárskej práce, úspešnosť tejto evalvácie nie je úplne najlepšia. Preto sme sa rozhodli vyskúšať natrénovať model aj na základe ResNet50 architektúry. Náš model na vstupe očakáva maticu s rozmermi 256x256x1. Na koniec nášho modelu sme ešte pridali 3 plne prepojené vrstvy. Obr. 11 ukazuje štruktúru modelu. Samotná ResNet architektúra bola popísaná vyššie.

```
Layer (type)                 Output Shape
=================================================
resnet50 (Functional)        (None, 2048)
_________________________________________________
flatten_1 (Flatten)          (None, 2048)
_________________________________________________
dense_3 (Dense)              (None, 2048)
_________________________________________________
dense_4 (Dense)              (None, 2048)
_________________________________________________
dense_5 (Dense)              (None, 20)
=================================================
```

Obr. 11 Architektúra modelu konvolučnej neurónovej siete použitím ResNet architektúry

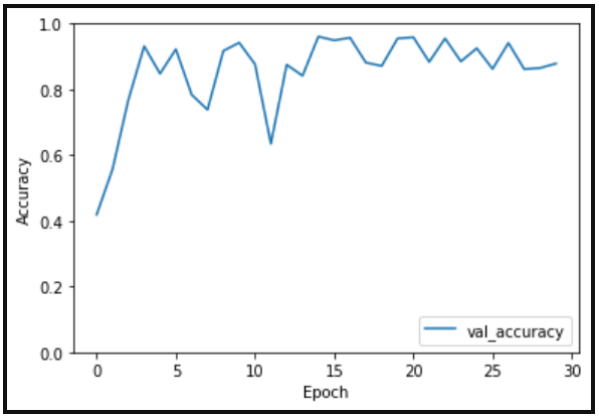
5 Dosiahnuté výsledky v klasifikácii malvéru

Nasledujúca časť obsahuje výsledky úspešnosti na vyššie uvedených datasetoch, v kombinácii s modelmi neurónovych sietí. Na trénovanie modelov sme použili platformu Tensorflow, ktorá obsahuje viaceré nástroje, knižnice, a zdroje od komunity developerov strojového učenia. Umožňuje jednoducho navrhnúť a implementovať strojové učenie v rôznych typoch aplikácií [25]. Modely sme trénovali na grafickej karte NVIDIA GeForce RTX 2080 Ti. Všetky modely sme trénovali na datasetoch, ktoré obsahujú grayscale obrázky, aby sme vedeli porovnať výsledky.

5.1 Malimg dataset a konvolučná neurónová sieť

Na začiatok sme sa rozhodli pre kombináciu Malimg datasetu s konvolučnou neurónovou sieťou uvedenou vyššie. Išlo o potvrdenie toho, že klasifikáciu malvéru má zmysel riešiť pomocou neurónových sietí. Táto časť bola inšpirovaná touto prácou [18].

Malimg sme si rozdelili tak, aby každá trieda mala 75 percentné zastúpenie v trénovacej sade, 15 percent vo validačnej sade a 15 percent v testovacej sade. Model sme trénovali na 30 epochách veľkosťami jednotlivých dávok (angl. batchsize) 128. Najlepší výsledok validačnej presnosti počas trénovania, ktorý tento model dosiahol je 96,07 percenta. Obr. 12 ukazuje hodnoty validačnej presnosti po každej epoche trénovania modelu.



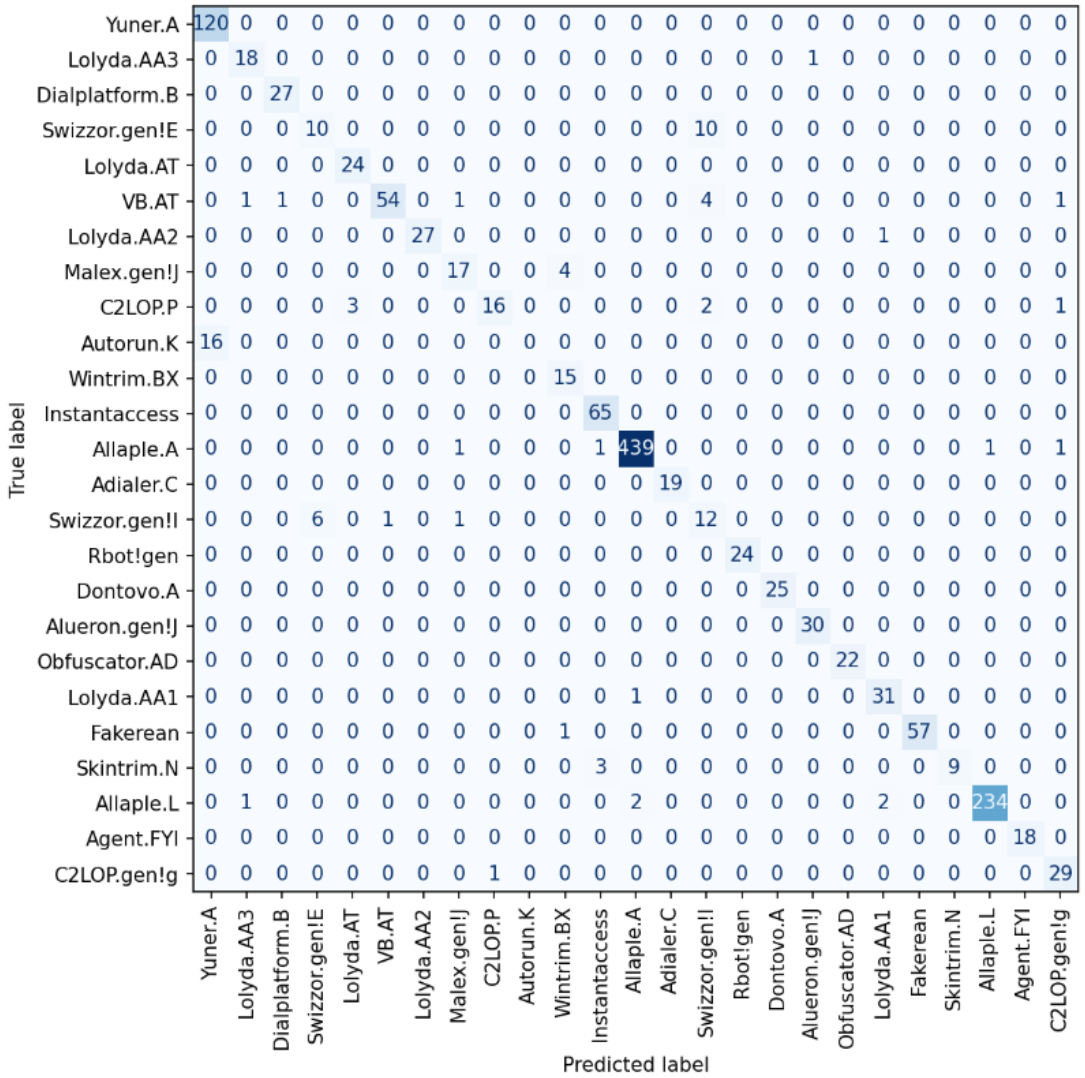
Obr. 12 Prehľad hodnoty validačnej presnosti počas trénovania modelu konvolučnej siete na Malimg datasete

Na testovacej sade mal model hodnotu presnosti 95,11 percent. Obr. 13 obsahuje hodnoty precision, recall a f1 skóre pre jednotlivé triedy na natrénovanom modely.

	precision	recall	f1-score	support
Yuner.A	0.88	1.00	0.94	120
Lolyda.AA3	0.90	0.95	0.92	19
Dialplatform.B	0.96	1.00	0.98	27
Swizzor.gen!E	0.62	0.50	0.56	20
Lolyda.AT	0.89	1.00	0.94	24
VB.AT	0.98	0.87	0.92	62
Lolyda.AA2	1.00	0.96	0.98	28
Malex.gen!J	0.85	0.81	0.83	21
C2LOP.P	0.94	0.73	0.82	22
Autorun.K	0.00	0.00	0.00	16
Wintrim.BX	0.75	1.00	0.86	15
Instantaccess	0.94	1.00	0.97	65
Allaple.A	0.99	0.99	0.99	443
Adialer.C	1.00	1.00	1.00	19
Swizzor.gen!I	0.43	0.60	0.50	20
Rbot!gen	1.00	1.00	1.00	24
Dontovo.A	1.00	1.00	1.00	25
Alueron.gen!J	0.97	1.00	0.98	30
Obfuscator.AD	1.00	1.00	1.00	22
Lolyda.AA1	0.91	0.97	0.94	32
Fakerean	1.00	0.98	0.99	58
Skintrim.N	1.00	0.75	0.86	12
Allaple.L	1.00	0.98	0.99	239
Agent.FYI	1.00	1.00	1.00	18
C2LOP.gen!g	0.91	0.97	0.94	30
accuracy			0.95	1411

Obr. 13 Hodnoty Precision, Recall a F1 skóre pre triedy Malimg datasetu na modeli konvolučnej siete

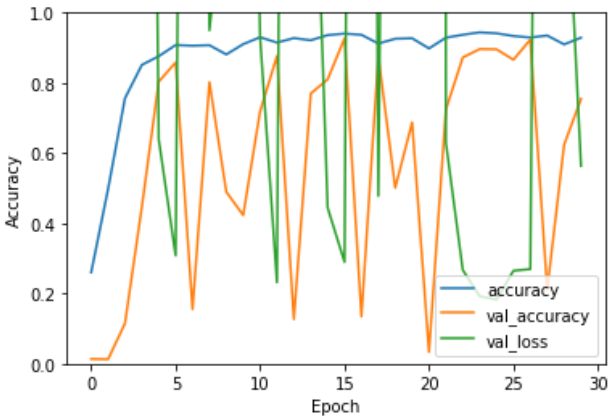
Na Obr. 14 sú dáta confusion matice natrénovaného modelu. Vidíme, že väčšina vzoriek testovacej časti datasetu bolo správne predikovaných. Avšak problematickou bola trieda Autorun.K, ktorej všetky vzorky boli ohodnotené ako Yuner.A.



Obr. 14 Confusion matica modelu konvolučnej siete na Malimg datasete

5.2 Malimg dataset a ResNet50

Na Malimg datasete sme sa rozhodli vyskúšať a natrénovať aj predpripravenú architektúru ResNet50. Trénovali sme na 30 epochách a veľkosťami jednotlivých dávok (angl. batchsize) 64. Obr. 15 ukazuje hodnoty presnosti, validačnej presnosti a validačnej loss funkcie na Malimg datasete a ResNet50 modeli. Najlepšia hodnota validačnej presnosti počas trénovania modelu bola 92,7 percent.



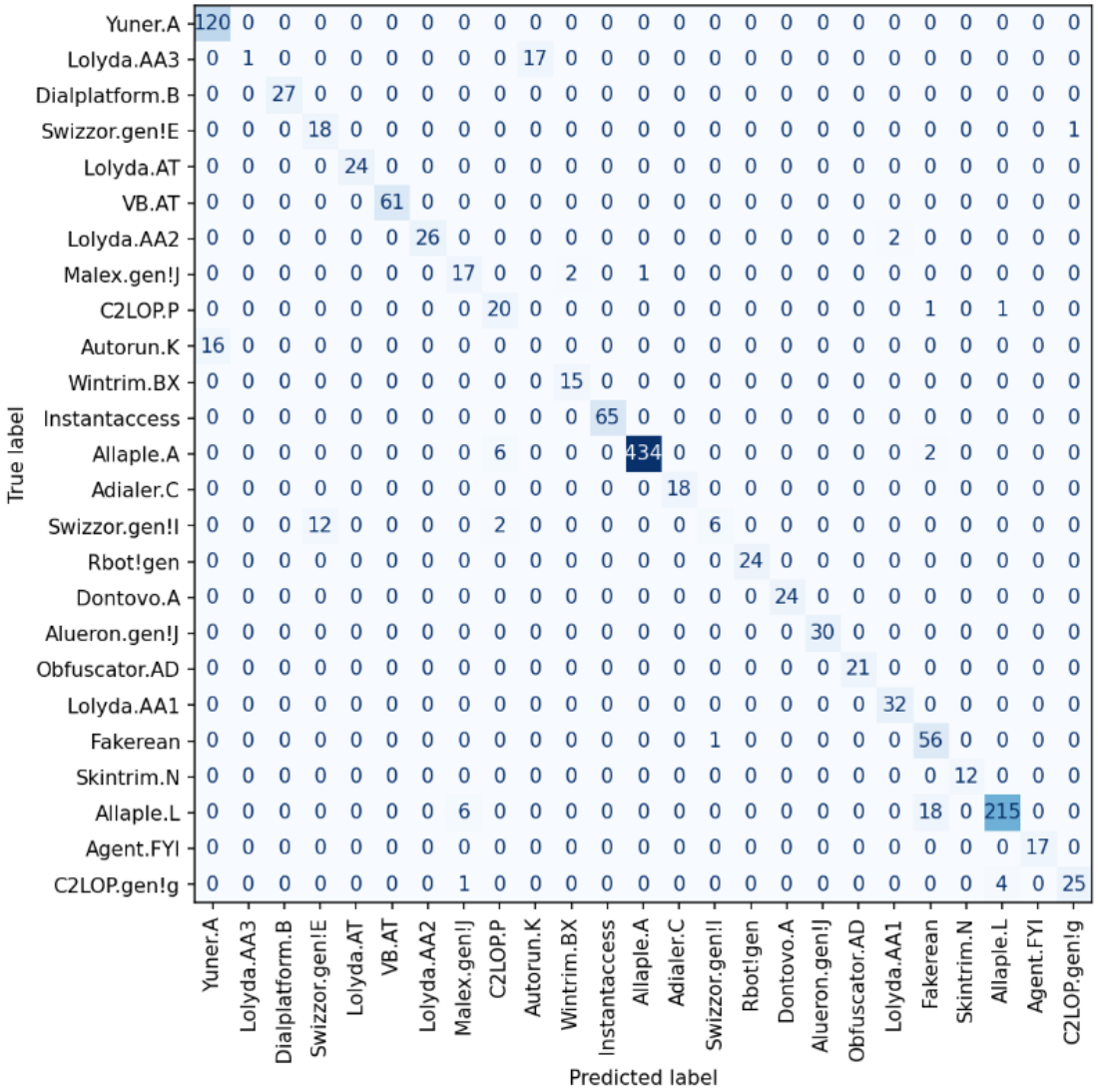
Obr. 15 Hodnoty presnosti, validačnej presnosti a validačnej loss funkcie na Malimg datasete a ResNet50 modeli

Na testovacej sade mal model hodnotu presnosti 93,36 percent. Obr. 16 obsahuje hodnoty precision, recall a f1 skóre pre jednotlivé triedy na natrénovanom modeli použitím ResNet50.

	precision	recall	f1-score	support
Yuner.A	0.88	1.00	0.94	120
Lolyda.AA3	1.00	0.06	0.11	18
Dialplatform.B	1.00	1.00	1.00	27
Swizzor.gen!E	0.60	0.95	0.73	19
Lolyda.AT	1.00	1.00	1.00	24
VB.AT	1.00	1.00	1.00	61
Lolyda.AA2	1.00	0.93	0.96	28
Malex.gen!J	0.71	0.85	0.77	20
C2LOP.P	0.71	0.91	0.80	22
Autorun.K	0.00	0.00	0.00	16
Wintrim.BX	0.88	1.00	0.94	15
Instantaccess	1.00	1.00	1.00	65
Allaple.A	1.00	0.98	0.99	442
Adialer.C	1.00	1.00	1.00	18
Swizzor.gen!I	0.86	0.30	0.44	20
Rbot!gen	1.00	1.00	1.00	24
Dontovo.A	1.00	1.00	1.00	24
Alueron.gen!J	1.00	1.00	1.00	30
Obfuscator.AD	1.00	1.00	1.00	21
Lolyda.AA1	0.94	1.00	0.97	32
Fakerean	0.73	0.98	0.84	57
Skintrim.N	1.00	1.00	1.00	12
Allaple.L	0.98	0.90	0.94	239
Agent.FYI	1.00	1.00	1.00	17
C2LOP.gen!g	0.96	0.83	0.89	30
accuracy			0.93	1401

Obr. 16 Hodnoty Precision, Recall a F1 skóre pre triedy Malimg datasetu na modeli Resnet50

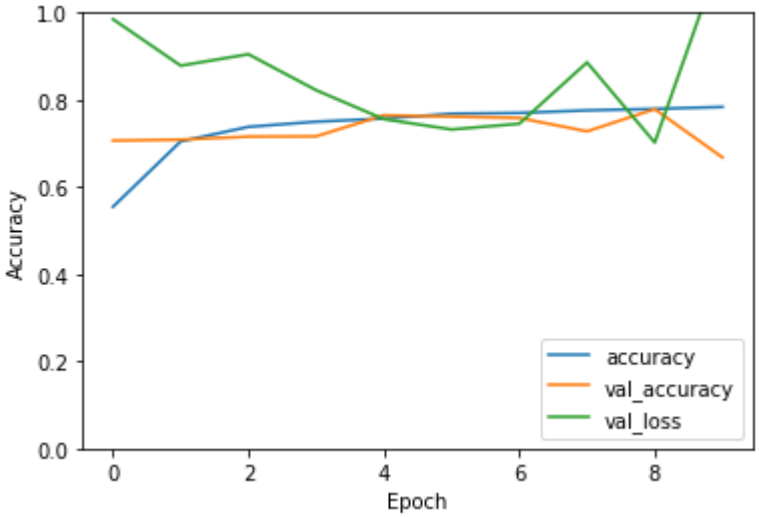
Na Obr. 17 sú dáta confusion matice natrénovaného modelu. Na tejto matici a takisto podľa Obr. 16 si môžeme všimnúť problematické triedy, keďže majú zlé hodnoty precision, recall alebo f1 skóre. Nimi sú Lolyda.AA3, Autorun.K a Swizzor.gen!.



Obr. 17 Confusion matica modelu ResNet50 siete na Malimg datasete

5.3 MalwareBaazar dataset a konvolučná sieť

Na MalwareBaazar datasete sme sa rozhodli trénovať model konvolučnej neurónovej siete na obidvoch vyššie uvedených ohodnoteniach jednotlivých vzoriek. Ohodnotenie od Hybrid-analysis nedosahovalo dobré hodnoty validačnej presnosti, maximálne len 68 percent, avšak s ohodnotením od MalwareBaazar to bolo lepšie. Hodnota validačnej presnosti bola 77,88 percent. V nasledujúcej časti sú ukázané výsledky natrénovaného modelu konvolučnej siete na MalwareBaazar datasete s ohodnotením od MalwareBaazar. Obr. 18 ukazuje hodnoty presnosti, validačnej presnosti a validačnej loss funkcie. Model sme trénovali na 10 epochách s batchsize 128.

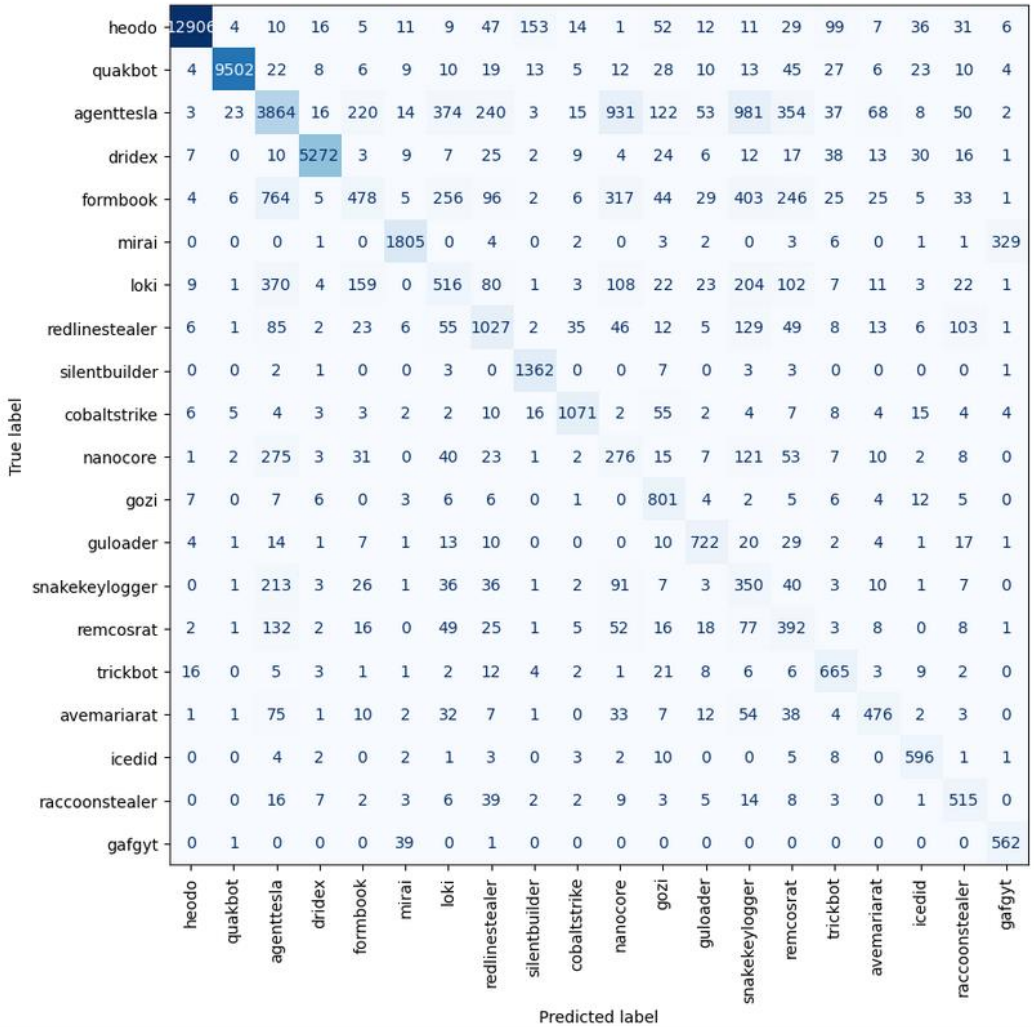


Obr. 18 Hodnoty presnosti, validačnej presnosti a validačnej loss funkcie na modeli konvolučnej siete natrénovanej na datasete aj ohodnotení od MalwareBaazar

Na testovacej sade mal model hodnotu presnosti 79,13 percent. Obr. 19 obsahuje hodnoty precision, recall a f1 skóre pre jednotlivé triedy na natrénovanom modeli konvolučnej siete. Obr. 20 je confusion maticou predikcií na testovacej sade MalwareBazaar datasete. Vidíme, že mimo diagonály matice sa nachádzajú veľké hodnoty a to viackrát, čo ukazuje, že sa nám nepodarilo natrénovať najspoľahlivejší model.

	precision	recall	f1-score	support
heodo	0.99	0.96	0.98	13459
quakbot	1.00	0.97	0.98	9776
agenttesla	0.66	0.52	0.58	7378
dridex	0.98	0.96	0.97	5505
formbook	0.48	0.17	0.26	2750
mirai	0.94	0.84	0.89	2157
loki	0.36	0.31	0.34	1646
redlinestealer	0.60	0.64	0.62	1614
silentbuilder	0.87	0.99	0.92	1382
cobaltstrike	0.91	0.87	0.89	1227
nanocore	0.15	0.31	0.20	877
gozi	0.64	0.92	0.75	875
guloader	0.78	0.84	0.81	857
snakekeylogger	0.15	0.42	0.22	831
remcosrat	0.27	0.49	0.35	808
trickbot	0.70	0.87	0.77	767
avemariarat	0.72	0.63	0.67	759
icedid	0.79	0.93	0.86	638
raccoonstealer	0.62	0.81	0.70	635
gafgyt	0.61	0.93	0.74	603
accuracy			0.79	54544

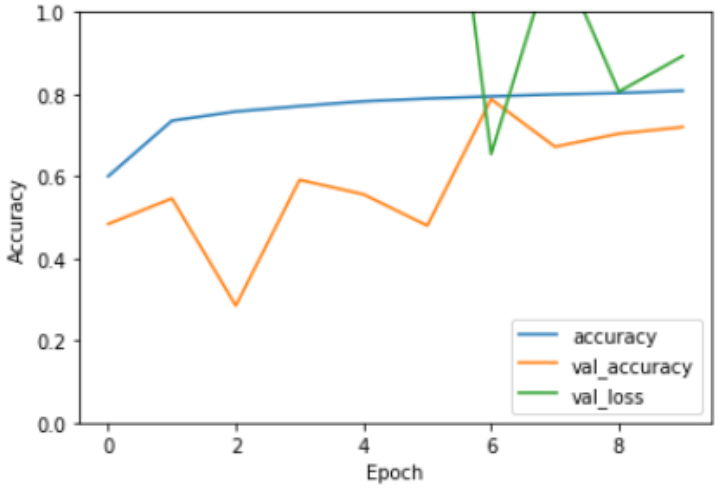
Obr. 19 Hodnoty Precision, Recall a F1 skóre pre triedy MalwareBazaar datasetu na modeli konvolučnej siete



Obr. 20 Confusion matica modelu konvolučnej siete na MalwareBazaar datasete

5.4 MalwareBaazar dataset a ResNet50

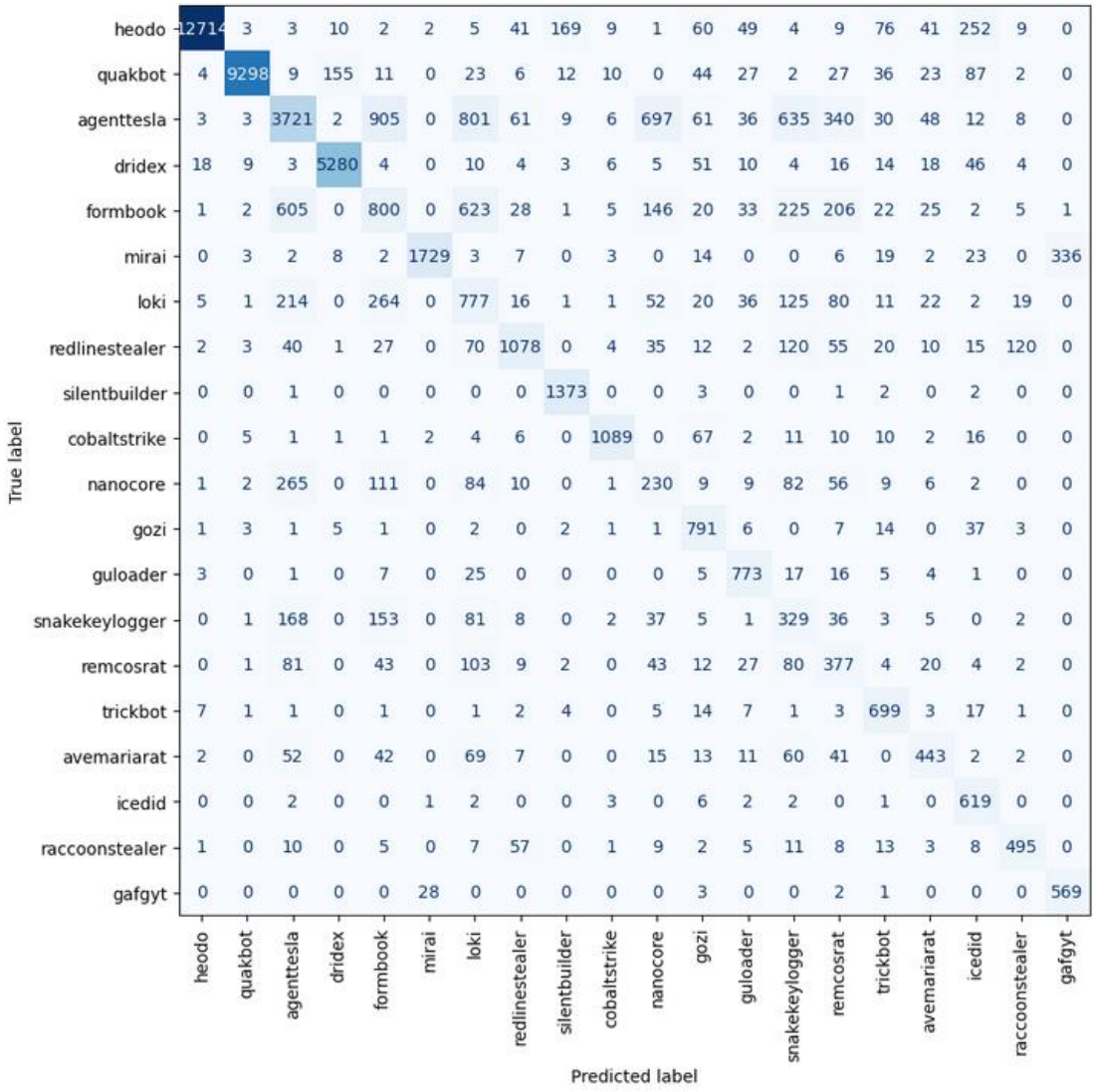
Na MalwareBaazar datasete s ohodnotením od MalwareBaazar sme sa rozhodli vyskúšať aj ResNet50 model. Najväčšia hodnota validačnej presnosti počas trénovania modelu bola 78,79 percent. Model sme trénovali na 10 epochách s batchsize 64. Výsledky sú prezentované v zmysle výsledkov predchádzajúcich modelov, tentokrát na obrázkoch Obr. 21, Obr. 22 a Obr. 23. Vidíme,



Obr. 21 Hodnoty presnosti, validačnej presnosti a validačnej loss funkcie na ResNet50 modeli natrénovanej na datasete aj ohodnotení od MalwareBaazar

	precision	recall	f1-score	support
heodo	1.00	0.94	0.97	13459
quakbot	1.00	0.95	0.97	9776
agenttesla	0.72	0.50	0.59	7378
dridex	0.97	0.96	0.96	5505
formbook	0.34	0.29	0.31	2750
mirai	0.98	0.80	0.88	2157
loki	0.29	0.47	0.36	1646
redlinestealer	0.80	0.67	0.73	1614
silentbuilder	0.87	0.99	0.93	1382
cobaltstrike	0.95	0.89	0.92	1227
nanocore	0.18	0.26	0.21	877
gozi	0.65	0.90	0.76	875
guloader	0.75	0.90	0.82	857
snakekeylogger	0.19	0.40	0.26	831
remcosrat	0.29	0.47	0.36	808
trickbot	0.71	0.91	0.80	767
avemariarat	0.66	0.58	0.62	759
icedid	0.54	0.97	0.69	638
raccoonstealer	0.74	0.78	0.76	635
gafgyt	0.63	0.94	0.75	603
accuracy			0.79	54544

Obr. 22 Hodnoty Precision, Recall a F1 skóre pre triedy MalwareBazaar datasetu na ResNet50 modeli



Obr. 23 Confusion matica modelu ResNet50 na MalwareBazaar datasete

6 Prehľad podobných prác

Pre našu prácu je podstatná časť porovnanie rôznych riešení, ktoré už boli vytvorené. V tejto časti sú rozpísané rôzne prístupy k riešeniu tohto problému a ich úspešnosť.

Autori v článku [18] na riešenie problému použili pre reprezentáciu súboru do vizuálnej podoby prevod na greyscale. Ich neurónová sieť je založená na princípe konvolučnej neurónovej siete. Neurónovú sieť trénovali na datasetoch Malimg [23] a na datasete Microsoft Kaggle competition dataset od Microsoftu [22], ktorý v roku 2015 otvoril súťaž na riešenie tohto problému. Úspešnosť tohto prístupu je ukázaná na presnosti klasifikácie. Pre Malimg je úspešnosť 98.52% a pre dataset od Microsoftu je rovná v jednom prípade 98.99% a v druhom 99.97%.

Autori v práci [30] použili pre reprezentáciu malvéru postup, kedy sa daný malvér spúšťa v bezpečnom prostredí. Následne z logov získali pole čísel, ktoré dávajú na vstup pre neurónovú sieť. V tomto postupe sa reprezentácia nedáva do obrázkovej formy, ale pole sa hneď dá na vstup pre neurónovú sieť. Neurónová sieť je vytvorená na základe konvolučnej neurónovej siete, a rekurentných vrstiev. Ako dataset používa kombináciu datasetu od VirusShare [31], Maltrieve [32] a osobnú zbierku. Pre evaluáciu je použitá 3-násobna krížová validácia. Table 1. a Table 2. ukazujú hodnoty úspešnosti, presnosti a recall.

Tab. 4 hodnoty úspešnosti, presnosti a recall začiatok [30]

Family	Multiplug	Kazy	Morstar	Zusy	Soft Pulse	Somoto
accuracy	98.9	100	100	100	100	100
precision	99.8	99.9	99.9	57.5	99.1	100
recall	99	100	100	100	100	100

Tab. 5 hodnoty úspešnosti, presnosti a pokračovanie [30]

Family	Mikey	Amonetize	Eldorado	Kryptik	AVERAGE
accuracy	0	99.1	99.4	96.6	89.4
precision	0	100	100	100	85.6
recall	0	99.6	99.5	96.2	89.4

Autori v rámci článku [21] najprv spracovali súbory a následne získali čiernobiely obrázok na základe SimHashu [33], kde každý bit z hashu je prevedený na hodnotu od 0 do 255 nasledujúcim spôsobom: 0 => 0, 1 => 255. Pre trénovanie a testovanie svojho modelu používa Microsoft Kaggle competition dataset [22]. Neurónová sieť je založená na logike konvolučnej neurónovej siete. Klasifikačná presnosť tohto riešenia dosiahla maximálne hodnotu 98.26%, a v priemere 98.862%.

Autori v rámci práce [34] pri reprezentácií použili pre nás úplne nový prístup. Je založený na vytváraní troj-kanálových RGB obrázkov z binárnej reprezentácie vzorky, z inštrukcií jazyka assembler a stringov súboru. Na trénovanie a testovanie takisto použili Microsoft Kaggle competition dataset [22].

Záver

V tejto práci sme sa venovali problematike klasifikácie malvéru pomocou neurónových sietí. Na začiatku práce je predstavená problematika veľkého rastu počtu detegovaného malvéru. Po úvode do problematiky je predstavený pojem malvér. Opisujeme jeho účel existencie. V ďalšom kroku je porovnaná statická a dynamická analýza malvéru. Súčasťou statickej analýzy je predstavenie formátu Windows spustiteľných súborov. Takisto sú opísané rôzne metódy statickej analýzy.

Klasifikácia malvéru je dosiahnuteľná viacerými spôsobmi. V práci popisujeme rôzne metódy, ako malvér klasifikovať. Tieto metódy sú často časovo náročné. Preto má zmysel riešiť problém klasifikácie pomocou overeného spôsobu strojového učenia, konkrétne o klasifikáciu obrázkov.

V práci boli predstavené rôzne spôsoby reprezentácie malvéra do obrázkovej formy. Na trénovanie sme modelov sme porovnali viaceré datasety. Následne sme overili klasifikáciu malvéru na datasete Malimg. Výsledky potvrdzujú vhodnosť použitého prístupu. Preto sme chceli tento prístup overiť na inom väčšom datasete. Ten sme získali od platformy MalwareBazaar. Ukázalo sa, že nie je také jednoduché natrénovať neurónovú sieť na hocijakých vzorkách malvéru. Výsledky natrénovaných modelov neboli ideálne. Predpokladom je, že tento fakt nastal kvôli rôznorodosti formátov súborov. Do budúcna je potrebné sa bližšie pozrieť na distribúciu rôznych typov súborov v datasete, a analyzovať možné spôsoby riešenia.

Výsledky tejto práce umožňujú pripraviť službu, ktorá bude poskytovať klasifikáciu malvéru použitím rôznych modelov neurónových sietí.

Zoznam použitej literatúry

[1] "Malware Statistics & Trends Report | AV-TEST." https://www.av-test.org/en/statistics/malware/ (accessed May 16, 2022).

[2] I. A.Saeed, A. Selamat, and A. M. A. Abuagoub, "A Survey on Malware and Malware Detection Systems," *IJCA*, vol. 67, no. 16, pp. 25–31, Apr. 2013, doi: 10.5120/11480-7108.

[3] G. McGraw and G. Morrisett, "Attacking Malicious Code: A Report to the Infosec Research Council," *IEEE Softw.*, vol. 17, no. 5, pp. 33–41, Sep. 2000, doi: 10.1109/52.877857.

[4] M. Sikorski and A. Honig, *Practical malware analysis: the hands-on guide to dissecting malicious software*. no starch press, 2012.

[5] J. Saxe and H. Sanders, *Malware data science: attack detection and attribution*. No Starch Press, 2018.

[6] Karl-Bridge-Microsoft, "PE Format - Win32 apps." https://docs.microsoft.com/en-us/windows/win32/debug/pe-format (accessed May 16, 2022).

[7] "Portable Executable File Format." https://blog.kowalczyk.info/articles/pefileformat.html (accessed May 16, 2022).

[8] "VirusTotal - Home." https://www.virustotal.com/gui/home/upload (accessed Feb. 10, 2022).

[9] "Free Automated Malware Analysis Service - powered by Falcon Sandbox." https://www.hybrid-analysis.com/ (accessed May 17, 2022).

[10] E. Carrera, *pefile*. 2022. Accessed: May 16, 2022. [Online]. Available: https://github.com/erocarrera/pefile

[11] "ANY.RUN - Interactive Online Malware Sandbox." https://any.run/ (accessed May 17, 2022).

[12] "Cuckoo Sandbox - Automated Malware Analysis." https://cuckoosandbox.org/ (accessed May 17, 2022).

[13] K. Monnappa, *Learning Malware Analysis: Explore the concepts, tools, and techniques to analyze and investigate Windows malware*. Packt Publishing Ltd, 2018.

[14] *ssdeep-project/ssdeep*. ssdeep-project, 2022. Accessed: May 17, 2022. [Online]. Available: https://github.com/ssdeep-project/ssdeep

[15] *DinoTools/python-ssdeep*. DinoTools, 2022. Accessed: May 17, 2022. [Online]. Available: https://github.com/DinoTools/python-ssdeep

[16] "YARA - The pattern matching swiss knife for malware researchers." http://virustotal.github.io/yara/ (accessed May 18, 2022).

[17] "Malware Analyst's Cookbook and DVD," *Guide books*. https://dl.acm.org/doi/abs/10.5555/1964877 (accessed May 18, 2022).

[18] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, and F. Iqbal, "Malware Classification with Deep Convolutional Neural Networks," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, Feb. 2018, pp. 1–5. doi: 10.1109/NTMS.2018.8328749.

[19] A. Singh, A. Handa, N. Kumar, and S. K. Shukla, "Malware Classification Using Image Representation," in *Cyber Security Cryptography and Machine Learning*, Cham, 2019, pp. 75–92. doi: 10.1007/978-3-030-20951-3_6.

[20] "objdump(1) - Linux manual page." https://man7.org/linux/man-pages/man1/objdump.1.html (accessed May 18, 2022).

[21] S. Ni, Q. Qian, and R. Zhang, "Malware identification using visualization images and deep learning," *Computers & Security*, vol. 77, pp. 871–885, Aug. 2018, doi: 10.1016/j.cose.2018.04.005.

[22] "Microsoft Malware Classification Challenge (BIG 2015)." https://kaggle.com/c/malware-classification (accessed Feb. 10, 2022).

[23] "malimg_dataset.zip," *Dropbox*. https://www.dropbox.com/s/ep8qjakfwh1rzk4/malimg_dataset.zip (accessed May 18, 2022).

[24] "MalwareBazaar | Malware sample exchange." https://bazaar.abuse.ch/ (accessed May 18, 2022).

[25] "CS231n Convolutional Neural Networks for Visual Recognition." https://cs231n.github.io/convolutional-networks/ (accessed May 18, 2022).

[26] Ashray Bhandare, "Convolutional Neural Networks," 18:03:50 UTC. Accessed: May 18, 2022. [Online]. Available:

https://www.slideshare.net/ashraybhandare/convolutional-neural-networks-88119848

[27] "Detailed Guide to Understand and Implement ResNets," *CV-Tricks.com*, Sep. 17, 2019. https://cv-tricks.com/keras/understand-implement-resnets/ (accessed May 18, 2022).

[28] B. T, "Comprehensive Guide on Multiclass Classification Metrics," *Medium*, Oct. 19, 2021. https://towardsdatascience.com/comprehensive-guide-on-multiclass-classification-metrics-af94cfb83fbd (accessed May 18, 2022).

[29] "scikit-learn: machine learning in Python — scikit-learn 1.1.0 documentation." https://scikit-learn.org/stable/ (accessed May 18, 2022).

[30] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep Learning for Classification of Malware System Call Sequences," in *AI 2016: Advances in Artificial Intelligence*, vol. 9992, B. H. Kang and Q. Bai, Eds. Cham: Springer International Publishing, 2016, pp. 137–149. doi: 10.1007/978-3-319-50127-7_11.

[31] "VirusShare.com." https://virusshare.com/ (accessed Feb. 10, 2022).

[32] K. Maxwell, *krmaxwell/maltrieve*. 2022. Accessed: Feb. 10, 2022. [Online]. Available: https://github.com/krmaxwell/maltrieve

[33] M. S. Charikar, "Similarity estimation techniques from rounding algorithms," in *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, New York, NY, USA, Máj 2002, pp. 380–388. doi: 10.1145/509907.509965.

[34] Y. Jian, H. Kuang, C. Ren, Z. Ma, and H. Wang, "A novel framework for image-based malware detection with a deep neural network," *Computers & Security*, vol. 109, p. 102400, Oct. 2021, doi: 10.1016/j.cose.2021.102400.

Prílohy

Príloha A: CD médium – bakalárska práca v elektronickej podobe