TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Pokročilé metódy grafickej analýzy komplexných dát

DIPLOMOVÁ PRÁCA

2012 **Martin CHALUPKA**

TECHNICKÁ UNIVERZITA V KOŠICIACH

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

POKROČILÉ METÓDY GRAFICKEJ ANALÝZY KOMPLEXNÝCH DÁT

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	9.2.1 Informatika
Školiace pracovisko:	Katedra počítačov a informatiky (KPI)
Školiteľ:	doc. Ing. František Jakab, PhD.
Konzultant:	Ing. Ivan Klimek

Košice 2012 **Martin CHALUPKA**

Abstrakt v SJ

Táto práca sa zaoberá možnosťami využitia grafickej analýzy dát v prostredí informačných technológií. Obsahuje popis vybraných grafických štatistických metód a uvádza príklady použitia niektorých z nich na vizualizáciu rôznych typov dát. Zaoberá sa hlavne aplikáciou týchto metód v oblasti sieťových technológií a bezpečnosti sietí, napr. pre monitorovanie sieťovej prevádzky alebo rozoznávanie typu prenášaných dát. Práca obsahuje aj konkrétnu implementáciu týchto metód na príklade algoritmu na detekciu škodlivého softvéru alebo programu pre odporúčanie internetových stránok používateľom.

Kľúčové slová

Detekcia škodlivého softvéru, Fourierova transformácia, Vlnková transformácia, Entropia, Logistická regresia, Web odporúčania

Abstrakt v AJ

This diploma thesis is studying a possible usage of graphical analysis in computer science. It describes different graphical and statistical methods and gives examples of visualization of many data types. The work is especially focused on their application in computer networks and network security, e.g. monitoring of network traffic or identification of the type of data being transferred. The thesis contains also two example implementations of such methods, one aimed at detection of malicious software and the second is a system which proposes pages to the users based on their habits.

Kľúčové slová v AJ

Malware detection, Fourier transform, Wavelet transform, Entropy, Logistic regression, Web recommendation

Zadanie práce

TECHNICKÁ UNIVERZITA V KOŠICIACH

Fakulta elektrotechniky a informatiky

Katedra počítačov a informatiky

ZADANIE
DIPLOMOVEJ PRÁCE

Študijný odbor: **9.2.1 Informatika**

Študijný program: **Informatika**

Názov práce:

Pokročilé metódy grafickej analýzy komplexných dát
Emerging methodes of graphical analysis of complex data

Študent (tituly, meno, priezvisko): **Bc. Martin Chalupka**

Školiteľ (tituly, meno, priezvisko): **doc. Ing. František Jakab, PhD.**

Školiace pracovisko: **Katedra počítačov a informatiky**

Konzultant práce (tituly, meno, priezvisko): **Ing. Ivan Klimek**

Pracovisko konzultanta: **Katedra počítačov a informatiky**

Pokyny na vypracovanie diplomovej práce:

1. **Prieskum možnosti využitia grafických metód analýzy komplexných dát**
2. **Analýza a aplikácia spôsobov využitia prezentovaných metód na odhalenie škodlivého kódu**
3. **Návrh a aplikácia využitia prezentovaných metód pri analýze Web komunikácie**
4. **Zhodnotenie riešenia**
5. **Spracovanie dokumentácie podľa štandardov katedry**

Jazyk, v ktorom sa práca vypracuje: slovenský

Termín pre odovzdanie práce: 27.04.2012

Dátum zadania diplomovej práce: 31.10.2011

prof. Ing. Ján Kollár, CSc.
vedúci garantujúceho pracoviska

-12-

prof. Ing. Liberios Vokorokos, PhD.
dekan fakulty

Čestné vyhlásenie

Vyhlasujem, že som celú diplomovú prácu vypracoval/a samostatne s použitím uvedenej odbornej literatúry.

Košice, 27. apríl 2012 .....

vlastnoručný podpis

Poďakovanie

V prvom rade, by som chcel poďakovať vedúcemu práce Doc. Ing. Františkovi Jakabovi, PhD. za odborné vedenie a pripomienky pri tvorbe diplomovej práce.

Ďalej chcem poďakovať Bc. Jánovi Hrončákovi, ktorý spolupracoval pri analýze a implementácií štatistických metód logistickej regresie, pričom obsahom svojej diplomovej práce prispel ďalšími alternatívami pokročilých metód analýzy dát.

Taktiež chcem osobitne poďakovať konzultantovi práce Ing. Ivanovi Klimekovi za trpezlivosť, odborné vedenie, pomoc, návrhy postupov a použitých metód, myšlienky a prototypy analýzy, riešenia množstva problémov pri analýze a vývoji a poskytnutie ďalších cenných praktických znalostí z danej problematiky.

Na záver chcem poďakovať mojej rodine za neustálu pomoc a podporu v akademickom aj osobnom živote.

Ďakujem

Predhovor

V informatike ako jednom z najrýchlejšie sa rozvíjajúcich vedných odborov stále vznikajú nové možnosti a postupy, ktoré dovoľujú softvérovým inžinierom efektívnejšie a rýchlejšie vytvárať požadované programy, a aplikácie s čoraz širšou funkcionalitou a robustnosťou. Táto skutočnosť však so sebou prináša aj nežiaduce následky ako sú napríklad nové bezpečnostné hrozby v podobe stále nových a čoraz komplikovanejších škodlivých programov, ktoré využívajú metódy na zamedzenie odhalenia, pôvodne navrhnuté na iné účely, alebo aplikácie používajúce automatické generovanie kódu kompilátormi, ktorých výsledný zdrojový kód je vždy odlišný. Medzi nežiaduce následky neustáleho rozširovania internetu a objemu jeho obsahu patrí aj rapídne zväčšenie celosvetovej informačnej domény, s čím je spojená aj jej hraničná neprehľadnosť. Z celosvetového hľadiska každodenne narastá počet nových internetových stránok s rôznym obsahom, a zároveň každodenne vznikajú nové typy škodlivých programov, aplikácií a iných hrozieb.

Pri súčasnom tempe vývoja informačných technológií je preto čoraz ťažšie používať konvenčné metódy a je nutné hľadať nové spôsoby, ktoré by dlhodobo riešili nastolené problémy v tejto oblasti. Jednou z nekonvenčných metód je využitie grafických štatistických metód analýzy dát, ktoré sa bežne využívajú v iných vedných odboroch, pričom je dokázané, že ich aplikácia môže priniesť požadované výsledky aj v odbore informačných technológií.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Zoznam symbolov a skratiek

DFT **D**iscrete **F**ourier **t**ransform

FFT **F**ast **F**ourier **t**ransform

HTTP **H**yper**t**ext **T**ransfer **P**rotocol

RGB **R**ed **G**reen **B**lue

STFT **S**hort-**t**ime **F**ourier **t**ransform

TCP **T**ransmission **C**ontrol **P**rotocol

URL **U**niform **R**esource **L**ocator

Slovník termínov

Adware (angl. Advertising supported software) je softvér, ktorý po svojom spustení automaticky zobrazuje, prehráva alebo sťahuje reklamný obsah na počítač obete.

Clickstream je záznam o činnosti používateľa (napr. prehliadanie web stránok).

Cookie (taktiež HTTP cookie, Web cookie) je malé množstvo dát uložených webovou stránkou v prehliadači, pričom tieto dáta sú následne poslané späť pri opakovanej návšteve rovnakej stránky.

Fourierova transformácia (angl. Fourier transform) je algoritmus slúžiaci na rozklad postupnosti hodnôt do zložiek rôznych frekvencií.

Histogram alebo **Histogram početností** je stĺpcový diagram tvorený obdĺžnikmi, ktorých základne majú dĺžku zvolených intervalov a ktorých výšky majú veľkosť príslušných absolútnych početností zvolených tried.

Kompresia je zhustenie obsahu alebo redukcia počtu bitov ktoré sú potrebné na reprezentáciu určitého dátového obsahu.

Malware je všeobecné označenie pre škodlivý softvér. Patria sem napríklad vírusy, trójske kone, spyware a adware.

Scalogram je v spracovaní signálu vizuálny spôsob zobrazenia vlnkovej transformácie.

Shannonova entropia (angl. Shannon entropy) je veličina udávajúca mieru neusporiadanosti hodnôt vstupnej diskrétnej premennej.

Spektrogram je časovo premenná spektrálna reprezentácia, ktorá ukazuje, ako sa mení spektrálna hustota signálu za čas.

Spyware (angl. Spy ware) je softvér ktorý sa bez vedomia používateľa pokúša nájsť a odcudziť citlivé dáta z počítača (napríklad heslá).

Vlnka (angl. Wavelet) je vlnová oscilácia s amplitúdou, ktorá začína od nuly, jej hodnota rastie a klesá späť do nulovej hodnoty.

Web (angl. World Wide Web) je systém vzájomne previazaných hypertextových dokumentov, prístupných na internete.

Zero-Day útok (angl. Zero-Day attack) je typ počítačového útoku, ktorý ešte nebol zaznamenaný softvérovými vývojármi.

Úvod

Vo vedných odboroch ako matematika, fyzika, elektrotechnika a medicína sa vo veľkej miere využívajú metódy analýzy dát, ktorých využitie v informatike bolo donedávna považované za príliš nepresné, málo exaktné. V dnešnej dobe, kedy sa veľkosti diskov pohybujú v terabajtoch, prenosové rýchlosti v gigabajtoch a softvér je generovaný poloautomaticky, je stále potrebné hľadať nové, rýchlejšie, presnejšie a výkonnejšie metódy analýzy tohto nepretržite narastajúceho množstva dát. Práve tu začína mať využitie týchto metód svoje opodstatnenie, pričom rozsah ich využitia sa stále zväčšuje a neustále sú objavované ďalšie možnosti ich aplikácie.

Hoci metódy používané v tejto práci je možné zaradiť do rôznych skupín ako napríklad matematické, grafické, štatistické a iné, všetky pokročilé metódy analýzy dát sú založené v konečnom dôsledku na ich kombinácii. Vo väčšine prípadov aj keď samotný základ metódy nepatrí medzi grafické metódy analýzy dát, v prvom bode analýzy má najvyššiu výpovednú hodnotu grafické znázornenie výsledkov danej metódy.

Po formulácii úlohy práce v prvej kapitole druhá kapitola práce ozrejmuje základy terminológie a používa matematické definície na popísanie metód, ktoré boli vybrané ako najvhodnejšie pre potreby analýzy komplexných dát v prostredí informačných technológií.

Tretia kapitola sa zameriava na proces výberu a aplikáciu najvhodnejších metód analýzy na binárne dáta. Pomocou vybraných pokročilých grafických metód je potrebné nájsť a zostaviť nielen postup analýzy pre určenie samotného typu súboru, ale pokúsiť sa navrhnúť a implementovať systém schopný podľa reprezentačnej vzorky identifikovať určenú podmnožinu zadaného typu súborov. Návrh a implementácia takéhoto systému využívajúceho pokročilé grafické metódy analýzy dát je následne možné použiť pri detekcií jednej z najrozšírenejších počítačových hrozieb a to škodlivého softvéru.

V štvrtej kapitole sú opísané možnosti aplikácie vybraných metód analýzy na rôzne druhy web komunikácie. Možnosti použitia daných metód závisia v tomto prípade hlavne od šírky analyzovaného časového intervalu, pričom so zmenou šírky analyzovaného intervalu je spojená aj zmena vlastností analyzovaných dát. Výsledkom aplikácie vybraných metód analýzy web komunikácie je reálny systém web odporúčaní,

ktorý pomocou analýzy web komunikácie rôznych používateľov dokáže vytvoriť čo najvhodnejšie odporúčanie web stránky pre každého používateľa.

Záver práce zhŕňa výsledky a poznatky predchádzajúcich kapitol a zároveň polemizuje o ďalších možnostiach rozvoja a smerovania práce.

1 Formulácia úlohy

S ohľadom na zadanie diplomovej práce je potrebné oboznámiť sa so štatistickými a grafickými metódami analýzy komplexných dát v informačných technológiách. Ďalej je potrebné analyzovať možnosti aplikácie prezentovaných metód na odhalenie škodlivého softvéru a ich využitie pri analýze web komunikácie. Nakoniec je potrebné zhrnúť závery a výsledky riešenia a podľa štandardov katedry vypracovať príslušnú dokumentáciu.

2 Metódy grafickej analýzy dát

Metódy analýzy dát popísané ďalej v tejto práci sú založené na využití matematických metód a postupov, ktoré dokážu efektívne spracovávať a analyzovať dáta prenášané a využívané informačnými systémami. Základným princípom spoločným pre všetky tieto postupy je získanie informácií o štruktúre, type alebo charaktere dát bez ich hĺbkovej analýzy klasickými informačnými postupmi.

Väčšina opísaných postupov sa zameriava na analýzu dát prenášaných medzi dvoma uzlami, ale taktiež analýzu statických dát, ktoré je možné reprezentovať sekvenciou bitov na časovej osi. Z toho vyplýva, že sa jedná o analýzu , ktorú je možné popísať predpisom

$$X = \int_{-\infty}^{\infty} f(x)dt \quad .$$

Pokročilé metódy grafickej analýzy dát popísané ďalej je možné rozdeliť podľa hladiny významnosti analyzovaných dát v závislosti od šírky analyzovaného časového intervalu (najnižšia, stredná a najvyššia hladina významnosti). S narastajúcou dĺžkou vyhodnocovaného časového intervalu sa menia vlastnosti vstupných dát, ktoré je možno analyzovať. Vlastnosti a spôsoby analýzy týchto troch typov dát sú bližšie popísané v kapitole č. 4.

Pri analýze dát sú okrem ďalej popísaných pokročilejších metód často používané základné algebrické operácie ako napríklad určenie maximálnej hodnoty poľa, výpočet aritmetického priemeru poľa hodnôt alebo výpočet početnosti hodnôt poľa.

2.1 Entropia

Entropia je vo všeobecnosti veličina, ktorá meria neusporiadanosť (náhodnosť, neporiadok) systému. Pôvodne bol pojem entropia zavedený v termodynamike na vyjadrenie neusporiadanosti danej fyzikálnej sústavy. Claude E. Shannon ako prvý zaviedol pojem entropie aj do informatiky ako vedného odboru, pričom pojem entropie je podľa tohto matematika a elektrotechnického inžiniera nazývaný aj Shannonova entropia. [4]

2.2 Shannonova entropia

V teórií informatiky vyjadruje entropia mieru nepravidelnosti spájanú tiež s náhodnosťou daných premenných. V tomto kontexte shannonová entropia udáva mieru neusporiadanosti hodnôt vstupnej diskrétnej premennej, pričom táto hodnota tiež udáva pravdepodobnosť výskytu očakávanej hodnoty.

Shannonova entropia je definovaná matematicky takto:

Nech X je diskrétna náhodná premenná s konečným počtom možných hodnôt X_1,X_2, ..., X_N s pravdepodobnosťou výskytu p_1, p_2,... p_N, respektíve kde $p_i \geq 0$ $i = 1,2,...,n$ $\sum_{i=1}^{n} p_i = 1$. Nech h je funkcia definovaná na intervale [0,1] a h(p) je interpretovaná ako nepravidelnosť asociovaná s udalosťou (zmenou) $X = x_i$, i=1,2,...,n alebo informácia o tom že premenná X nadobudla hodnotu x_i. Pre každé n definujeme funkciu H_n s premennými $p_1,p_2,...,p_n$. Potom funkcia $H_n(p_1,p_2,...p_n)$ je interpretovaná ako priemerná nepravidelnosť spojená s nastaním udalosti $\{X= x_i\}$. [5]

2.3 Diskréna Fourierova transformácia

Je algoritmus slúžiaci na rozklad postupnosti hodnôt do zložiek rôznych frekvencií. Je to transformácia, ktorá zobrazí vstupnú funkciu (zväčša definovanú v časovej oblasti) na výstupnú funkciu, ktorá je frekvenčnou reprezentáciou vstupných dát a vyžaduje nenulové vstupné pole hodnôt, ktoré musí mať konečnú dĺžku na časovej osi [1]. Diskrétna Fourierova transformácia je matematicky definovaná takto:

Sekvencia N komplexných čísel x_0,x_1, ... x_{N-1} je transformovaná na sekvenciu N komplexných čísel X_0,X_1, ... X_{N-1} pomocou diskrétnej Fourierovej transformácie podľa predpisu

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0,....,N-1 .$$

Algoritmus diskrétnej Fourierovej transformácie je pre využitie a aplikáciu v oblasti informačných technológií pracujúcich s veľkým počtom vstupných údajov v reálnom čase príliš pomalá, čo vyplýva už priamo z jej definície. Preto sa častejšie využíva rýchla Fourierova transformácia, ktorá dokáže dosiahnuť rovnaký výsledok za oveľa kratší výpočtový čas. Na ilustráciu na výpočet transformácie N vstupných čísel je potrebné vykonať pri použití diskrétnej Fourierovej transformácie (N^2) operácií, pričom

pri použití rýchlej Fourierovej transformácie je na rovnaký počet vstupných dát potrebné len (N.logN) operácií [1].

2.4 Rýchla Fourierova transformácia

Je efektívny algoritmus na výpočet diskrétnej Fourierovej transformácie (DFT) a jej inverznej funkcie. Rýchla Fourierova transformácia je matematicky definovaná takto:

Sekvencia N komplexných čísel $x_0, x_1, \ldots x_{N-1}$ je transformovaná na sekvenciu N komplexných čísel $X_0, X_1, \ldots X_{N-1}$ pomocou diskrétnej fourierovej transformácie podľa predpisu

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i 2\pi k \frac{n}{N}} \qquad k = 0,\ldots,N-1 \ .$$

Pre tento výpočet existuje veľké množstvo rôznych algoritmov, ktoré zahŕňajú široký rozsah matematiky, od jednoduchých algoritmov využívajúcich komplexné čísla až po teóriu grúp a číslicových teórií. Najčastejšie a najviac využívaný je Cooley-Tourkey algoritmus. Tento algoritmus je založený na princípe „rozdeľ a panuj“ (divide and conquer), ktorý rekurzívne rozloží diskrétnu Fourierovu transformáciu akejkoľvek veľkosti N = N1.N2 do menších DFT. Najznámejšie použitie Cooley-Turkey algoritmu je pomocou delenia pôvodnej transformácie na 2 časti pri každom kroku transformácie. Z toho vyplýva aj obmedzenie algoritmu na veľkosť vstupných dát, ktorá by mala byť mocninou čísla 2. Pri splnení tejto podmienky má rýchla Fourierová transformácia najvyššiu účinnosť [3].

2.5 Krátkodobá Fourierova transformácia

Je algoritmus patriaci do skupiny Fourierových transformácií, primárne slúžiaci na stanovenie frekvencie a fázy všetkých častí vstupného signálu v celej šírke daného časového úseku. Vo všeobecnosti je Fourierova transformácia jednorozmerná funkcia. Pri krátkodobej Fourierovej transformácií je však signál vytvorený z postupnosti malých častí vstupu (okien). Spojením tejto postupnosti znovu vzniká celá časová os signálu na ktorej je vynesený výsledok transformácie v podobe dvojdimenzionálnej reprezentácie signálu [2].

Krátkodobá Fourierova transformácia (STFT) je definovaná vo všeobecnosti na spojitej časovej osi vstupného signálu podľa predpisu

$$STFT\{x(t)\} \equiv X(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-j\omega t}dt$$

.

Pre využitie v informačných technológiách, kde vystupujú všetky veličiny reprezentované v diskrétnej forme, je krátkodobá Fourierova transformácia definovaná predpisom

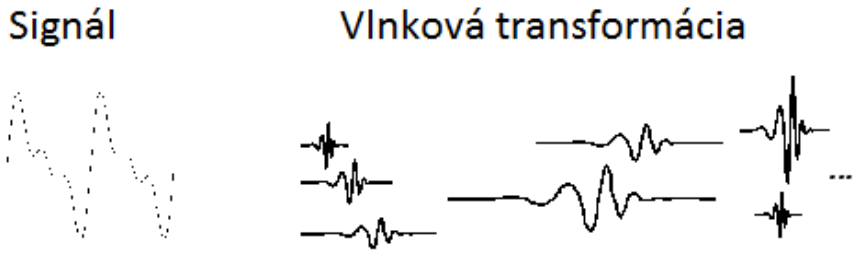
$$STFT\{x[n]\} \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w(n-m)e^{-j\omega n}$$

,

pričom okno (w – window) je pri spracovaní signálu funkcia, ktorá má takmer vždy nulovú hodnotu a v zadanej časti (imaginárnom obdĺžniku) je obor hodnôt funkcie zhodný s definičným oborom, prípadne upravený pomocou určitého predpisu.

2.6 Transformácia pomocou vlniek

Rovnako ako Fourierova transformácia aj spojitá vlnková transformácia používa vnútorné mechanizmy na určenie miery podobnosti medzi vzorovým signálom a analyzovanou funkciou. Vo vlnkovej transformácií je daným vzorovým signálom funkcia zvaná vlnka, ψ. Vlnka (wavelet) je funkcia, ktorá začína v nulovom bode, jej funkčná hodnota postupne rastie až dosiahne maximálnych hodnôt a následne klesá až po nulový bod v ktorom začala [8]. Spojitá vlnková transformácia teda slúži na vyhľadávanie rôznych foriem vlniek, či už posunutých, komprimovaných, alebo natiahnutých. Príklad vlnkovej transformácie signálu je zobrazený na Obr. 1.



Obr. 1 Vlnková transformácia, posun a zmena mierky

Postup vlnkovej transformácie je zobrazený na Obr. 2 a je možné ho opísať týmito 5timi bodmi [17]:

1.) Základná vlnková funkcia sa porovná s časťou originálneho signálu.

2.) Vypočíta sa hodnota C, ktorá udáva, ako úzko súvisí vlnka s danou časťou signálu. Čím väčšia je absolútna hodnota čísla C, tým je väčšia podobnosť funkcií.

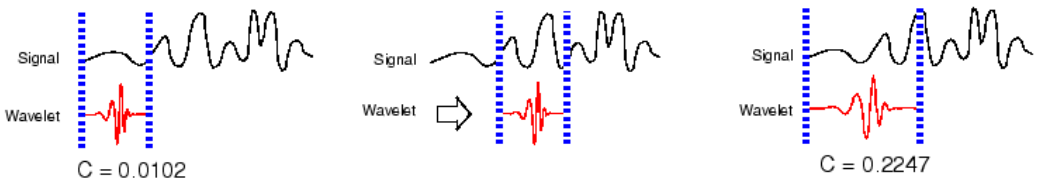
3.) Posun vlnky doprava a opakovanie bodov 1 a 2 pozdĺž celej dĺžky signálu.

4.) Zmena mierky vlnky a opakovanie bodov 1 až 3.

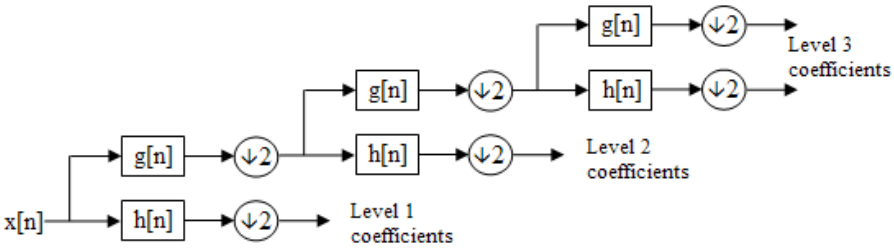
5.) Opakovanie bodov 1 až 4.

Tak ako pri Fourierovych transformáciách existuje veľké množstvo rôznych matematických modelov a transformácií, teória vlniek tiež pozostáva z veľkého množstva rozličných transformácií. Keďže v informatike sa pracuje výlučne s diskrétnymi hodnotami, zameriame sa na používanie skupiny diskrétnych vlnkových transformácií.

Diskrétne vlnkové transformácie je možné vypočítať pre 1 až n stupňov. Táto transformácia signálu *X* je vypočítaná prechodom signálu cez viacero dolno a horno priepustných filtrov, pričom dekompozícia signálu je pre *n*-tý stupeň transformácie je vypočítaná podľa schémy na Obr. 3.



Obr. 2 Postup vlnkovej transformácie - bod 2, 3 a 4



Obr. 3 Trojstupňové filtrovanie vlnkovej transformácie

2.7 Logistická regresia

V štatistike je vyjadrený pojem regresia ako spôsob vyjadrenia „priemernej“ závislosti dvoch alebo viacerých premenných, v podstate priebeh korelačnej závislosti.

2.7.1 Logistická regresia pre binárne dáta

Tento model je vhodný v prípade, keď cieľová premenná nadobúda jednu z len dvoch možných hodnôt reprezentujúcich úspech alebo zlyhanie, alebo viac všeobecne, prezenciu alebo absenciu daného javu. Logistická regresia je značne používaná v zdravotníctve, sociálno-vedných odboroch, ako aj v marketingových aplikáciách, napríklad pri predpovediach sklonu zákazníka ku kúpe výrobku alebo k ukončeniu predplatného.

Logistická regresia pre binárnu vysvetľovanú premennú odhaduje pomocou vysvetľujúcich premenných pravdepodobnosť že vysvetľovaná premenná bude 1. [32]

2.7.2 Model

Model predpokladá, že máme k dispozícii k nezávislých rovnako rozdelených pozorovaní $y_1,..,y_k$, a že i-te pozorovanie môžeme považovať za realizácie náhodnej premennej Y_i. Predpokladáme, že Y_i má binomické rozdelenie

$$Y_i \sim B(n_i, \pi_i)$$

s parametrom n_i a pravdepodobnosťou π_i. Pre individuálne (nezoskupené) data platí $n_i = 1$, pre všetky i.

Ďalej predpokladáme, že pravdepodobnosť π_i transformovaná pomocou funkcie *logit* je lineárnou funkciou prediktorov

$$logit(\pi_i) = \log \frac{\pi_i}{1-\pi_i} = \mathbf{x'_i}\boldsymbol{\beta}$$

kde $\mathbf{x'_i}$ je vektor vysvetľujúcich premenných a $\boldsymbol{\beta}$ je vektor príslušných regresných koeficientov. Regresné koeficienty $\boldsymbol{\beta}$ môžu byť interpretované v rovnakom duchu ako v lineárnych modeloch mysliac na to, že ľavá strana rovnice modelu je logit a nie stredná hodnota. β_j vlastne reprezentuje zmenu v logite pravdepodobnosti pri jednotkovej zmene j-tého prediktora držiac všetky ostatné prediktory konštantné.

Exponovaním rovnice získavame šance pre i-tu premennú

$$\frac{\pi_i}{1-\pi_i} = \exp\{\mathbf{x'_i}\boldsymbol{\beta}\}$$

Toto je multiplikatívny model pre šance. Ak by sme chceli urobiť jednotkovú zmenu j-tého prediktora, šance by sa znásobili o $\exp(\beta_j)$. Na ukážku uvažujme lineárny prediktor $\mathbf{x'_i}\boldsymbol{\beta}$, ak zvýšime x_j o jednotku, získavame $\mathbf{x'_i}\boldsymbol{\beta} + \beta_j$. Exponovaním dostávame $\exp\{\boldsymbol{x'_i}\boldsymbol{\beta}\}$ krát $\exp\{\beta_j\}$. Teda exponovaný koeficient $\exp\{\beta_j\}$ reprezentuje *pomer šancí (odds ratio).* Preklad výsledkov do pomeru šancí je často užitočný, pretože môžeme narábať z viac zrozumiteľnou stupnicou(škálou) pri zachovaní relatívne jednoduchého modelu, čo uľahčuje interpretáciou výsledkov. [32]

Výpočtom z predchádzajúcej rovnice dostávame odhadovanú pravdepodobnosť π_i, ktorá má tvar.

$$\pi_i = \frac{\exp\{\mathbf{x'_i}\boldsymbol{\beta}\}}{1+\exp\{\mathbf{x'_i}\boldsymbol{\beta}\}}$$

3 Grafická analýza binárnych dát

Táto časť práce pojednáva o využití popísaných metód pri analýze komplexných dát v podobe súborov reprezentovaných v binárnej podobe. Aby bolo možné pomocou daných metód analyzovať obsah alebo typ dát, je nutné upraviť ich pôvodnú binárnu reprezentáciu do spracovateľnej podoby. Keďže väčšina spomínaných metód sa používa na spracovanie signálu, binárne dáta zvyčajne reprezentujeme v dvojrozmernej sústave, kde na osi Y bude reprezentovaná samotná hodnota bitu (skupiny bitov) a os X je imaginárna časová os vyjadrujúca postupnosť (usporiadanie) dát v analyzovanom súbore.

Využitím tohto typu analýzy je možné identifikovať obsah alebo typ prenášaných dát bez nutnosti vyhľadávania známych signatúr alebo magických čísel. Hlavným princípom takejto analýzy je využitie skutočnosti, že všetky údaje vyskytujúce sa v informačných systémoch majú určitú neusporiadanú podobu, ktorá síce záleží na obsahu dát, ale z vonkajšieho pohľadu je ovplyvnená hlavne postupnosťou určitých dátových blokov, prípadne opakovaných výskytoch časti reprezentovanej znakmi alebo postupnosťou znakov.

Pomocou analýzy veľkého množstva dát rovnakého typu je možné vytvoriť imaginárne modely vonkajšej štruktúry, aj keď nepoznáme ich presný obsah, ale poznáme len ich typ. Následne je možné úplne bez hlbšej analýzy obsahu neznámych dát len pomocou matematických postupov a grafickej analýzy stanoviť ich zaradenie. Tento prístup má veľký význam a uplatnenie najmä z hľadiska bezpečnosti operačných systémov a sieťovej prevádzky. Vďaka využitiu tohto postupu je možné identifikovať väčšinu dát bez potreby analýzy hlavičky súboru, alebo potreby hľadania známych reťazcov. Využitie tohto postupu je potrebné napríklad v prípade, keď útočník pomocou malých zmien v obsahu súboru dokáže zmiasť antivírový program, pričom tieto malé zmeny nemusia mať dosah na skúmanú vonkajšiu štruktúru a súbor by nebol označený ako hrozba (viď napr. [28]).

Z hľadiska bezpečnosti funguje väčšina doteraz používaných ochranných mechanizmov na princípe vyhľadávania už identifikovaných signatúr v obsahu prenášaných dát. Preto predstavuje veľké riziko každá nová bezpečnostná hrozba, ktorá ešte nebola identifikovaná a zväčša je ťažké jej zamedziť v krátkom čase po jej prvom výskyte. Ako príklad môže byť použitý výskyt nového typu vírusu, ktorý ešte nebol

identifikovaný žiadnou entitou a signatúra, podľa ktorej by bolo možné overiť a zistiť výskyt tejto hrozby, ešte nebola pridaná do zabezpečovacích systémov. Táto situácia je tiež nazývaná nultý deň útoku (Zero-Day). [27]

Pri postupe, kde je využitá analýza vonkajšej štruktúry dát bolo možné jednoducho zabrániť takémuto typu útoku, pretože každý vírus je v podstate iba určitý typ spustiteľného súboru, alebo zdrojového kódu, ktorý má podobnú štruktúru ako väčšina iných spustiteľných súborov bez ohľadu na signatúru, funkciu alebo dáta obsiahnuté v ňom. Po identifikovaní známeho typu súboru môže byť užívateľ, prípadne poskytovateľ danej služby upozornený na výskyt daného typu dát, pričom následne je jednoduché určiť, či ide o bezpečnostnú hrozbu (napríklad vyžiadaním potvrdenia užívateľa či je daný typ vyžiadaný alebo nie). Ďalšou možnosťou je identifikácia samotného škodlivého softvéru pomocou vopred naučeného vzoru štruktúry, pričom vychádzame z predpokladu istej podobnosti medzi dátovými segmentmi obsiahnutými v analyzovanom súbore.

3.1 Binárna reprezentácia dátového obsahu

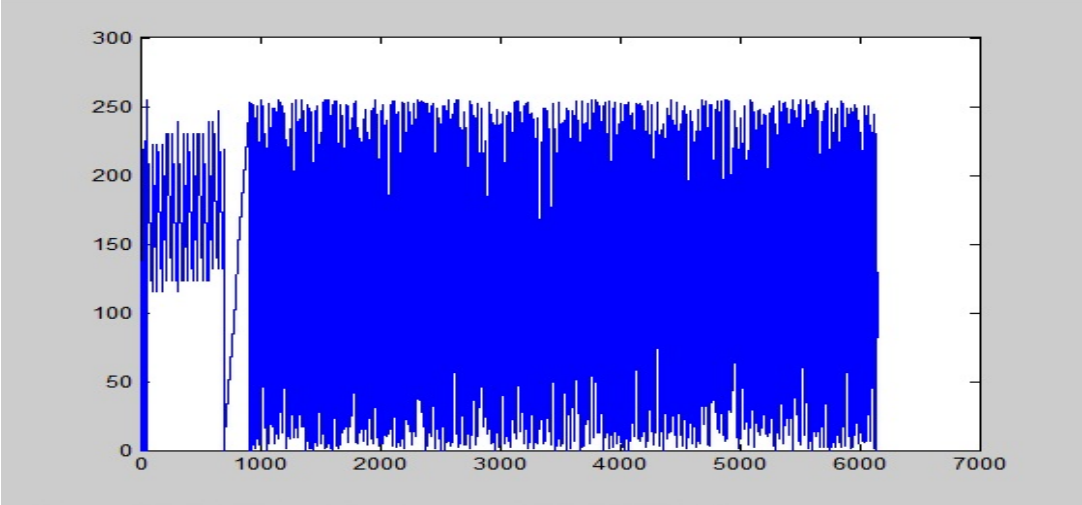
Každý súbor alebo iná forma dátovej reprezentácie je v informačných systémoch reprezentovaná v binárnej podobe. Keďže na analýzu dát reprezentovaných v tvare vlnovej funkcie (signálu) je potrebné, aby mala vstupná veličina viac stavov, na získanie informačnej hodnoty o danom vstupe nie je možné použiť binárnu reprezentáciu. Preto je potrebné používať reprezentácie dát vyšších rádov.

Ako najvhodnejšia reprezentácia sa javí najmenšia dátová jednotka - skupina 8 bitov. Dátový blok 1 bajtu je totiž rozložiteľný na 2 znaky v hexadecimálnej sústave, pričom 2 hexadecimálne znaky reprezentujú jeden ASCII znak. Táto reprezentácia je vhodná aj preto, že veľké množstvo dátových typov je reprezentovaných práve postupnosťou ASCII znakov, prípadne pozostáva z RGB údajových blokov reprezentovaných taktiež hodnotami od 0 až po 255. Vytvorením postupnosti bajtov vznikne pri grafickej reprezentácii forma vlnovej funkcie, na ktorú môžeme aplikovať transformácie a matematické metódy.

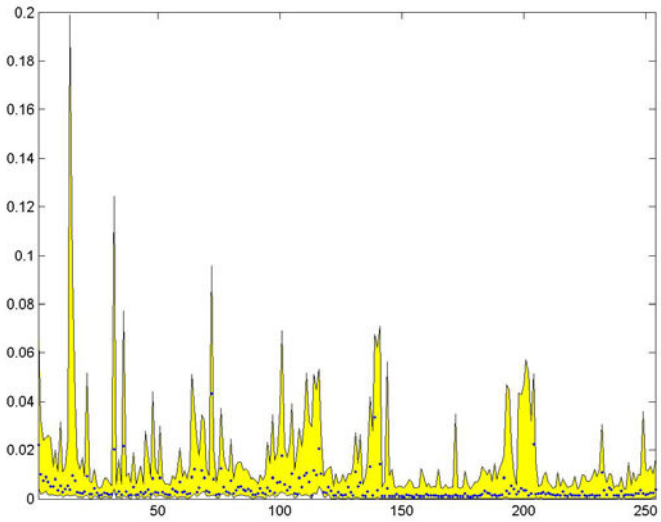
Príklad reprezentácie dát v tejto podobe je zobrazený na Obr. 4 (hexadecimálna) a Obr. 5 (grafická reprezentácia).

```
0000f6a0  20 30 20 52 20 2f 46 69 74 20 5d 0a 3e 3e 0a 65  | 0 R /Fit ].>>.e|
0000f6b0  6e 64 6f 62 6a 0a 38 20 30 20 6f 62 6a 0a 3c 3c  |ndobj.8 0 obj.<<|
0000f6c0  2f 43 72 65 61 74 6f 72 20 28 32 35 30 29 0a 2f  |/Creator (250)./|
0000f6d0  50 72 6f 64 75 63 65 72 20 28 32 35 30 29 0a 2f  |Producer (250)./|
0000f6e0  43 72 65 61 74 69 6f 6e 44 61 74 65 20 28 44 3a  |CreationDate (D:|
0000f6f0  32 30 31 30 30 35 31 38 31 35 33 31 31 37 2b 30  |201005181531l7+0|
0000f700  32 27 30 30 27 29 0a 3e 3e 0a 65 6e 64 6f 62 6a  |2'00').>>.endobj|
0000f710  0a 78 72 65 66 0a 30 20 39 0a 30 30 30 30 30 30  |.xref.0 9.000000|
0000f720  30 30 30 30 20 36 35 35 33 35 20 66 20 0a 30 30  |0000 65535 f .00|
0000f730  30 30 30 30 30 30 30 39 20 30 30 30 30 30 20 6e  |00000009 00000 n|
0000f740  20 0a 30 30 30 30 30 36 32 36 39 39 20 30 30 30  | .0000062699 000|
0000f750  30 30 20 6e 20 0a 30 30 30 30 30 36 32 37 32 30  |00 n .0000062720|
0000f760  20 30 30 30 30 30 20 6e 20 0a 30 30 30 30 30 36  | 00000 n .000006|
```

Obr. 4 Reprezentácia dát v hexadecimálnej (v strede) a ASCII podobe (vpravo)



Obr. 5 Grafická reprezentácia postupnosti jedno-bajtových blokov



Obr. 6 Analýza početnosti výskytu prvkov

3.2 Analýza dát určením početnosti výskytu znakov

Jednou z najjednoduchších, ale zároveň aj najľahšie ovplyvniteľných metód analýzy obsahu dát je určenie početnosti výskytu obsiahnutých prvkov. Základom analýzy je predpoklad, že v určitých typoch súborov sa vyskytujú určité prvky viackrát ako v iných a početnosť ich výskytu sa nemení náhodne vo veľkej miere.

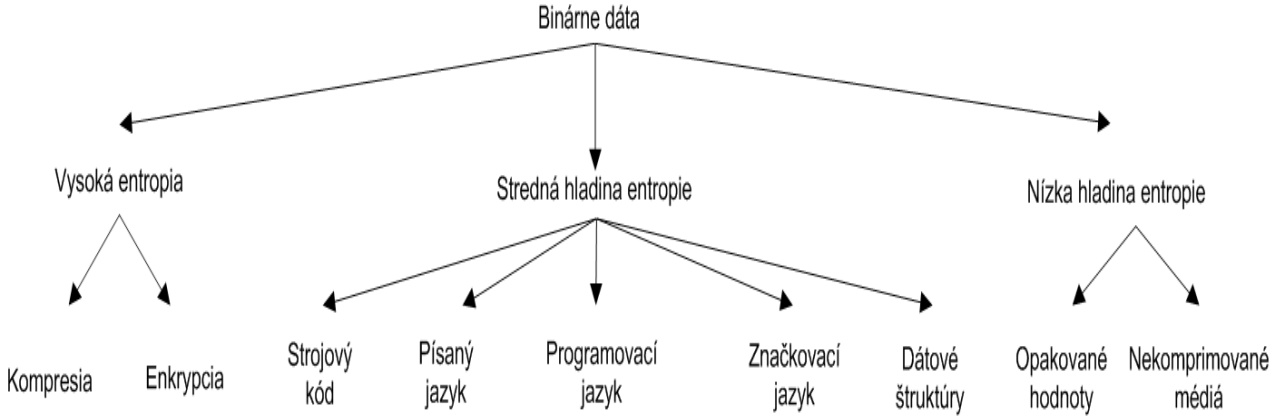
V našom prípade sa každý analyzovaný súbor skladá z postupnosti hodnôt 0 – 255, čo znamená, že každú jednu hodnotu môžeme brať ako samostatný prvok a zobraziť histogram početnosti výskytu každého prvku. Na Obr. 6 je zobrazená početnosť prvkov v súbore, pričom za samostatný prvok sa považuje decimálna hodnota jedného bajtu údajov. Analýzou veľkého množstva známych typov súborov je touto metódou možné vytvoriť modely, ktoré popisujú zastúpenie každého jedného prvku s maximálnou povolenou odchýlkou hodnoty. [24]

Problémov pri použití tejto metódy je niekoľko. Prvým problémom pri analýze súborov je reprezentácia početnosti. Keďže veľkosť testovaných dát je vždy rozdielna, jedinou možnosťou na ich analýzu je vyjadrenie početnosti prvkov percentuálne. Tento druh analýzy však môže byť neúčinný pri jednoduchom vložení neutrálnych dát (napr. NOP), pričom dôjde ku zmene pomeru početností výskytov prvkov. Ďalším problémom je kompresia súborov alebo časti dát pričom dochádza ku zámene prvkov a ku zmene ich početnosti výskytu.

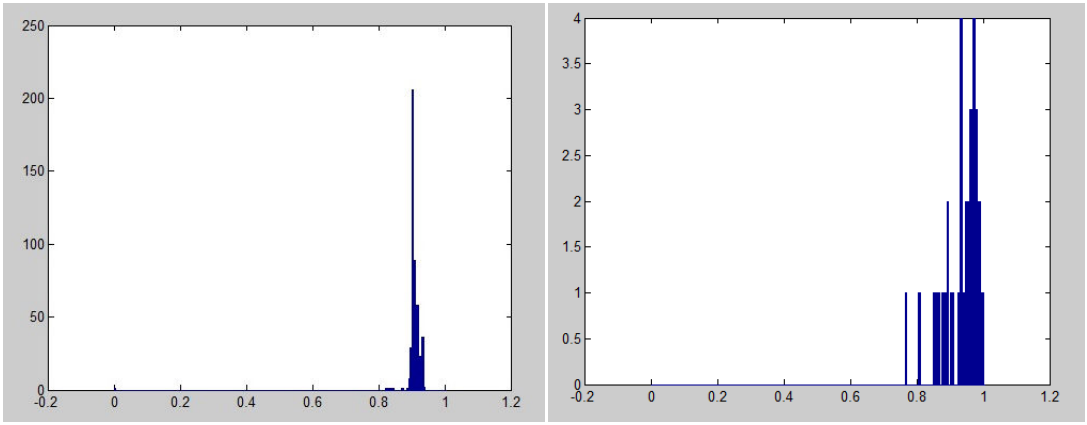
3.3 Analýza dát pomocou výpočtu entropie

Analýzou entropie binárnych dát sa venovalo už viacero vedeckých prác s rôznym zameraním, od forenznej analýzy diskov až po využitie entropie v kryptografii. Dnes je teda už možné podľa hladiny entropie vymedziť primitívne dátové typy, ktoré sa v rôznych pomeroch nachádzajú v analyzovaných súboroch. Príklad hierarchického rozdelenia primitívnych dátových typov je uvedený na Obr. 7 [16].

Na testovacích vzorkách dát je možné názorne reprezentovať priemernú hodnotu entropie rôznych analyzovaných typov súborov. Na Obr. 8 je zobrazený histogram priemerných hodnôt entropie na súboroch typu JPG a GIF, ktoré obsahujú dáta s pomerne veľkou úrovňou kompresie. Z daného grafického zobrazenia vidíme, že hodnoty entropie tohto typu dát sa pohybujú na vysokej úrovni.

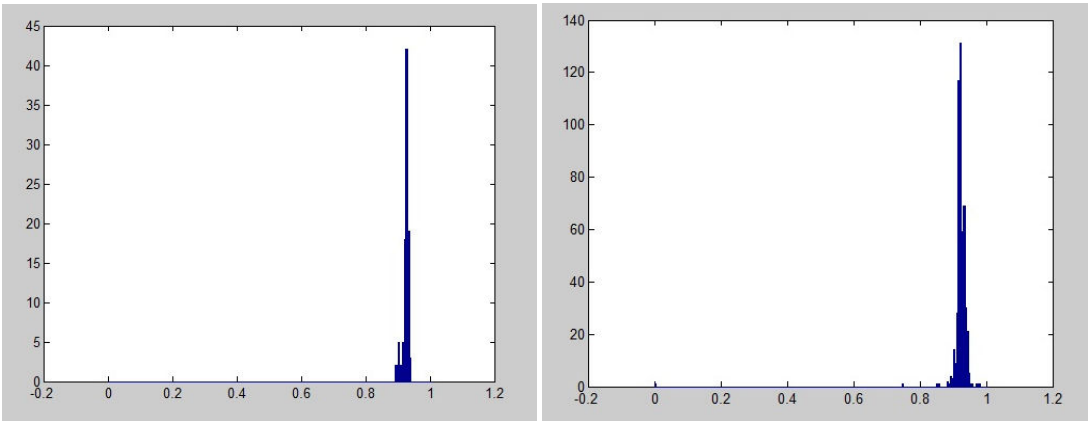


Obr. 7 Hierarchia primitívnych dátových typov



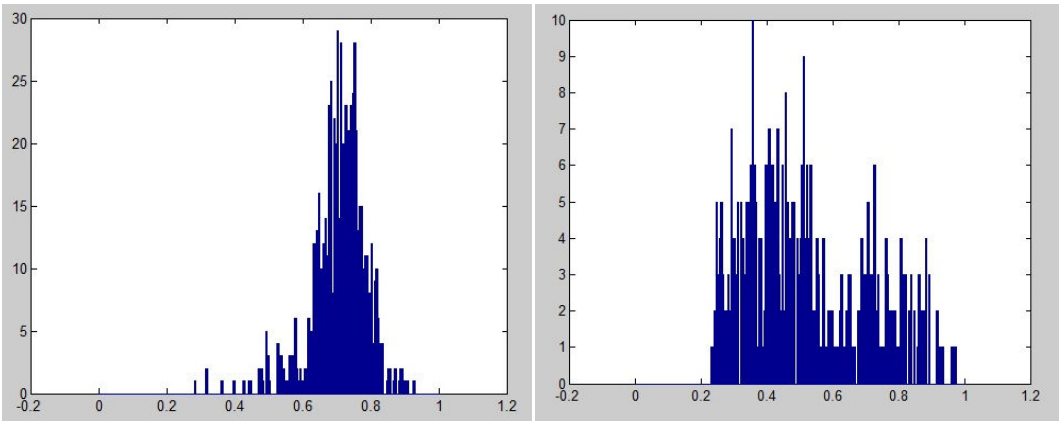
Obr. 8 Histogram entropie JPG (vľavo) a GIF súborov

Veľmi podobný interval hodnôt môžeme vidieť aj pri analýze ďalších typov súborov, ktorých dáta pozostávajú z komprimovaných blokov. Medzi ne patria napríklad súbory typu MP3, ZIP, RAR prípadne PDF. Pri všetkých predchádzajúcich typoch súborov bola analýzou zistená vyššia hodnota entropie. Na Obr. 9 je zobrazený histogram priemerných hodnôt entropie na súboroch typu MP3 a PDF.

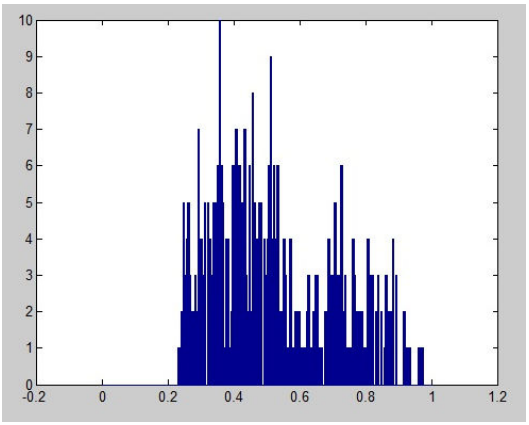


Obr. 9 Histogram entropie MP3 (vľavo) a PDF súborov

Druhou skupinou súborov sú súbory obsahujúce niektorú z foriem jazyka, od programovacích jazykov, cez písaný text až po strojový kód. Tieto súbory sa vyznačujú strednou úrovňou entropie. Ako príklad tejto skupiny súborov môžeme uviesť histogram priemerných hodnôt entropie súborov typu EXE (vľavo) a DOC na Obr. 10 a typu JAR na Obr. 11, kde je reprezentovaná široká škála hodnôt entropie so strednou úrovňou.



Obr. 10 Histogram entropie EXE (vľavo) a DOC súborov



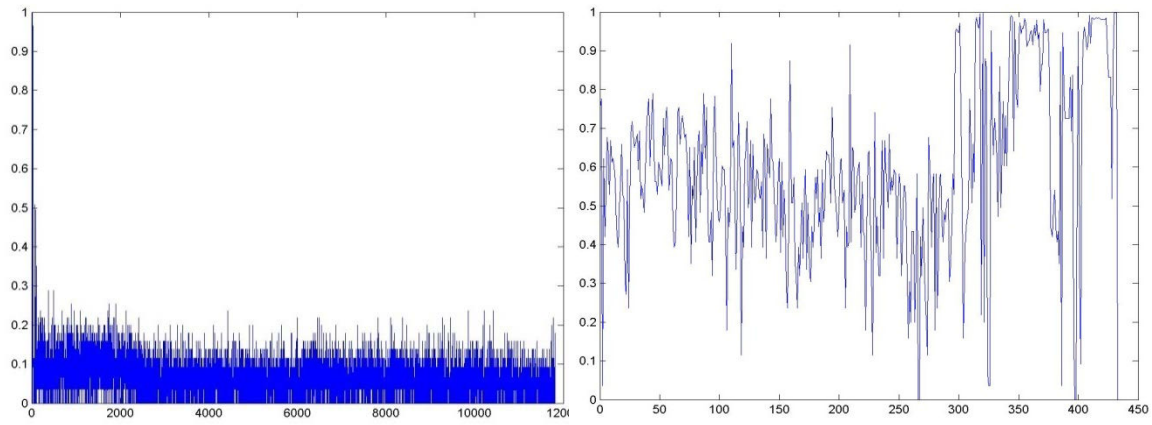
Obr. 11 Histogram entropie JAR súborov

3.3.1 Analýza entropie použitím statickej veľkosti okna

Výpočet hodnoty entropie na určitých dátach ako celku napríklad na celom súbore neposkytuje dostatočnú informačnú hodnotu z hľadiska analýzy typu alebo ich obsahu. Túto metódu nie je možné zväčša použiť kvôli skutočnosti, že takmer všetky dáta využívané v informačných technológiách obsahujú periodicky usporiadané ale aj neperiodické dáta a to v rôznom pomere a rozložení aj pri rovnakom type dát. [26]

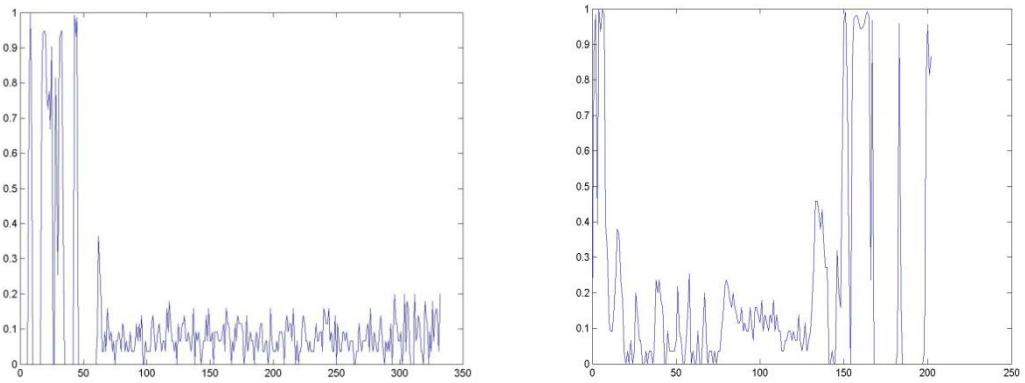
Preto je na využitie tejto vlastnosti dát z hľadiska analýzy potrebné rozdeliť spracovávané dáta do menších častí a na týchto menších intervaloch vypočítať ich entropiu. Kvôli zvýšeniu presnosti metódy je možné brať do úvahy daný interval aj s určitým presahom do nasledujúceho a predchádzajúceho intervalu, čím sa minimalizuje rozdiel hodnôt entropie susedných častí. Konečná grafická reprezentácia je spojením všetkých vypočítaných čiastkových entropií. Príkladom takejto grafickej reprezentácie dát s vysokou neusporiadanosťou je výsledok analýzy JPG súboru na Obr. 12 vľavo, vpravo je zobrazená reprezentácia vstupného súboru dát s výskytom usporiadaných dát – súboru EXE. [12]

V použitom vývojovom prostredí je funkcia entropie implementovaná obrátenou hodnotou klasickej matematickej entropie, čo znamená že pri najväčšej neusporiadanosti prvkov postupnosti je výsledná hodnota funkcie blízka alebo rovná nule a naopak pri zvýšení usporiadanosti dát je hodnota vyššia, prípadne blížiaca sa hodnote 1. Kvôli nutnosti použitia klasickej matematickej entropie bolo potrebné v použitom vývojovom prostredí implementovať vlastnú funkciu, preto budú príklady výpočtu entropie v nasledujúcich kapitolách už reprezentované v správnom tvare.



Obr. 12 Reprezentácia entropie JPG (vľavo) a EXE súboru

Pri využití tejto analýzy vzniká problém s dátami, na ktorých nie je úplne jasne rozoznateľný rozdiel medzi rôznymi zložkami, prípadne sa zložky regulárnych dát prekrývajú so zložkami iného typu, napr. kódu v spustiteľných súboroch. Tieto prekrytia sa vyskytujú u veľkej časti testovaných typov dát a preto nie je možné len pomocou výpočtu samotnej entropie dostatočne presne stanoviť ich obsah prípadne typ. Na Obr. 13 je príklad dát, pri ktorých nie je možné určiť o aké údaje sa jedná. [18]



Obr. 13 Dve reprezentácie dát s rôznorodou entropiou

3.3.2 Analýza entropie použitím pohyblivej veľkosti okna

Pri výpočte entropie súboru použitím funkcie okna je dôležité vhodne zvoliť túto veľkosť. Keďže sa táto práca zameriavala hlavne detekciou škodlivého softvéru v podobe binárnych spustiteľných súborov, hodnota veľkosti okna bola experimentálne stanovená na pevnú hodnotu 8 bajtov (8 znakov). Pri tejto veľkosti okna boli zaznamenané najvýraznejšie odlišnosti výstupných hodnôt analýzy pri porovnávaní súborov typu malware a iných typov súborov ako napríklad obrázkov, komprimovaných dát, prípadne súborov obsahujúcich písaný text. Pri analýze súborov iného typu by bolo potrebné experimentálne stanoviť príslušnú hodnotu, ktorá sa bude pravdepodobne líšiť od nami stanovenej. Ďalšou možnosťou by bolo použitie pohyblivej veľkosti okna, čo zvýši informačnú hodnotu výpočtu entropie binárnych. [25] Pohyblivá veľkosť okna môže byť použitá dvoma spôsobmi.

Prvým spôsobom je určenie veľkosti pred začatím samotnej analýzy, pričom veľkosť okna môže byť určená z určitej veličiny získanej predchádzajúcou analýzou. Ako príklad tejto veličiny môžeme použiť stanovenie veľkosti okna podľa veľkosti samotného súboru. Táto hodnota sa následne počas analýzy nemení, jediné, čo sa mení je pozícia samotného okna.

Druhým spôsobom je dynamická zmena veľkosti okna počas analýzy. Princíp tejto analýzy je založený na sledovaní zmeny entropie postupne naprieč celou dĺžkou analyzovaných dát. Namiesto posunu samotného okna sa pri každej iterácii analýzy okno zväčšuje n-násobne, pričom počiatočná pozícia okna ostáva rovnaká, až kým veľkosť okna nedosiahne veľkosť celého súboru. V konečnom dôsledku dostávame grafickú reprezentáciu narastajúcej hladiny entropie naprieč analyzovaným súborom.

Keďže sa postupne zväčšuje veľkosť okna, vstupné pole na výpočet entropie obsahuje viacej vstupných prvkov a hodnota entropie sa tiež mení. Výsledkom tejto analýzy, podľa ktorej môžeme určiť príslušný typ dát, je pole koeficientov rastu alebo klesania hodnôt výslednej krivky.

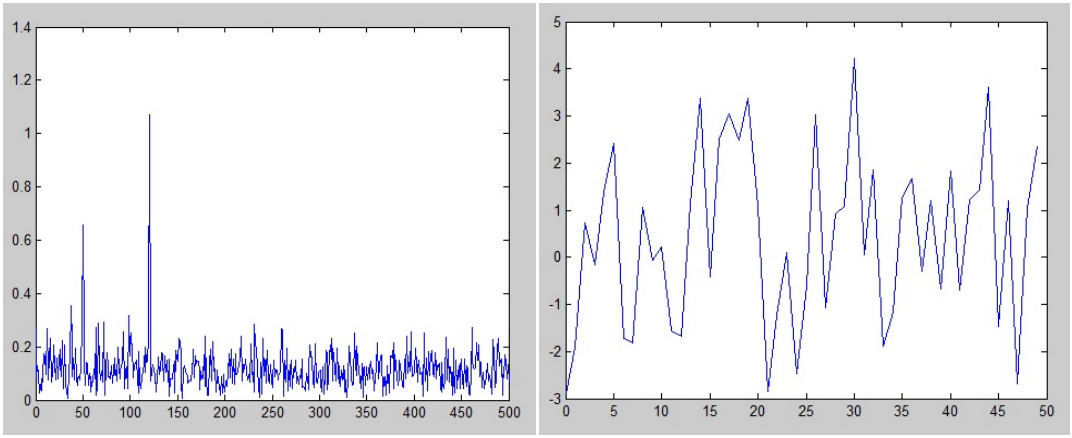
3.4 Analýza dát pomocou výpočtu frekvenčnej závislosti

Pri tomto type analýzy využívame periodickosť dát, pričom vyhľadávame dáta alebo ich zoskupenia, ktoré sa s určitou frekvenciou vyskytujú a opakujú. Ich frekvencia a amplitúda sú čiastočne závislé ako od štruktúry obsiahnutých dát, tak aj od typu celého analyzovaného súboru. Tento druh analýzy môžeme taktiež využiť pri vyhodnocovaní výsledkov predchádzajúcich analýz ako napríklad analýzy entropie súboru. Základnými nástrojmi, ktoré používame pri frekvenčnej analýze sú analýza pomocou Fourierových transformácií a analýza pomocou vlniek.

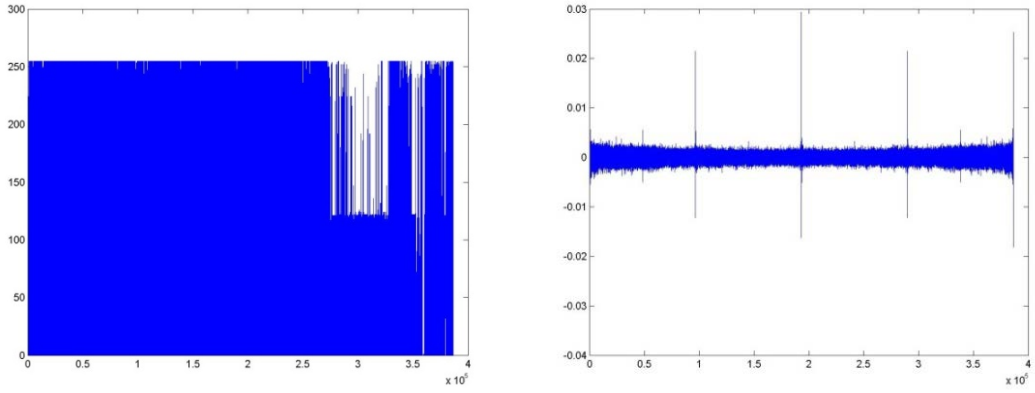
3.4.1 Analýza dát s použitím rýchlej Fourierovej transformácie

Keďže podstatou tejto matematickej metódy je získanie a vykreslenie frekvenčného spektra obsiahnutého vo vstupných dátach, je pomocou nej možné analyzovať periodickosť alebo určitú usporiadanosť v na prvý pohľad neusporiadanej postupnosti hodnôt. Na ilustráciu aplikácie rýchlej fourierovej transformácie môže slúžiť zobrazenie vstupných a výstupných dát na jednoduchej vlnovej funkcii na Obr. 14. Každá usporiadanosť alebo periodickosť vstupných dát sa zobrazí vo výstupnej funkcií ako špička na určitom frekvenčnom pásme. Čím výraznejšia je usporiadanosť výskytu istej postupnosti hodnôt vo vstupnej vlnovej funkcií, tým je fáza danej špičky výraznejšia na výslednom frekvenčnom spektre.

Pri analýze veľkého množstva dát pomocou rýchlej Fourierovej transformácie sa frekvenčné spektrum zahusťuje, avšak významná symetria alebo usporiadanosť vstupných dát je stále jasne odlíšiteľná podľa špičiek na výstupnom frekvenčnom spektre. Jasné usporiadanie dát je možné rozpoznať z výsledku Fourierovej transformácie na Obr. 15, kde bola použitá rýchla Fourierova transformácia na spustiteľný súbor reprezentovaný postupnosťou skupín ôsmich bitov.



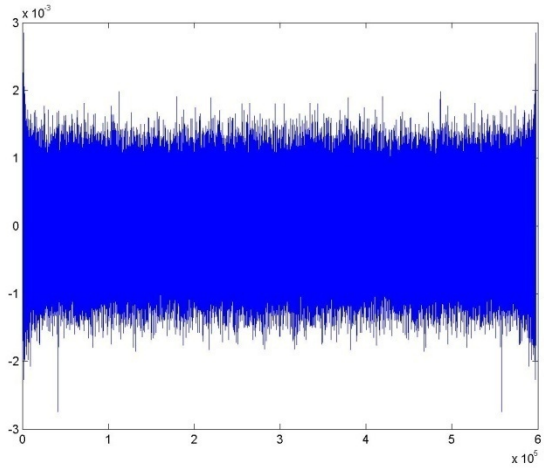
Obr. 14 Fourierova transformácia: vstupná vlnová funkcia (vľavo) a výsledné frekvenčné spektrum (vpravo)



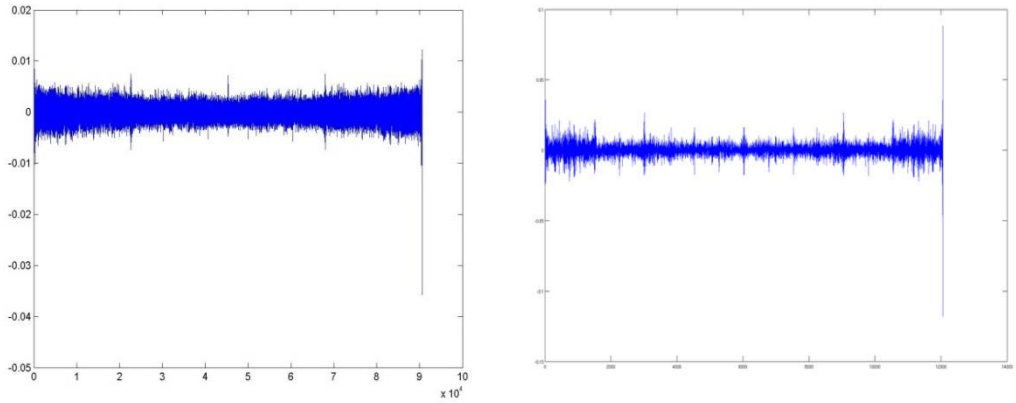
Obr. 15 Vstupné binárne dáta (vľavo) a výsledné frekvenčné spektrum (vpravo)

Naproti tomu pri použití Fourierovej transformácie na dáta, ktoré by mali byť v princípe neusporiadané a ich hodnota by nemala byť závislá na čase, respektíve je veľmi malá pravdepodobnosť opakovaného periodického výskytu určitej časti, je výsledkom husté frekvenčné spektrum. V tomto spektre nie je možné identifikovať vrchol alebo viac vrcholov grafu ktoré by sa výrazne líšili od ostatných. Takýmto prípadom je aj Fourierova transformácia nad bitovou reprezentáciou textového súboru vo formáte PDF, ktorý je zobrazený na Obr. 16, pretože tento formát obsahuje komprimované časti. Čím je kompresia vyššia, tým vyššia je aj entropia výsledného súboru a frekvenčná závislosť jeho zložiek nižšia.

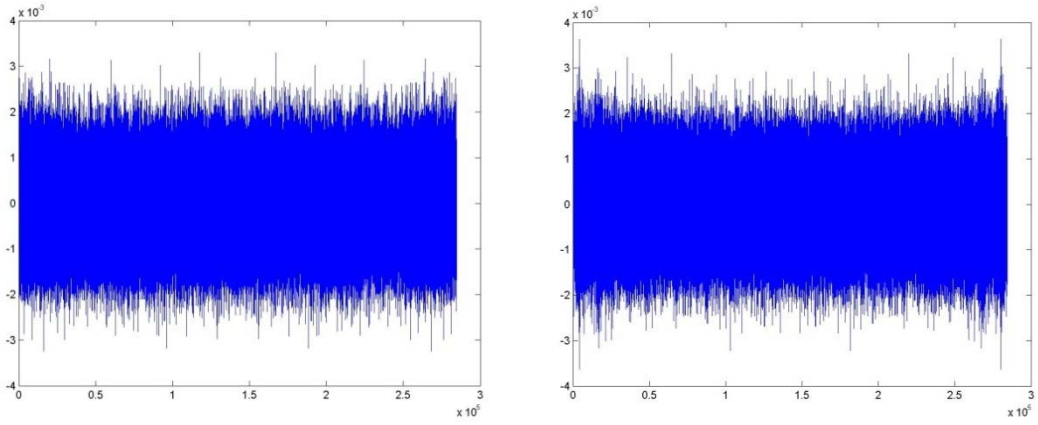
Ďalšie príklady použitia Fourierovej transformácie sú zobrazené na Obr. 17 (binárny kód spustiteľného súboru) a Obr. 18 (vľavo súbor vo formáte JPG, vpravo GIF) pričom je znova graficky ľahko odlíšiteľné či sa jedná o náhodné alebo usporiadané dáta.



Obr. 16 Výsledné frekvenčné spektrum PDF súboru



Obr. 17 Dva príklady transformácie spustiteľného súboru

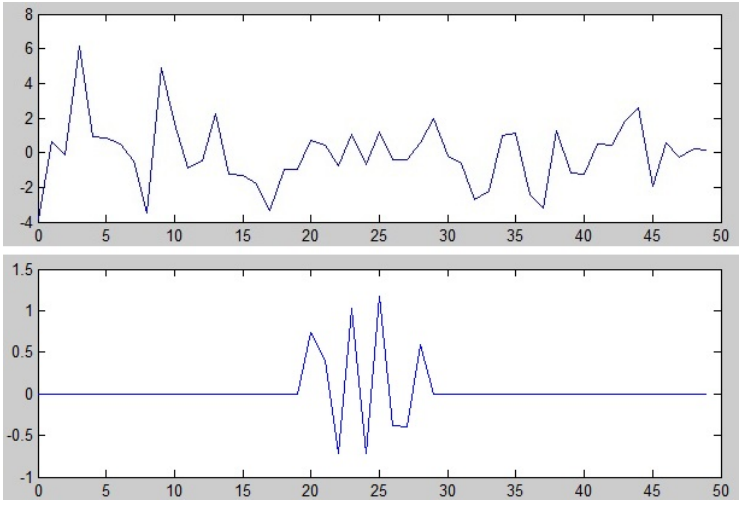


Obr. 18 Transformácia JPG (vľavo) a GIF súboru

3.4.2 Analýza dát použitím krátkodobej Fourierovej transformácie

Problémom použitia rýchlej Fourierovej transformácie je fakt, že pri aplikácií na väčšiu dĺžku vstupného intervalu dát rastie aj množstvo a hustota vrcholov výsledného frekvenčného spektra. Táto vlastnosť je nežiadúca z hľadiska analýzy vlastností vstupnej veličiny. Preto je vhodné rozdeliť analyzované dáta na menšie časti, na ktorých je možné aplikovať Fourierovu transformáciu len pre vybranú časť intervalu a získať tak frekvenčné spektrum, ktoré je možné ľahšie rozložiť a analyzovať, pretože pri menšej vzorke dát je ľahšie izolovať vzorky daného spektra a preto je nutné na analýzu použiť rozloženie vstupného intervalu. Na spojenie výsledkov čiastkových transformácií je následne potrebné použiť vyššie stupne grafickej reprezentácie ako je grafická reprezentácia dát pomocou dvojrozmernej sústavy. Najvhodnejšou formou takéhoto grafického znázornenia je spektrogram, pomocou ktorého je možné znázorniť jediný graf vytvorený zo spojenia čiastkových transformácií. Na výpočet hodnôt spektrogramu bola použitá ďalšia forma Fourierovej transformácie, krátkodobá Fourierova transformácia.

Krátkodobá fourierova transformácia využíva na transformáciu celého vstupného súboru po častiach matematickú funkciu nazvanú okno, ktorá je postupne aplikovaná pozdĺž celého intervalu. Príklad použitia funkcie okna na vstupné dáta je zobrazený na Obr. 19. [6]

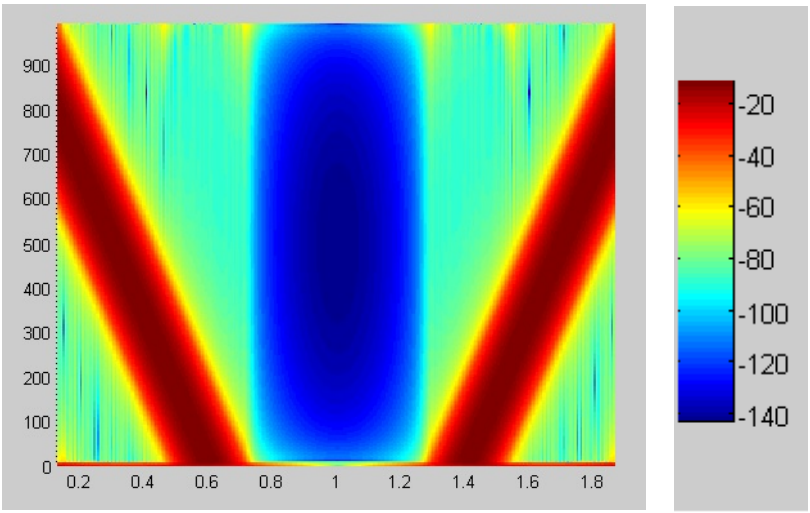


Obr. 19 Použitie funkcie okna

Pri využití tohto druhu analýzy dát vznikajú rôzne problémy, ktoré znižujú presnosť použitej metódy. Pri každej frekvenčnej analýze dát po častiach je nutné počítať s určitými stratami a nepresnosťami vznikajúcimi na hranách analyzovaných

častí. Keďže analyzujeme daný súbor dát ako celok a závislosť dát sa môže vyskytovať naprieč celou dĺžkou definičného oboru, pri rozdelení na menšie celky môže s väčšou alebo menšou pravdepodobnosťou dôjsť k zníženiu výpovednej hodnoty výsledku. Tento problém je možné čiastočne riešiť pomocou použitia prekrývania dvoch susedných intervalov, čo v konečnom dôsledku znižuje nepresnosť analýzy. Veľkosť prekrytia by mala byť zvolená s prihliadnutím na žiadaný pomer rýchlosti a presnosti výpočtu.

Výsledným grafickým zobrazením oboru hodnôt krátkodobej fourierovej transformácie je spektrogram. Spektrogram je špeciálny typ grafickej reprezentácie dvojdimenzionálnej funkcie závislej na čase, pričom graf reprezentuje trojrozmernú maticu veličín a to času, frekvencie a fázy. Spektrogram môže byť realizovaný 2D zobrazením, kde frekvencia a čas predstavujú hodnoty X a Y a fáza je reprezentovaná farebnou škálou, alebo 3D zobrazením, kde sú reprezentované všetky hodnoty osami zobrazenia (x,y,z). Príklad zobrazenia vlnovej funkcie spektrogramom je na Obr. 20.



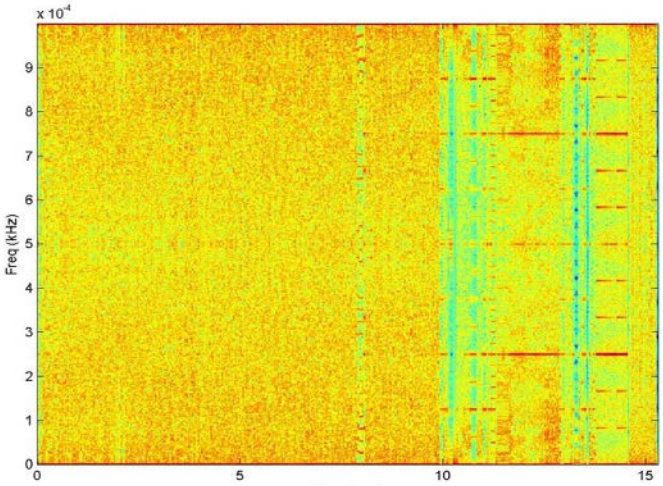
Obr. 20 Grafické zobrazenie pomocou spektrogramu

Z definície krátkodobej Fourierovej transformácie je zrejmé, že algoritmus použije na každý čiastkový interval osobitne rýchlu Fourierovu transformáciu, vypočíta frekvenčné spektrum a fázu každej časti spektra reprezentuje na spektrograme zmenou farby. Výhodou využitia tohto druhu fourierovej transformácie oproti diskrétnej transformácií je hlavne forma výstupu. Ak sa totiž v daných dátach vyskytuje požadovaná usporiadanosť, jej výskyt je limitovaný len na niektorú oblasť dát, prípadne je jej výskyt limitovaný určitou dĺžkou intervalu. Taktiež je možné jednoducho určiť frekvenčné pásmo, v ktorom sa vyskytuje hľadaná odlišnosť. Podľa týchto vlastností je možné s vysokou pravdepodobnosťou určiť o aký typ dát sa jedná, prípadne aký typ dát

sa nachádza v určitej časti dát (využitie napr. ak ide o dáta v počítačových sieťach, ktoré sú prenášané po častiach a nie v celku).

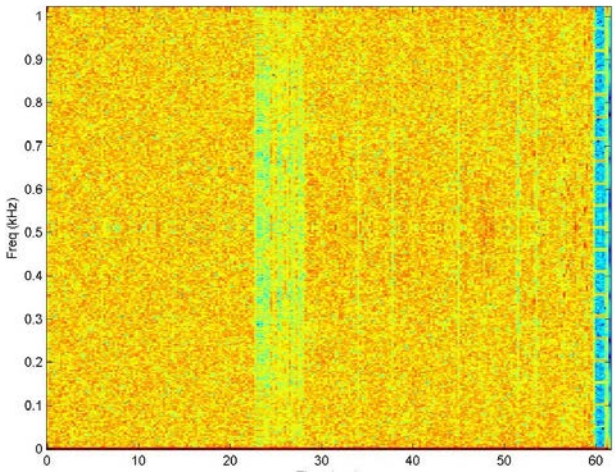
Obr. 21 znázorňuje použitie krátkodobej Fourierovej transformácie na spustiteľnom súbore. Na výslednom spektrograme je fáza označujúca neusporiadanosť vyznačená žltou farbou prechádzajúcou postupne do červeného spektra, a jasne rozlíšiteľná časť prechádzajúca do modrého farebného spektra predstavuje usporiadanosť v dátach. V zobrazenom prípade predstavuje táto časť výskyt strojového kódu v spustiteľnom súbore.

Týmto spôsobom analýzy dát je možné nájsť periodickosť alebo usporiadanosť v regulárnych dátach. Ako príklad je možné uviesť analýzu textového súboru vo formáte PDF, ktorý obsahuje periodickú časť textu. Spektrogram transformácie je zobrazený na Obr. 22 vpravo, vľavo je zobrazená časť obsahu daného súboru.



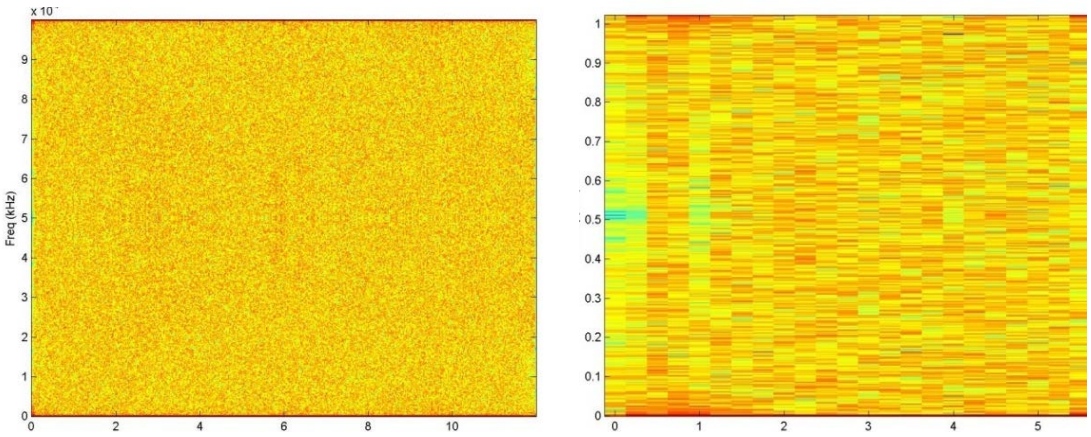
Obr. 21 Zobrazenie spektrogramu transformácie spustiteľného súboru

1. VAR
   var x : T
2. ARRAY
   Type Pole = array [1..n] of T
   From the intuitive specification: | x1:T | x2:T | … | xn:T |
   Constant size, static array
3. RECORD
   type T = record
   s1 : T1;
   s2 : T2;
   …
   sn : Tn
   end
4. RECORD
   type T = record s1 : T1, s2 : T2, …, sn-1 : Tn-1
   case sn : Tn of
   c1: (s11 : T11, s12 : T12, …, s1n1 : T1n1)
   c2: (s21 : T21, s22 : T22, …, s2n2 : T2n2)
   …



Obr. 22 Obsah PDF súboru

Analýzou regulárnych dát bez výskytu periodickosti dostaneme spektrogram zobrazený na Obr. 23, na ktorom je graf tvorený výhradne červeným a žltým farebným spektrom. Periodickosť je v tomto prípade viditeľná až pri zväčšení mierky grafu, kde sa na začiatku väčšiny dát vo forme súborov nachádza hlavička, ktorá obsahuje metadáta. Usporiadanosť začiatočnej časti súboru je zobrazená na Obr. 23 vpravo.

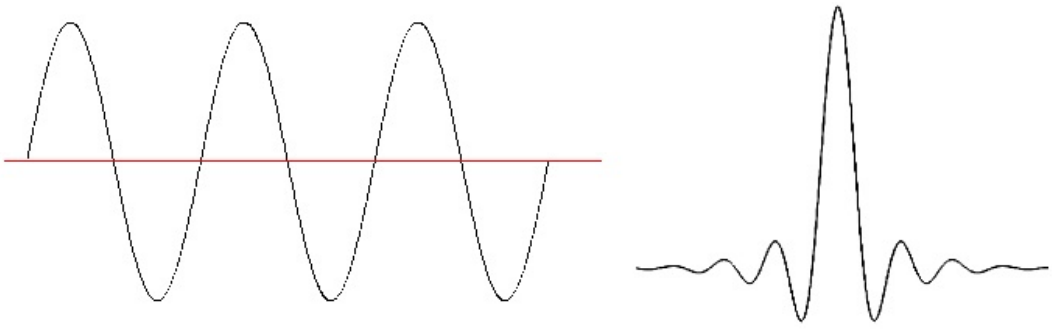


Obr. 23 Transformácia JPG súboru (vľavo) a detail hlavičky JPG (vpravo)

3.4.3 Analýza dát pomocou vlniek

Pri predchádzajúcej analýze sme sa zameriavali na použitie metód, ktoré sú založené na analýze sínusových a kosínusových funkcií začínajúcich a končiacich v nekonečne. Keďže v našom prípade sa jedná zväčša o nepravidelné krátke úseky údajov nadobúdajúce určitú vlnovú štruktúru, potrebujeme nástroj, ktorý zvládne analýzu vlnových funkcií s obmedzenou dĺžkou trvania.

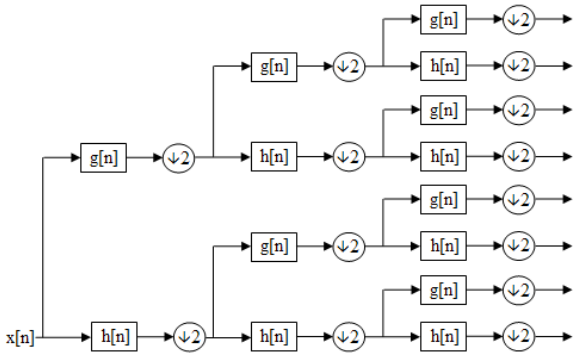
Takýmto nástrojom je analýza pomocou vlniek (wavelet analysis). Vlnka (wavelet) je funkcia, ktorá začína v nulovom bode, jej funkčná hodnota postupne rastie až dosiahne maximálnych hodnôt a následne klesá až po nulový bod v ktorom začala [8]. Porovnanie sínusovej vlny a vlnky na Obr. 24.



Obr. 24 Príklad sínusovej vlnovej funkcie a vlnky (meyer wavelet)

Tak ako Fourierove transformácie rozkladajú vstupný signál na viacero harmonických signálov, vlnková analýza rozkladá signál na viacero samostatných častí pôvodného signálu. Jednou z výhod vlnkovej transformácie oproti Fourierovým transformáciám je nepotrebnosť prekrytí pri analýze dát (signálu) po menších častiach, čo vyplýva už zo samotného princípu tejto analýzy.

Na analýzu dát pomocou vlniek sme vybrali z veľkého množstva nástrojov analýzu pomocou takzvaného vlnkového blokového rozkladu (Wavelet packet decomposition), ktorá je zároveň podobná krátkodobej Fourierovej transformácií. Táto transformácia síce patrí medzi diskrétne vlnkové transformácie, ale oproti samotnej diskrétnej vlnkovej transformácií (Discrete wavelet transformation) signál prechádza viacerými filtrami. Pri diskrétnej transformácií je každý stupeň vypočítaný len pomocou predchádzajúceho stupňa a jeho koeficientov, pričom pri vlnkovom blokovom rozklade je zostavený kompletný binárny strom s 2^n rôznymi skupinami koeficientov. Na Obr. 25 je zobrazená schéma vlnkového blokového rozkladu, kde *g[n]* a h*[n]* sú aproximačné koeficienty.

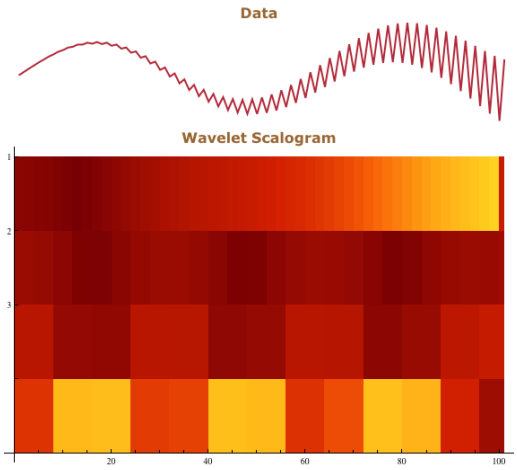


Obr. 25 Schéma vlnkového blokového rozkladu

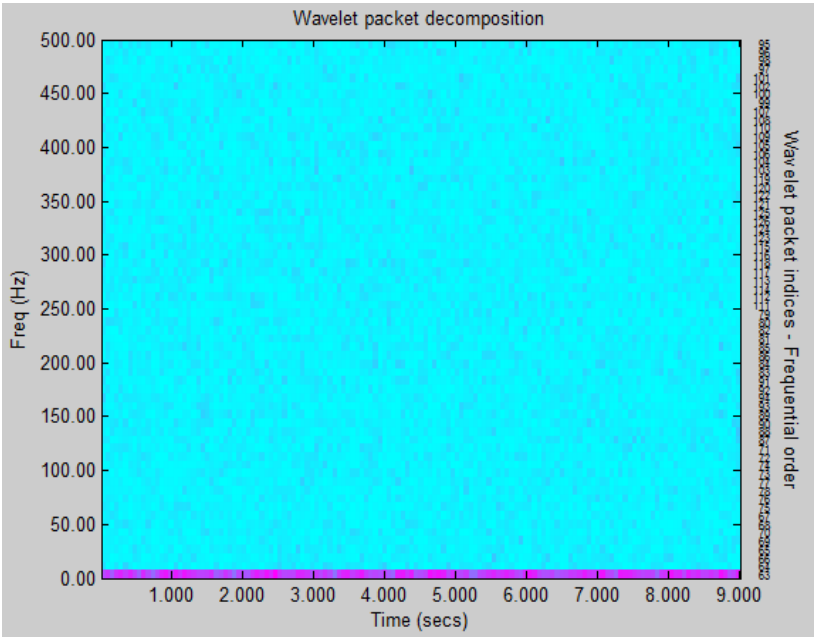
Na grafickú reprezentáciu vlnkovej transformácie sa používa vizuálna metóda scalogram [11], ktorá zobrazuje transformáciu pomocou troch osí. Os X predstavuje časovú os, os Y reprezentuje stupeň transformácie a os Z reprezentuje hodnotu vlnkového koeficientu. Os Z je zväčša reprezentovaná pomocou farebného odlíšenia, prípadne zmenou jasnosti. Príklad vlnkového scalogramu je na Obr. 26.

Pri analýze pomocou vlnkového blokového rozkladu sme podobne ako v predchádzajúcich frekvenčných analýzach skúmali vizuálne rozdiely grafov. Na Obr. 27 a Obr. 28 sú uvedené dva príklady vlnkovej transformácie. Na Obr. 27 je možné

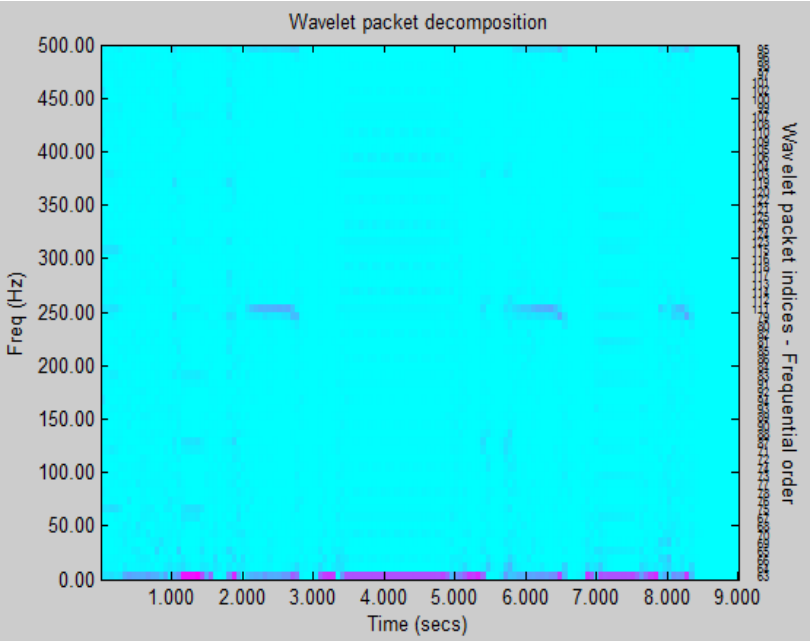
vidieť nesúrodé oblasti vyznačujúce sa nerovnomerne rozloženými tmavšími oblasťami grafu, naopak pri grafe transformácie spustiteľného súboru je takýchto nerovnomerných tmavých oblastí málo, prípadne je entropia ich výskytu nižšia.



Obr. 26 Príklad vlnkového skalogramu



Obr. 27 Príklad vlnkovej analýzy JPG súboru



Obr. 28 Príklad vlnkovej analýzy spustiteľného súboru

3.5 Analýza dát využitím viacnásobnej transformácie

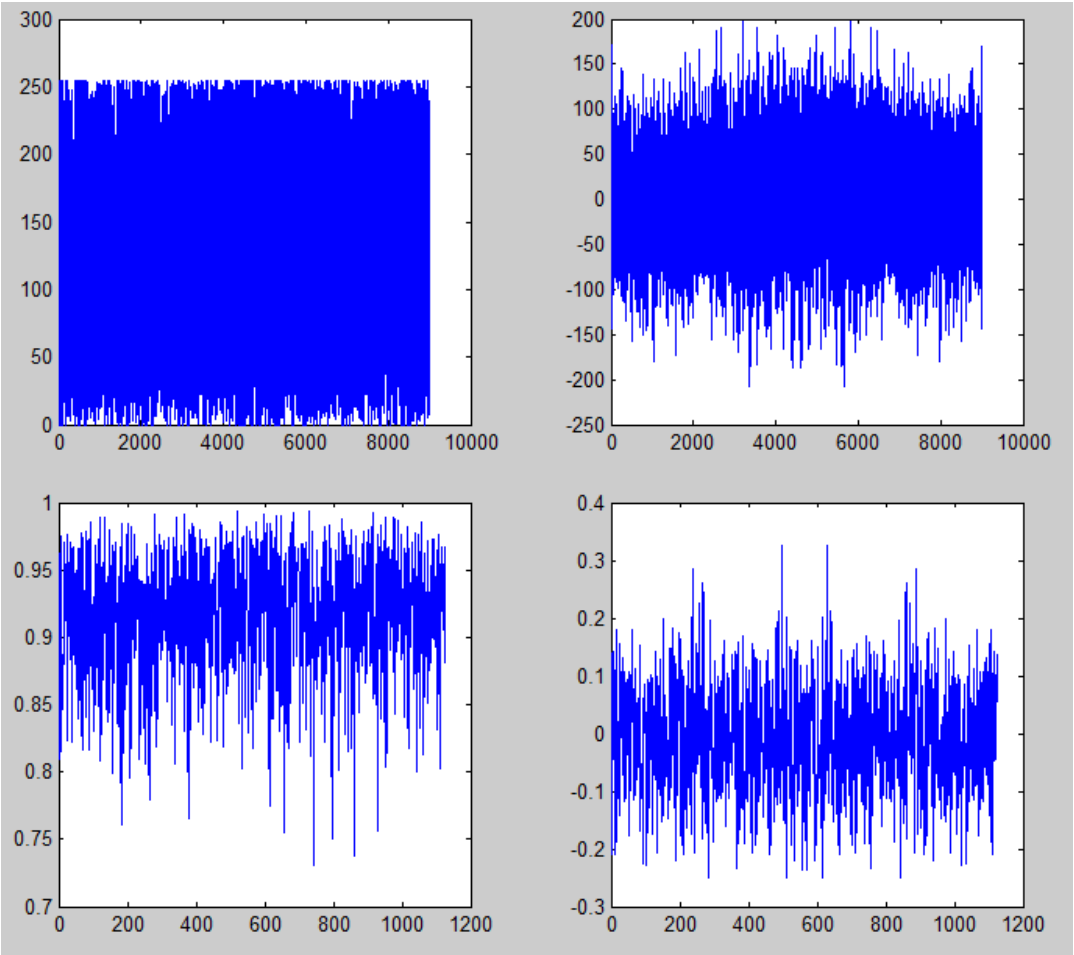
Pri používaní výpočtu entropie sme ukázali, že okrem vyjadrenia usporiadanosti alebo neusporiadanosti prvkov vstupného súboru dát dokáže entropia odkryť alebo zvýrazniť aj iné vlastnosti dát, ktoré by inak bolo ťažké získať a vyhodnotiť. Vďaka tejto vlastnosti sa výpočet entropie uplatnil vo viacerých odvetviach pri analýze dát od elektrotechniky, fyziky, matematiky až po informačné technológie.

V našom prípade analýzy dát využívame výpočet entropie ako základ pre ďalšiu frekvenčnú analýzu. V prvom bode analýzy sa vstupný súbor rozdelí na menšie časti, pričom z každej časti je vypočítaná hodnota entropie. Z takto vypočítaných hodnôt zostavíme nové pole, pričom sa môžeme zamerať len na výsledné hodnoty čiastkových výpočtov a pozorovať ich zmenu, alebo môžeme na tieto dáta aplikovať frekvenčnú analýzu (Fourierove transformácie, vlnkové transformácie).

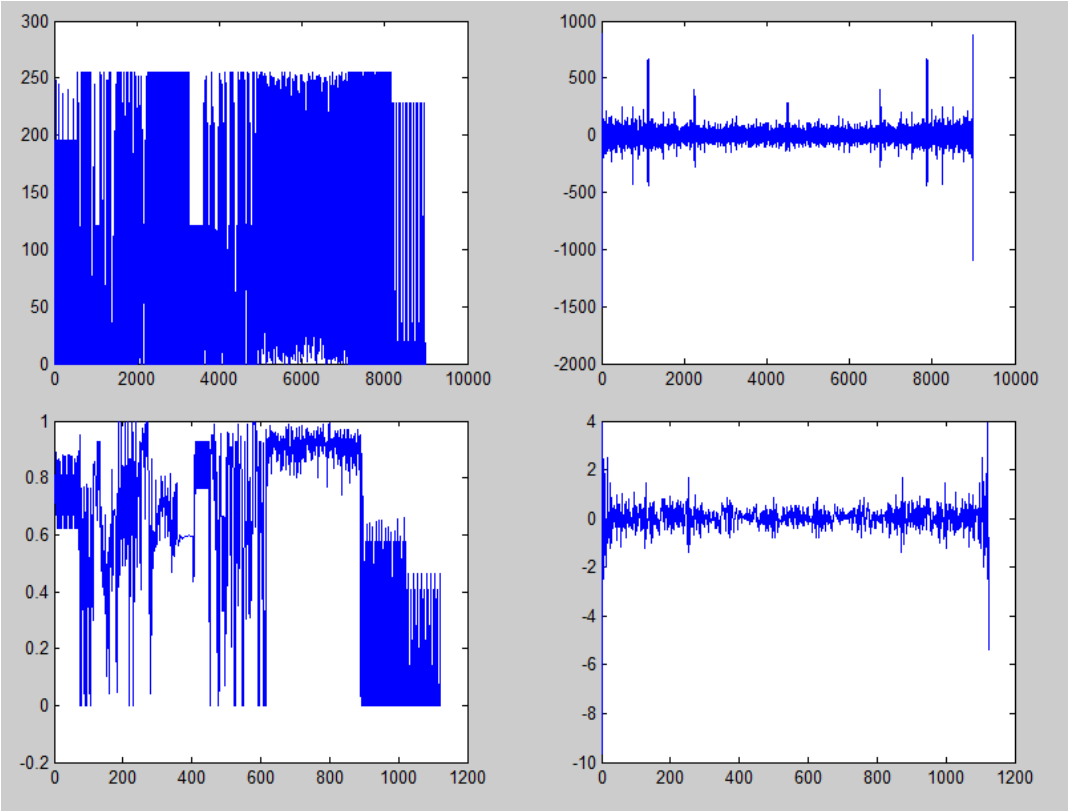
Na Obr. 29 je zobrazená grafická reprezentácia počiatočných dát, rýchla rouriérová transformácia vstupných dát, ďalej pole entropie a frekvenčnú analýzu tohto poľa pomocou rýchlej fourierovej transformácie. Na tomto obrázku ide o analýzu obrázku vo formáte JPEG a na Obr. 30 ide o analýzu škodlivého spustiteľného súboru. Grafické zobrazenia pozostávajú z reprezentácie obsahu binárneho súboru (ľavo hore), ďalej výpočtu fourierovej transformácie nad týmito dátami (pravo hore), nasleduje grafická reprezentácia výpočtu entropie analyzovaného súboru po častiach (ľavo dole)

a nakoniec rýchla fourierová transformácia nad hodnotami vypočítanej entropie (pravo dole).

Pri použití rýchlej Fourierovej transformácie na analýzu frekvenčného spektra vypočítaných hodnôt entropie opäť vznikajú 2 problémy. Prvým je veľká hustota vrcholov výsledného grafického zobrazenia a druhým problémom je, že nie je možné priradiť určitú časť výsledného frekvenčného spektra ku vybranej časti súboru. To môže výrazne ovplyvniť analýzu, keďže väčšina súborov pozostáva z minimálne dvoch primitívnych dátových typov (hlavička a samotné dáta). Preto je výhodné zvoliť na frekvenčnú analýzu vypočítaného poľa entropie kombináciu dvoch metód - krátkodobej Fourierovej transformácie a transformácie pomocou vlniek. [27]

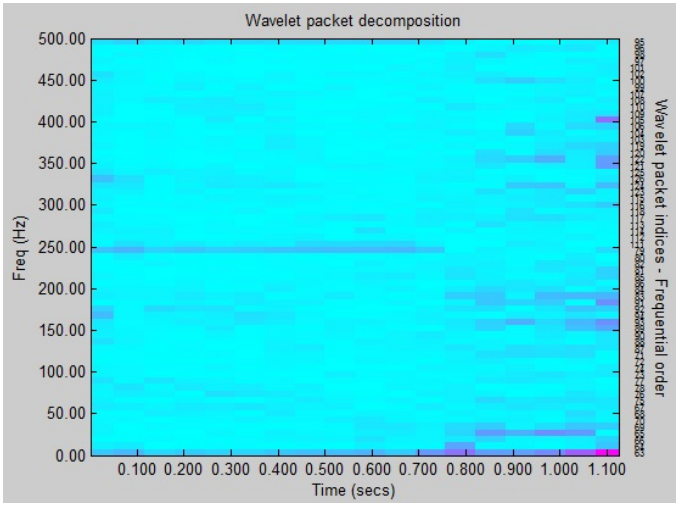


Obr. 29 Výpočet entropie a Fourierove transformácie JPG súboru

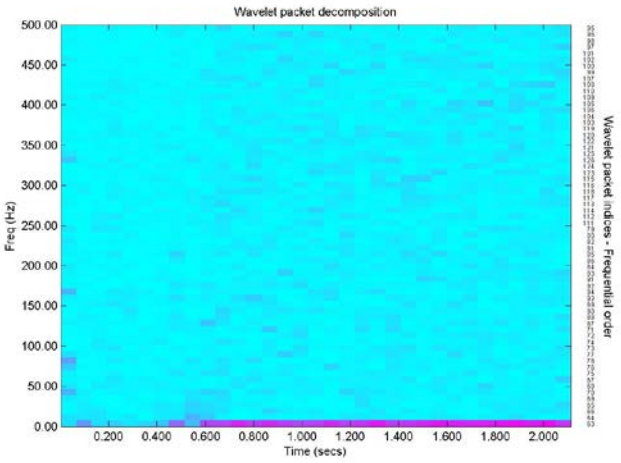


Obr. 30 Výpočet entropie a fourierová transformácia malware súboru

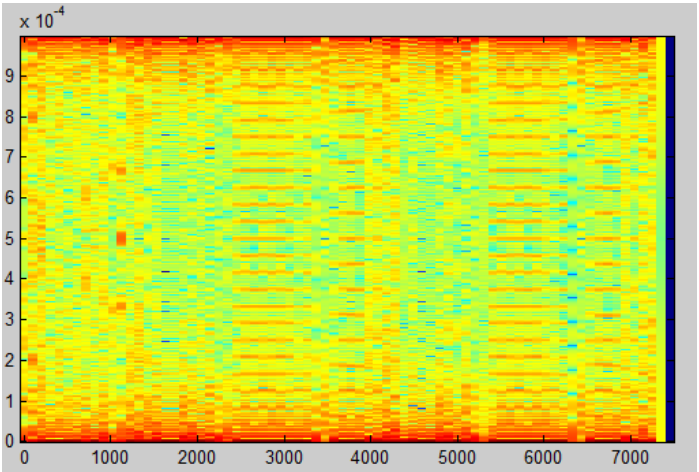
Frekvenčná analýza pomocou vlnkovej transformácie nad poľom hodnôt entropie je zobrazená na Obr. 31 a Obr. 32. Pri tejto analýze však platia iné pravidlá ako pri priamej analýze súboru binárnych dát. V poli entropie sú rozoznateľné 2 primitívne typy dát. Prvým typom sú kvázi náhodné hodnoty, prípadne komprimované dáta, pričom aj na výsledku vlnkovej transformácie je možné túto časť odlíšiť podľa neusporiadaného vzoru farebného spektra. Naproti tomu v intervaloch, kde sa často opakujú na rovnakej úrovni podobné hodnoty (tmavých častí grafu) sa nachádzajú dáta ako napríklad hlavička súboru, strojový kód, alebo programovací jazyk. Rovnakým spôsobom je možné pomocou spektrogramov na Obr. 33 a Obr. 34 rozoznať rôzne primitívne typy dát obsiahnuté v analyzovaných súboroch. Oproti scalogramu sa na spektrograme nevyskytujú rady a dáta sú viac spojité. Taktiež hodnota je reprezentovaná nielen zmenou kontrastu, ale aj zmenou samotnej farby časti grafu.



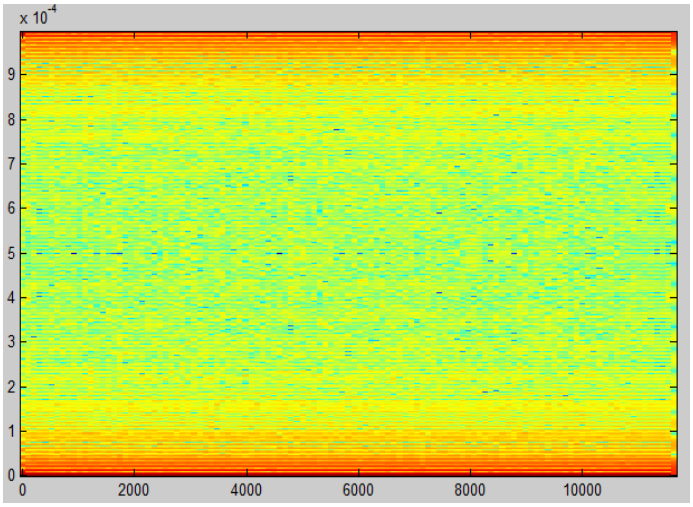
Obr. 31 Vlnková transformácia entropie spustiteľného súboru



Obr. 32 Vlnková transformácia entropie JPG súboru



Obr. 33 Krátkodobá fourierová transformácia entropie spustiteľného súboru



Obr. 34 Krátkodobá fourierová transformácia entropie JPG súboru

3.6 Detekcia škodlivého softvéru

Na detekciu škodlivého softvéru sa v súčasnosti používajú dva najrozšírenejšie spôsoby. Prvým spôsobom je vyhľadávanie signatúr (podpisov), pričom signatúra je v tomto prípade algoritmus alebo hash funkcia, ktorá jednoznačne identifikuje konkrétny škodlivý softvér. Tento spôsob detekcie škodlivého softvéru je síce najrýchlejší a najpoužívanejší, existuje však veľa spôsobov jeho obídenia. Jeho hlavným nedostatkom je fakt, že pokiaľ antivírový program neobsahuje vo svojej databáze danú signatúru, škodlivý softvér nie je možné detekovať. Takýto systém je ľahko zraniteľný voči útokom Zero-Day.

Druhým spôsobom, ktorý je menej používaný, je vyhľadávanie vzorov správania. Pri detekcii škodlivého softvéru sa sleduje aktivita aplikácie, napríklad ktoré aktivity robí daný spustiteľný súbor (napr. periodické odosielanie dát) [14]. Po detekovaní podozrivej aktivity následne antivírový program ohlási používateľovi podozrivý súbor a ten rozhodne, či bola daná aktivita regulárna alebo ide o škodlivý softvér. [29]

Najnovšie programy obsahujúce škodlivý kód už využívajú rôzne obranné mechanizmy na zabránenie ich odhalenia alebo na zamedzenie študovania ich zdrojového kódu prostredníctvom spätného inžinierstva. [19] Jednou z takýchto metód, ktorá patrí medzi v súčasnosti najpoužívanejšie, je metóda kompresie časti kódu, pričom po použití tejto metódy už vo väčšine prípadov nie je možné pomocou pôvodnej signatúry označiť súbor ako malware. [21]

3.6.1 Pokročilé metódy detekcie škodlivého softvéru

Jednou z úloh tejto práce je detekcia jedného z množstva rôznych typov škodlivého softvéru a to konkrétne spustiteľnými binárnymi súbormi operačného systému Windows, ktoré zároveň tvoria aj najpočetnejšiu skupinu s celosvetovo najväčším výskytom. Skupina spustiteľných binárnych súborov pokrýva vo všeobecnosti adware, trójske kone, počítačové červy, počítačové vírusy, spyware a ďalšie typy škodlivého softvéru. [23]

Keďže spomínaná skupina tvorí najpočetnejšiu bezpečnostnú hrozbu, je v nej zaznamenaný aj najvyšší počet výskytov Zero-Day útokov. Práve pomocou v tejto práci navrhnutých metód grafickej analýzy dát je možné niektoré z týchto útokov odchytiť, prípadne upozorniť na možnosť ohrozenia a označiť skúmaný binárny súbor za potenciálnu hrozbu.

Samotná detekcia škodlivého softvéru je zameraná na analýzu vlastností skúmaných vzoriek ako škodlivého softvéru, tak aj bežne vyskytujúcich sa súborov, ktoré neobsahujú škodlivý kód. Detekcia škodlivého softvéru je teda postavená na princípe porovnávania podobnosti skúmaných vlastností pomocou navrhnutých grafických a matematických metód.

Pomocou navrhnutých metód je analýzou možné pre každý skúmaný objekt vypočítať desiatky hodnôt, ktoré opisujú jeho vlastnosti, ako napríklad výpočet celkovej entropie súboru, výpočet čiastkovej entropie, použitie transformácií na frekvenčnú analýzu buď samotného obsahu alebo čiastkových výpočtov entropie, výpočet entropie nad výsledkami frekvenčnej analýzy atď. Cieľom tejto práce je návrh minimálnej a zároveň dostatočne presnej kombinácie výsledných premenných analýzy, ktorá by slúžila na detekciu vybraného typu škodlivého softvéru. Táto forma bola navrhnutá postupne pomocou logickej a experimentálnej analýzy reprezentatívnych vzoriek dát.

3.6.2 Algoritmus detekcie škodlivého softvéru

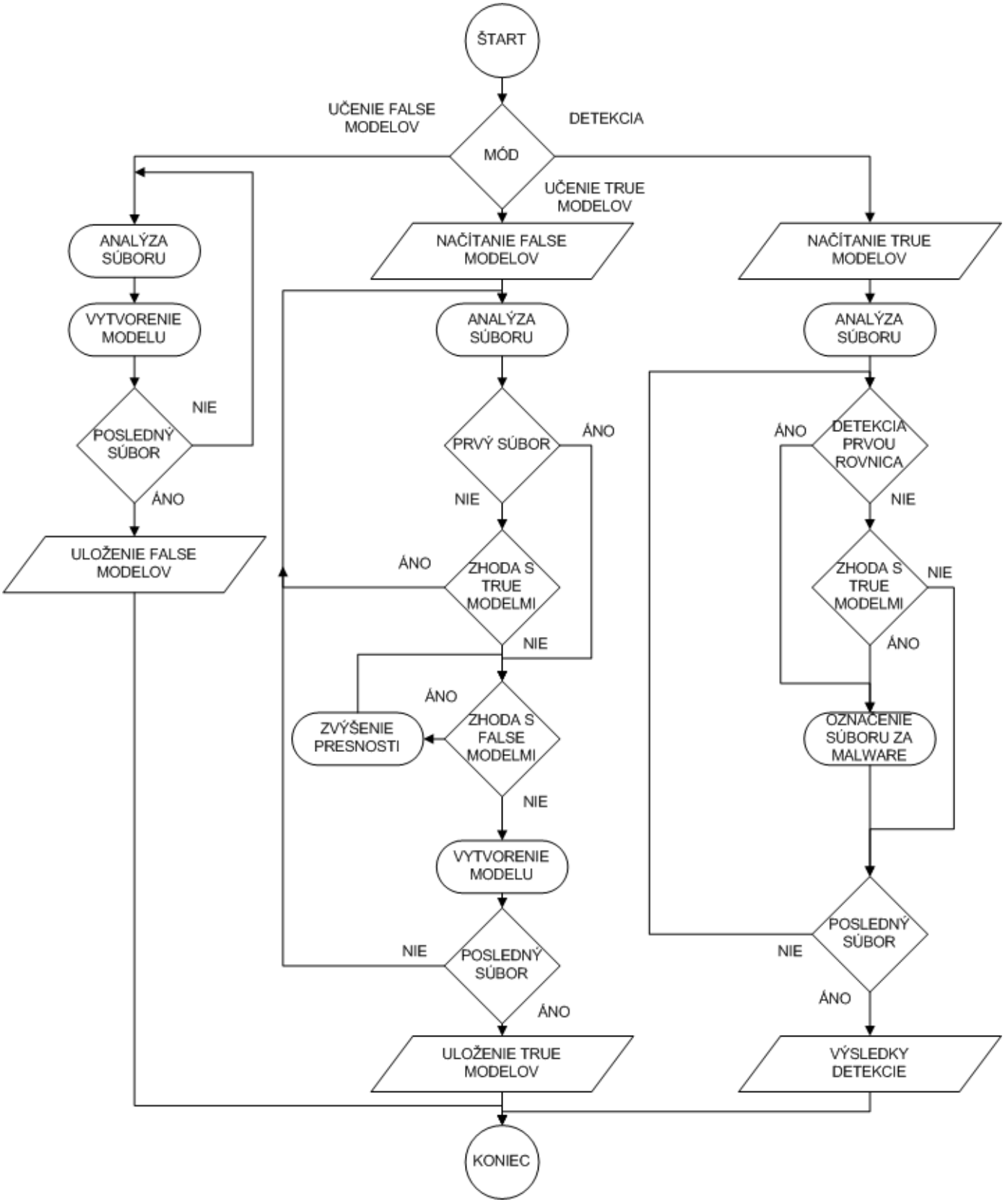
Keďže cieľom tejto práce je navrhnúť systém detekcie škodlivého softvéru bez hlbšej analýzy jeho obsahu (kódu/textu/dát) pomocou ich vonkajších vlastností, nie je možné teoreticky bez experimentálnej zložky analýzy vyvodiť potrebné závery. Z tohto dôvodu sa celá práca zameriavala výlučne na analýzu reprezentatívnych vzoriek vybraných typov súborov. Každá reprezentatívna vzorka je tvorená vybranými 300 až

1 000 súbormi, pričom vzorka reprezentujúca škodlivý softvér je tvorená 6 000 súbormi daného typu. Celá skúmaná vzorka je tvorená viac ako 12 000 súbormi rôznych typov a veľkostí. Pomocou popísaných metód analýzy bolo teda možné nad touto vzorkou zostaviť algoritmus, ktorý dokáže s určitou presnosťou nájsť v danej vzorke škodlivý softvér.

Samotný algoritmus detekcie pozostáva z dvoch základných častí. Prvou časťou algoritmu je využitie grafických a štatistických metód na nájdenie intervalov vybraných hodnôt analýzy. Tieto intervaly musia jednoznačne určiť či daný súbor obsahuje škodlivý kód, alebo či je daný súbor iného typu. Pomocou výsledných intervalov je možné zostaviť výslednú minimálnu formu rovnice pre tieto premenné. Pozitívnym výsledkom tejto rovnice budú len súbory, ktoré boli označené ako malware, ďalej budeme tieto výsledky označovať ako TRUE POSITIVE. Ak by existoval súbor, ktorý nespadá do kategórie malware, a výsledná rovnica by ho napriek tomu označila za malware, tento výsledok budeme označovať ako FALSE POSITIVE. Rovnica v prvej časti algoritmu je postavená na princípe, že nie je potrebné označiť čo najväčšie množstvo súborov ako TRUE POSITIVE, ale jej základnou podmienkou je aby v žiadnom alebo minimálnom počte prípadov označila FALSE POSITIVE výsledok.

Druhou časťou algoritmu je vytvorenie modelov pomocou vybraných premenných, ktoré opisujú súbory typu malware a nespadajú do výslednej rovnice prvého bodu algoritmu. Vytvorenie modelov je taktiež podmienené minimálnym prípadne žiadnym výskytom FALSE POSITIVE výsledku detekcie.

Takýto princíp vytváranie modelov by sa dal popísať termínom učenie, keďže dané modely vznikajú postupným prechádzaním súborov typu malware a ich porovnávaním s už vytvorenými modelmi, prípadne porovnávaním s reprezentačnou vzorkou, na ktorej by mohol daný model vrátiť FALSE POSITIVE výsledok. Každý model okrem vybraných výsledných premenných analýzy obsahuje aj koeficient presnosti, ktorý slúži ako koeficient rovnice pre porovnávanie modelov a analyzovaných súborov.



Obr. 35 Vývojový diagram algoritmu detekcie škodlivého softvéru

Vývojový diagram popisujúci funkciu algoritmu je zobrazený na Obr. 35. Z diagramu je zrejmé, že algoritmus pracuje v troch po sebe nasledujúcich režimoch. Prvým režimom je vytvorenie FALSE modelov, ktoré následne pomáhajú pri vytváraní TRUE modelov. Nasledujúcim režimom je režim učenia sa TRUE modelov, kde sa porovnávajú analyzované súbory s už naučenými modelmi a FALSE modelmi a posledným je mód detekcie, v ktorom už prebieha samotná detekcia škodlivého softvéru nad neznámymi dátami. TRUE model pomenováva model, ktorého môže pokryť jeden alebo viac TRUE POSITIVE výsledkov, s 0% FALSE POSITIVE výsledkom. FALSE modely slúžia výlučne na testovanie hodnoty koeficientu presnosti TRUE MODELOV, avšak v prípade jeho aplikácie môže pokryť 1 až n súborov s FALSE POSITIVE výsledkom.

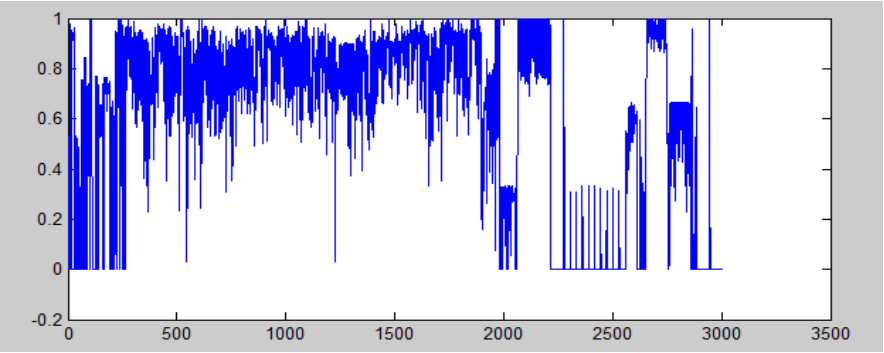
3.6.3 Zostavenie výsledných rovníc a ich parametrov

Pre zostavenie výsledných rovníc, ktoré by slúžili na výslednú detekciu určitého vybraného typu súborov, je v prvom rade potrebné vybrať z popísaných metód grafickej analýzy tie najvhodnejšie a ktoré zároveň najvýraznejšie vyjadrujú závislosť danej veličiny od typu a obsahu súboru. Pri skúmaní možností, či už experimentálne alebo využitím poznatkov publikovaných výsledkov podobných výskumov, sme za najvhodnejší spôsob analýzy zvolili využitie výpočtu entropie nad binárnym súborom a následnú frekvenčnú analýzu tohto medzivýsledku. Aj napriek výberu vhodnej metódy analýzy z desiatok rôznych kombinácií grafických a matematických metód ostáva problém extrahovania informácií z výsledku analýzy, ktorý obsahuje stále širokú informačnú doménu. Ako príklad môžeme uviesť analýzu súboru s veľkosťou jedného megabajtu, po ktorého analýze entropie a následnej frekvenčnej analýze dostávame scalogram obsahujúci viac ako 125 tisíc bodov a zároveň taktiež spektrogram s viac ako 70 tisíc bodmi.

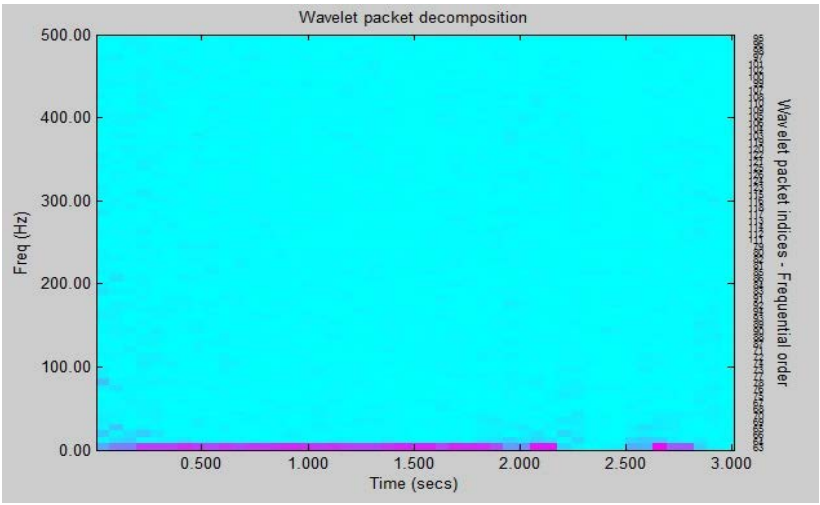
Prvým bodom analýzy je teda výpočet entropie skúmaného súboru. Výpočet entropie prebieha po častiach použitím funkcie okna s veľkosťou 8 znakov, ktorá bola experimentálne zvolená ako vhodná možnosť. Použitím funkcie okna a jeho postupným posúvaním naprieč celou dĺžkou hexadecimálnej reprezentácie skúmaného súboru dostaneme funkciu, ktorej obor hodnôt sa pohybuje v intervale od 0 po 1. Hodnotou 1 je reprezentovaná maximálna neusporiadanosť prvkov analyzovaného intervalu a hodnotou 0 je reprezentované ich maximálne usporiadanie. Na Obr. 36 je znázornený graf entropie nad danými časťami vstupného súboru. Keďže pri hodnote ôsmich znakov

ktoré sú naraz analyzované vzniká minimálna možnosť straty informačnej hodnoty kvôli naviazaniu na susedné intervaly, nie je potrebné používať presah. Naopak, v tomto prípade poskytuje väčšia odchýlka hodnôt susedných intervalov vyššiu informačnú hodnotu z pohľadu následnej frekvenčnej analýzy.

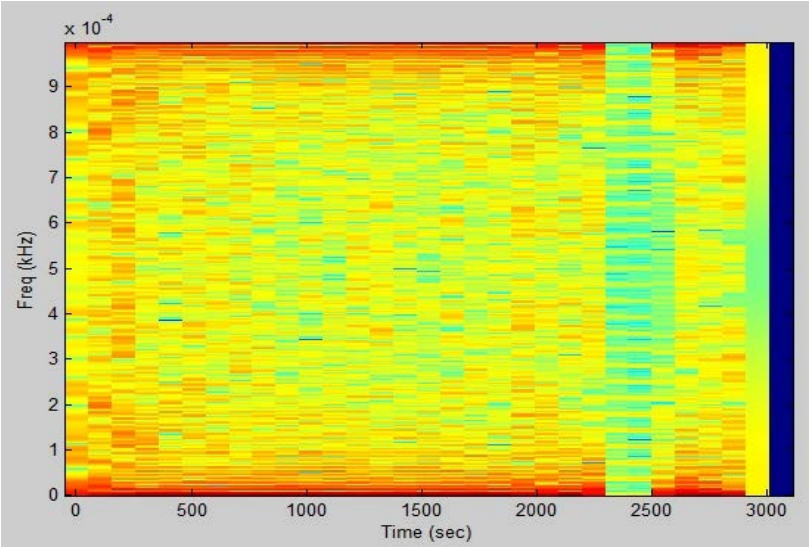
Ďalším bodom analýzy je použitie krátkodobej fourierovej transformácie a vlnkovej transformácie nad výsledným poľom výpočtu entropie. Použitie oboch druhov transformácie súčasne by sa mohlo zdať redundantné, čo môžeme vidieť na prvý pohľad aj pri porovnaní samotných grafických zobrazení na Obr. 37 a Obr. 38, avšak pri hlbšej analýze zisťujeme že hodnoty polí sa značne odlišujú a každá z transformácií v podstate popisuje inú vlastnosť dát. Táto skutočnosť vyplýva už zo základných princípov transformácií. Princípom Fourierovej transformácie je vyhľadávanie goniometrických funkcií naprieč celou dĺžkou skúmaného intervalu, naproti tomu pri vlnkovej transformácií sa zameriavame na vyhľadávanie krátkych spojitých oscilácií (vlniek), ktoré nemusia pokrývať celú dĺžku daného intervalu ale môžu sa nachádzať len v niektorej časti.



Obr. 36 Vývojový diagram algoritmu detekcie škodlivého softvéru



Obr. 37 Scalogram vlnkovej transformácie



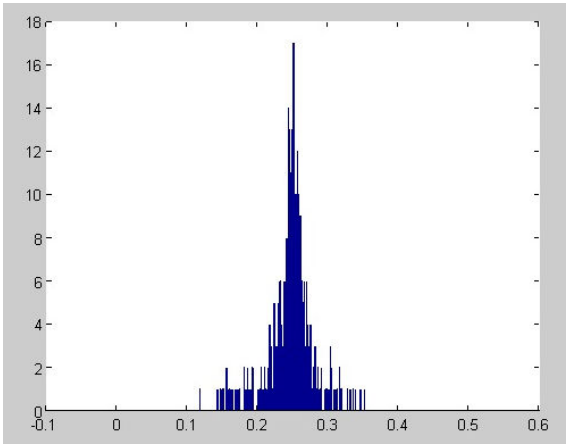
Obr. 38 Spektrogram fourierovej transformácie

Výsledkom tejto časti analýzy sú teda 3 grafické zobrazenia a to scalogram, spektrogram a zobrazenie samotného poľa vypočítaných čiastkových entropií. Z týchto troch grafických zobrazení (polí hodnôt) je ďalej potrebné vypočítať hodnoty premenných, ktoré určujú určitú vlastnosť zobrazenia jedinou hodnotou. Kombináciou rôznych matematických operácií je možné určiť desiatky rôznych hodnôt ako napríklad vypočítanie sumy hodnôt amplitúdy každého stĺpca naprieč celým intervalom grafu, pričom výsledná hodnota môže byť opäť suma týchto stĺpcov, ich maximálna alebo minimálna hodnota, prípadne ich priemerná hodnota. Pri týchto výpočtoch môžeme ďalej využívať aj frekvenčné filtre, ktoré boli popísané v predchádzajúcich kapitolách, čo aj vo výslednom algoritme využívame pri analýze vlnkovej transformácie.

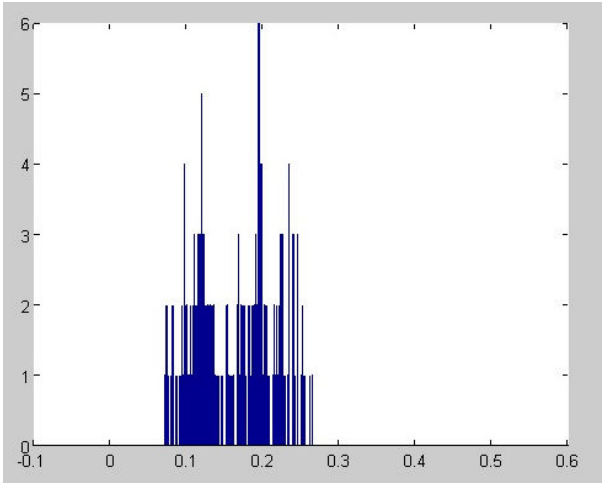
Ďalej je potrebné z takto vypočítaného množstva hodnôt logicky prípadne experimentálne vytvoriť najmenšiu množinu premenných, ktorá bude zároveň čo najpresnejšie popisovať vlastnosti skúmaného súboru, počet jej premenných však bude čo najnižší kvôli odstráneniu redundancie, pričom viacero premenných môže popisovať tú istú vlastnosť a je preto možné použiť iba najpresnejšiu z nich.

Na zostavenie výslednej rovnice, ktorá by slúžila na samotnú detekciu je potrebné uskutočniť túto analýzu nad vybranými vzorkami dát a určiť tak významnosť, redundanciu a rozsah hodnôt vybraných premenných. Na zobrazenie rozsahu výsledných hodnôt daných premenných použijeme štatistické zobrazenie histogram. Príklady histogramov vypočítaných hodnôt sú zobrazené na Obr. 39 -Obr. 42. Na Obr. 39 môžeme vidieť histogram hodnoty ktorá predstavuje maximálnu hodnotu

priemerných hodnôt každého stĺpca scalogramu. Tento histogram bol zostavený z analýzy spustiteľných súborov typu EXE.



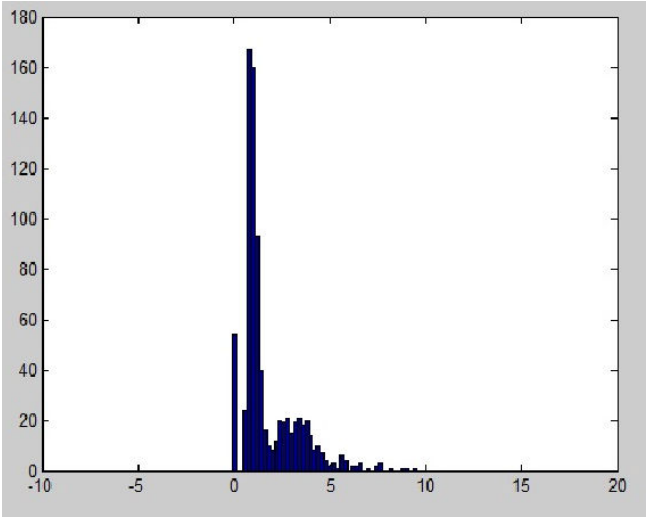
Obr. 39 Histogram hodnôt scalogramu EXE súboru



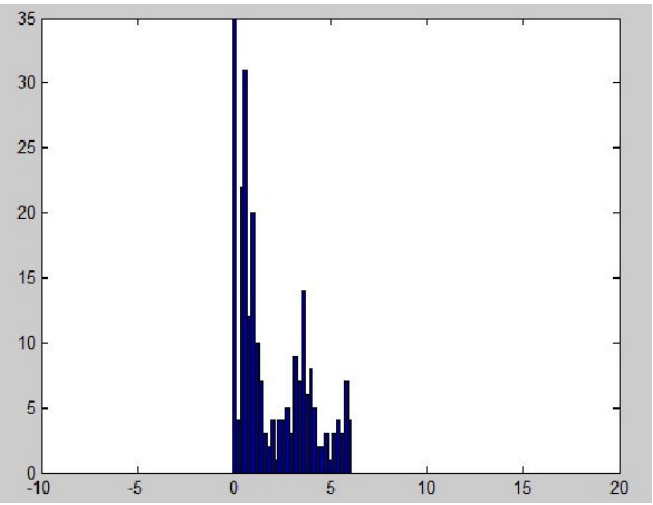
Obr. 40 Histogram hodnôt scalogramu JPG súboru

Obr. 40 môžeme vidieť histogram tej istej hodnoty, avšak tento krát ide o analýzu súborov obsahujúcich komprimovaný obrázok typu JPG. Ako môžeme vidieť na prvý pohľad, pri porovnávaní týchto grafických znázornení je možné rozlíšiť viacero intervalov. Môžeme určiť interval, na ktorom sa vyskytujú iba hodnoty JPG súborov, alebo iba hodnoty EXE súborov, prípadne interval, kde sa vyskytujú hodnoty oboch typov. Taktiež môžeme určiť maximálny možný interval hodnôt pre daný typ. Keďže je jasne viditeľné prekrytie daných intervalov hodnôt, nie je možné presne určiť typ súboru len pomocou jednej jedinej hodnoty. Preto je nutné použiť kombináciu viacerých premenných na presné a jednoznačné určenie príslušnosti analyzovaného súboru. Na histogramoch Obr. 41 a Obr. 42 napríklad nie je možné určiť podľa danej premennej príslušnosť súboru, pretože intervaly sa takmer úplne prekrývajú a výpovedná hodnota

tejto premennej je minimálna. Na nasledujúcich obrázkoch je reprezentovaný interval maximálnych hodnôt vypočítaných z priemerných hodnôt stĺpcov spektrogramu.



Obr. 41 Histogram hodnôt scalogramu EXE súboru

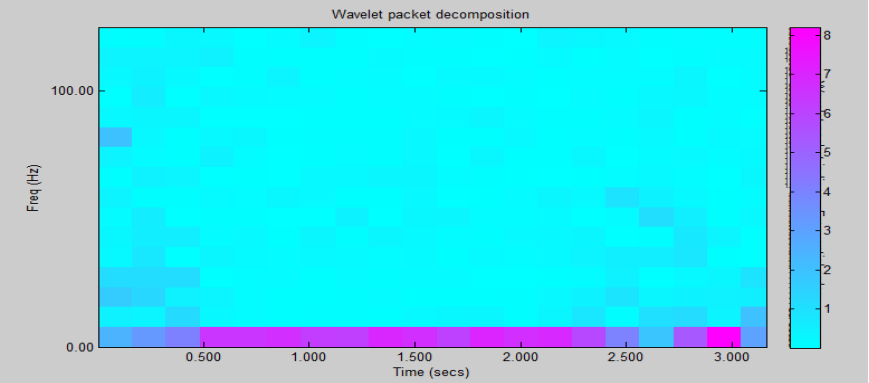


Obr. 42 Histogram hodnôt scalogramu JPG súboru

Grafickou analýzou intervalov hodnôt rôznych súborových typov boli nakoniec vybrané 4 premenné, pomocou ktorých je možné postaviť minimálnu formu rovnice detekcie škodlivého softvéru. Prvé dve premenné sú vypočítané z hodnôt scalogramu a ďalšie dve sú vypočítané z hodnôt spektrogramu.

Pri analýze výsledku vlnkovej transformácie však musíme upozorniť na jednu z vlastností tohto typu analýzy. Scalogram ako už bol v predchádzajúcich kapitolách popísaný reprezentuje na ypsilonovej osi grafu stupeň rozkladu vlnkovej transformácie, čo v konečnom dôsledku znamená, že najnižší stupeň reprezentuje frekvenčnú závislosť najnižších frekvencií a každý vyšší stupeň reprezentuje čoraz vyšších frekvencií. Experimentálne bolo v tejto práci zistené, že rozdelením scalogramu na 2 nezávislé časti

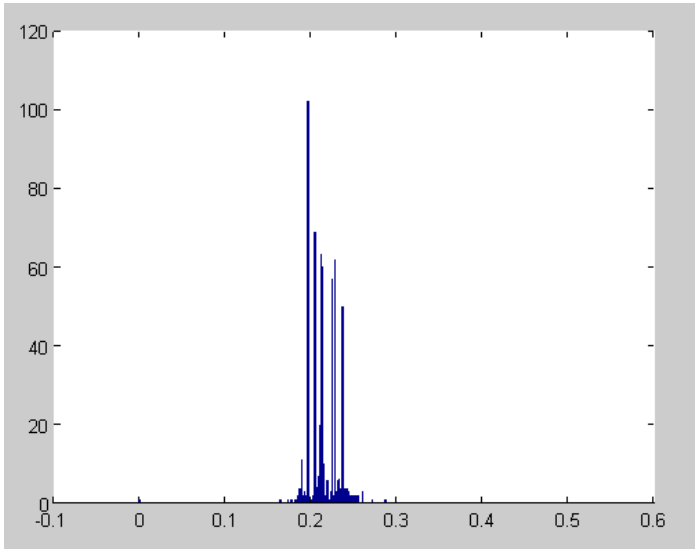
dostaneme pomocou ďalšej analýzy tie najpresnejšie výsledky. Prvou časťou je prvý stupeň transformácie, ktorý predstavuje pole hodnôt s dĺžkou časovej osi scalogramu a výškou 1. Druhá časťou scalogramu sa skladá zo všetkých nasledujúcich vyšších stupňov transformácie. Detailné zobrazenie prvého stupňa scalogramu, ktorý zároveň obsahuje aj najvyššie hodnoty môžeme vidieť na Obr. 43.



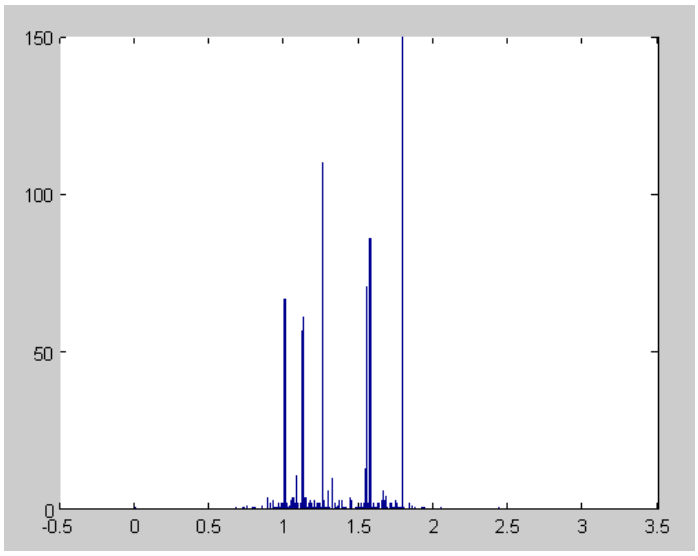
Obr. 43 Prvý stupeň scalogramu

Po rozdelení scalogramu na 2 časti je možné určiť prvé dve premenné výslednej rovnice. Prvou premennou je maximálna hodnota priemerných hodnôt vypočítaných po stĺpcoch pozdĺž hornej časti scalogramu (stupne 3 ... X), ďalej označovaná ako ENTRO_WAVE_HEIGHT_AVG_MAX. Ďalšou premennou je maximálna hodnota priemerných hodnôt vypočítaná z dvoch najnižšieho stupňov scalogramu (stupeň 1 a 2), ďalej označovaná ako ENTRO_WAVE_LOW_AVG_MAX. Tretia a štvrtá premenná sú určené pomocou krátkodobej fourierovej transformácie. Tretia premenná je priemerná hodnota vypočítaná z priemerných hodnôt stĺpcov spektrogramu, ďalej označovaná ako ENTRO_STFT_AVG_AVG. Posledná štvrtá hodnota je určená ako priemerná hodnota výsledkov aplikácie výpočtu entropie na stĺpce spektrogramu, ďalej označovaná ako ENTRO_STFT_ENTRO_AVG.

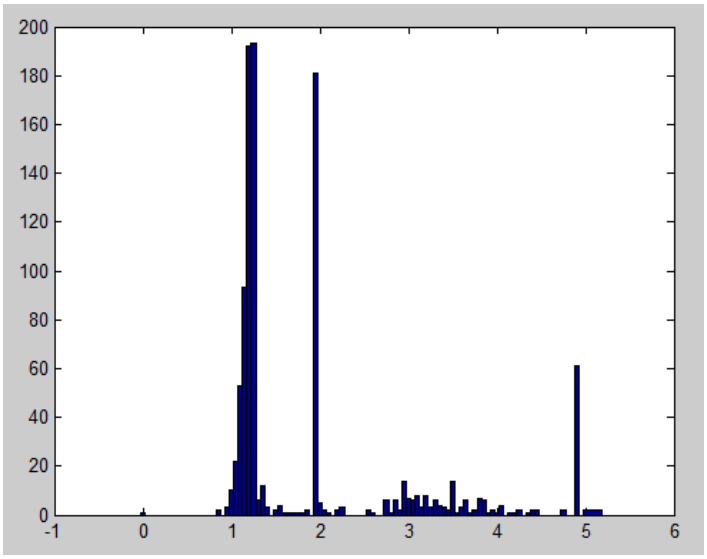
Kombináciou týchto štyroch premenných je možné zostaviť výsledné rovnice pre detekciu škodlivého softvéru. Rovnice boli zostavené pomocou grafickej analýzy spektrogramov rôznych dátových typov a zároveň testovaním daných rovníc na rôznych typoch dát. Spektrogramy daných premenných pre vybranú testovaciu vzorku súborov obsahujúcich škodlivý kód sú zobrazené na obrázkoch Obr. 44, Obr. 45, Obr. 46, Obr. 47.



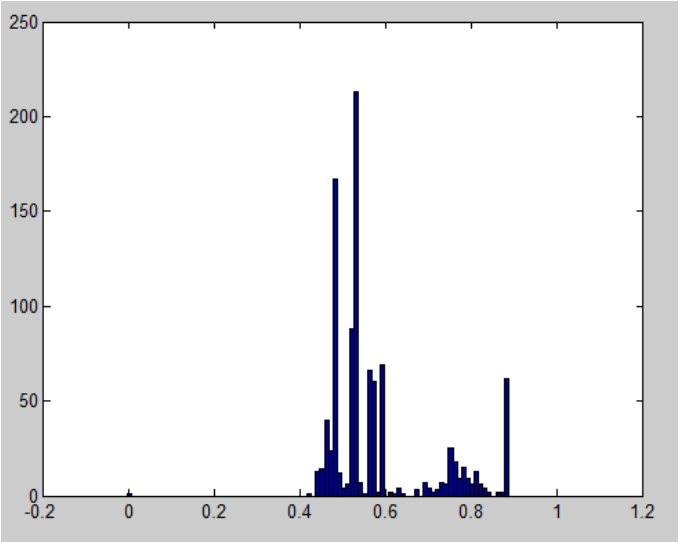
Obr. 44 Histogram hodnôt ENTRO_WAVE_HEIGHT_AVG_MAX



Obr. 45 Histogram hodnôt ENTRO_WAVE_LOW_AVG_MAX



Obr. 46 Histogram hodnôt ENTRO_STFT_AVG_AVG



Obr. 47 Histogram hotnôt ENTRO_STFT_ENTRO_AVG

Výsledné histogramy daných hodnôt nám určujú nielen samotný rozptyl, ale porovnaním s histogramami iných typov súborov bolo možné určiť intervaly na ktorých môžeme s určitosťou povedať, že daný súbor je typu malware. Pomocou prvej rovnice teda bude možné určiť len podmnožinu malware súborov, avšak s 0% FALSE POSITIVE výsledkov a pomocou druhej rovnice bude možné pokryť 100% malware avšak počet FALSE POSITIVE výsledkov bude vyšší a rovnica bude zároveň pokrývať aj podmnožiny iných typov súborov. Množinu výsledkov druhej rovnice budeme ďalej označovať termínom POSSIBLE MALWARE. Z praktického hľadiska metóda, ktorá detekuje 60% TRUE POSITIVE, s 0% FALSE POSITIVE výsledkom je oveľa užitočnejšia ako metóda, ktorá detekuje 80% TRUE POSITIVE výsledkov s 20% FALSE POSITIVE výsledkami. V prvom prípade je úplne isté že dosiahnutý kladný výsledok je správny, pričom na detekciu zostávajúcich 40% neznámeho výsledku je stále možné použiť alternatívne metódy detekcie. Napriek tomu v druhom prípade nie je možné žiaden výsledok označiť za úplne pravdivý, keďže použitie danej metódy so sebou prináša určitý hazard. [15]

Výsledná prvá rovnica je teda nasledujúca:

$$COND1 = ENTRO\_WAVE\_HEIGHT\_AVG\_MAX > 0.16 \;\&\; ENTRO\_WAVE\_HEIGHT\_AVG\_MAX < 0.22$$

$$COND2 = ENTRO\_WAVE\_LOW\_AVG\_MAX > 0.95 \;\&\; ENTRO\_WAVE\_LOW\_AVG\_MAX < 0.2$$

COND3 = (ENTRO_STFT_AVG_AVG> 1 & ENTRO_STFT_AVG_AVG < 1.4) |
(ENTRO_STFT_AVG_AVG> 1.85 & ENTRO_STFT_AVG_AVG < 2.1)

COND4 = (ENTRO_STFT_ENTRO_AVG> 0.42 & ENTRO_STFT_ENTRO_AVG < 0.58) |
(ENTRO_STFT_ENTRO_AVG> 0.75 & ENTRO_STFT_AVG_AVG < 0.83)

IF (COND1 & COND2 & COND3 & COND4) = MALWARE

Pomocou tejto rovnice sme na vybranej vzorke tvorenej 2000 súbormi rôznych typov schopný detekovať 57% TRUE POSITIVE výsledkov s 0% FALSE POSITIVE výsledkom. Výsledná druhá rovnica je nasledujúca:

COND1 = ENTRO_WAVE_HEIGHT_AVG_MAX > 0.16 &
ENTRO_WAVE_HEIGHT_AVG_MAX < 0.29

COND2 = ENTRO_WAVE_LOW_AVG_MAX > 0.8 &
ENTRO_WAVE_LOW_AVG_MAX < 2

COND3 = ENTRO_STFT_AVG_AVG> 0.8 & ENTRO_STFT_AVG_AVG < 4.8

COND4 =ENTRO_STFT_ENTRO_AVG> 0.42 & ENTRO_STFT_ENTRO_AVG < 0.9

IF (COND1 & COND2 & COND3 & COND4) = POSSIBLE MALWARE

Pomocou tejto rovnice sme na vybranej vzorke tvorenej 2000 súbormi rôznych typov schopný detekovať 97% TRUE POSITIVE výsledkov s 35% FALSE POSITIVE výsledkom. Keďže za požadovaný výsledok je možné prijať iba metódu, ktorá má 0% FALSE POSITIVE výsledkov, je potrebné aplikovať ďalšie alternatívne metódy detekcie škodlivého softvéru.

3.6.4 Modelovanie

Keďže podľa predpokladu nie je možné vytvoriť rovnicu, ktorá by zahŕňala všetky skúmané vzorky malware súborov a zároveň bola dostatočne presná na to, aby neoznačila súbory iného typu nepravdivo za malware, je nutné vytvoriť viacero rovníc s vyššou presnosťou a užšou podmnožinou detekovaných súborov tak, aby bolo možné pokryť čo najväčšiu časť vzorky s 0% FALSE POSITIVE výsledkom. Tieto rovnice budeme nazývať modelmi, pretože podľa definície modelu aj tieto rovnice popisujú zoskupenie objektov, ktoré sú vzájomne prepojené podľa určitých podmienok.

Na vytvorenie modelov (rovníc) použijeme rovnaký postup analýzy a rovnaké premenné ako v predchádzajúcej podkapitole. Využitie tohto postupu analýzy a premenných na vytvorenie modelov je možné vďaka tomu, že predchádzajúce výsledky analýzy nám potvrdili výraznú závislosť hodnôt premenných na type a obsahu súborov z hľadiska detekcie škodlivých súborov.

Ako model nám teda bude slúžiť usporiadaná pätica hodnôt, pričom prvé štyri hodnoty sú hodnoty premenných vypočítané analýzou súboru a piatou hodnotou je koeficient presnosti. Tento koeficient dosádzame do rovnice slúžiacej na určenie či daný súbor spadá do definovaného modelu. Výsledná rovnica má nasledovný tvar:

$$VAR1 = ENTRO\_WAVE\_HEIGHT\_AVG\_MAX$$

$$VAR2 = ENTRO\_WAVE\_LOW\_AVG\_MAX$$

$$VAR3 = ENTRO\_STFT\_AVG\_AVG$$

$$VAR4 = ENTRO\_STFT\_ENTRO\_AVG$$

$$IF\ (ABS(VAR1 - MOD1) + ABS(VAR2 - MOD2) + ABS(VAR3 - MOD3) + ABS(VAR4 - MOD4) < COEF\ ) = MODEL\ MATCH$$

kde ABS predstavuje funkciu absolútnej hodnoty, COEF je koeficient presnosti daného modelu a MOD1 až MOD4 sú hodnoty premenných modelu.

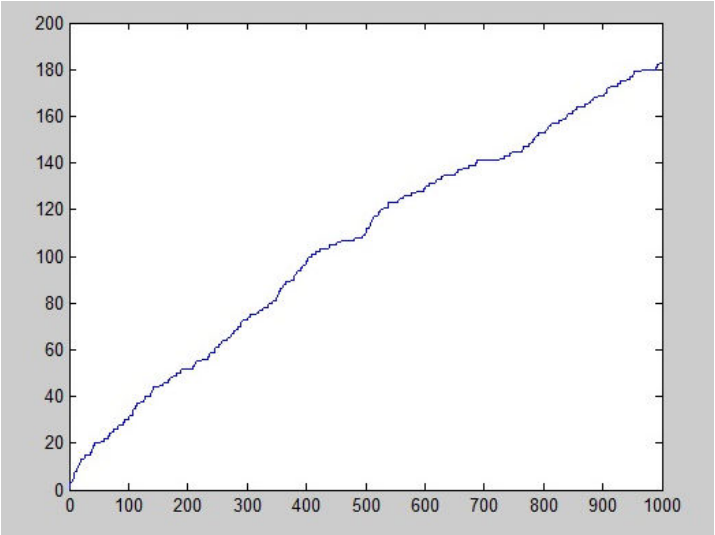
Algoritmus uvedený na Obr. 35 popisuje postup vytvárania modelov súborov typu malware (ďalej označované TRUE modely), postup vytvárania modelov, ktoré majú zabrániť výskytu FALSE POSITIVE výsledkov (ďalej označované FALSE modely) a nakoniec postup samotnej aplikácie modelov pri detekcií škodlivého softvéru. V prvom bode algoritmu sa vykonáva vetva označená „učenie false modelov“, ktorej úlohou je vytvoriť modely zo súborov rôznych typov okrem malware súborov. Tieto modely sú tvorené iba štyrmi premennými, pričom koeficient presnosti nie je potrebné pri týchto modeloch uvádzať. Na vytvorenie FALSE modelov bola v našom prípade použitá vzorka 1600 súborov pozostávajúca z viac ako desiatich typov súborov. Po vytvorení týchto modelov prechádza algoritmus do vetvy označenej „učenie true modelov“, kde sa dostávame ku vytvoreniu modelov slúžiacich na detekciu malware (TRUE modelov).

Na vytvorenie TRUE modelov bola použitá vzorka 2 000 súborov obsahujúcich rôzne druhy škodlivých súborov ako napríklad trójske kone, počítačové červy, adware a podobne. Počas vytvárania modelov je potrebné postupne prejsť všetky súbory danej vzorky, pričom výsledky analýzy každého súboru sú porovnané s už vytvorenými modelmi. Ak niektorý z týchto modelov popisujú práve analyzovaný súbor je tento súbor vynechaný a algoritmus pokračuje nasledovným súborom. Ak však nie je možné analyzovaný súbor popísať niektorým z už naučených modelov, je vytvorený nový model s predvolenou hodnotou presnosti (v našom prípade 0.05). Následne je ešte potrebné pomocou FALSE modelov preveriť či daný model nepopisuje aj súbory iného typu a môže dôjsť ku FALSE POSITIVE výsledku. Ak model popisuje aj jeden alebo viacero naučených FALSE modelov, je zvýšená presnosť modelu a preverovanie FALSE MODELOV sa opakuje, až pokiaľ nie je koeficient presnosti dostatočne nízky na to, aby model nemohol popísať žiaden zo súborov iného typu. Ideálny počet vytvorených modelov na popísanie všetkých malware súborov danej vzorky by mal byť čo najnižší, avšak ich počet je závislý ako od hodnoty koeficientu presnosti, tak aj od rôznorodosti typov malware súborov danej vzorky. Grafické zobrazenie počtu modelov v závislosti na počte analyzovaných súborov je zobrazené na Obr. 48.

3.6.5 Výsledky analýzy binárnych súborov

Výsledkom analýzy vybraných reprezentatívnych vzoriek rôznych typov súborov je algoritmus, ktorým je možné detekovať 96,4% TRUE POSITIVE výsledkov s 0% FALSE POSITIVE výsledkom. Z 2000 súborov, ktoré boli použité ako vzorka súborov typu malware, bolo správne detekovaných pomocou prvej rovnice 58% a pomocou vytvorených modelov bolo detekovaných ďalších 38,4% súborov. Pri maximálnom koeficiente presnosti 0.05 bolo vytvorených 294 modelov, ktoré spolu popísali 768 súborov. Ďalšie podrobné výsledky analýzy sú uvedené v Tab. 1. V tabuľke sú prezentované výsledky detekcie naučených súborov označených (Malware LEARN), pojem naučených používame preto, že práve pomocou tejto vzorky dát boli zostavené výsledné rovnice a modely, ako aj výsledky detekcie nenaučených súborov označených (Malware 1 a 2). Výsledky analýzy binárnych súborov v tejto tabuľke ako na naučenej tak aj na nenaučenej vzorke dát jasne dokazujú možnosť reálneho využitia týchto metód detekcie v praxi. Z daných výsledkov je možné vydedukovať, že existuje určitá majoritná skupina súborov škodlivého softvéru, ktorá má rovnaké alebo podobné

vlastnosti z pohľadu grafickej analýzy dát. Táto skupina súborov je v našom prípade detekovaná hlavne pomocou prvej rovnice a čiastočne pomocou modelov a táto skutočnosť je potvrdená aj výsledkami analýzy neznámych vzoriek škodlivého softvéru. Tento záver teda potvrdzuje jednu zo základných myšlienok práce a prezentuje možnosť využitia grafickej analýzy dát pri odhaľovaní nových neodhalených hrozieb (Zero-Day).



Obr. 48 Graf počtu vytvorených modelov

Tab. 1 Výsledky detekcie na rôznych typoch súborov

Typ súboru	Počet súborov	Prvá rovnica	MODEL MATCH	TRUE POSITIVE	FALSE POSITIVE	POSSIBLE MALWARE
Malware LEARN	2000	1160	768	96,4%	0%	768
Malware 1	2000	1073	531	80,2%	0%	787
Malware 2	2000	1086	549	81,75%	0%	792
EXE	691	0	0	0%	0%	591
JAR	273	0	0	0%	0%	28
DLL	414	0	0	0%	0%	321
JPG	674	0	0	0%	0%	178
PDF	812	0	0	0%	0%	10
DOC	356	0	0	0%	0%	320
MP3	100	0	0	0%	0%	6
XLS	244	0	0	0%	0%	244

4 Grafická analýza web komunikácie

Táto časť práce pojednáva o využití popísaných metód pri analýze komplexných dát, ktoré reprezentujú údaje o web komunikácií v počítačových sieťach využívajúcich priame pripojenie ku internetu. Jedná sa zväčša o údaje záznamov o web komunikácií, avšak na zostavenie presných postupností krokov užívateľov na sieti je niekedy potrebné zachytiť a analyzovať aj vybranú časť samotnej web komunikácie v podobe URL odkazov obsiahnutých v kóde web stránok.

Grafickú analýzu web komunikácie môžeme rozdeliť do dvoch hlavných častí a to podľa šírky analyzovaného časového intervalu a vplyvu reprezentovaných dát na výsledné zobrazenie, keďže s narastajúcou dĺžkou vyhodnocovaného časového intervalu sa menia vlastnosti vstupných dát, ktoré je možno analyzovať. Prvou časťou sú dáta z pohľadu najvyššej úrovne, kde sa analýza zameriava na veľké objemy prenášaných dát a veľké časové intervaly (dni, týždne, mesiace). Druhou časťou je analýza strednej úrovne web komunikácie, ktorá predstavuje analýzu prevádzky užívateľov v rámci hodín, minút a sekúnd, čo znamená že sa zaoberá komunikáciou vo forme požiadaviek a odpovedí.

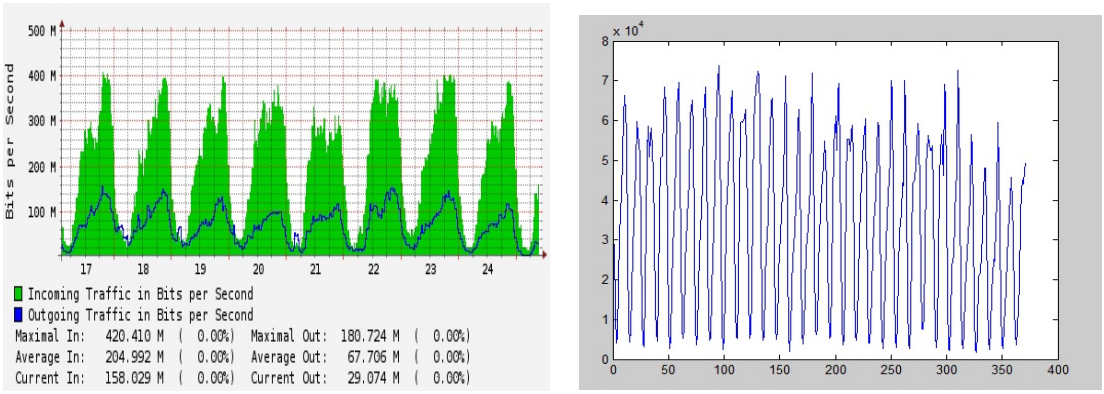
4.1 Najvyššia úroveň

Na tejto úrovni sa analýza zameriava na vstupné dáta, ktoré sú reprezentované svojou veľkosťou v závislosti na jednotke času. Jedná sa o dáta, ktoré reprezentujú sieťovú prevádzku medzi lokálnou sieťou a internetom a hlavne veľkosť prenesených dát za určitý čas. Pomocou matematických metód a postupov je možné tieto dáta spracovať a analyzovať určité procesy sieťovej prevádzky v dlhšom časovom intervale napríklad niekoľkých dní, týždňov, prípadne celých mesiacov.

4.1.1 Grafická reprezentácia sieťovej prevádzky

Je v princípe jednoduché vynesenie objemu prenesených dát na os Y grafu, pričom X je časová os. Grafická reprezentácia má základný informačný charakter, a z daného grafu pozorovateľ môže relatívne jednoducho avšak bez ďalších nástrojov značne nepresne určiť napríklad maximálne a minimálne hodnoty rýchlostí prenosu dát, využitie šírky prenosového pásma a podobne. Príklady grafickej reprezentácie sieťovej

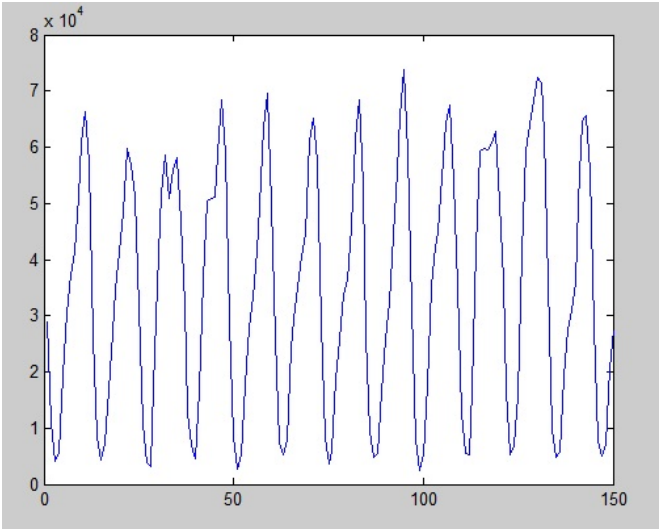
prevádzky sú na Obr. 49 a to reprezentácie sieťovej prevádzky za týždeň (vľavo) a reprezentácia sieťovej prevádzky za mesiac (vpravo).



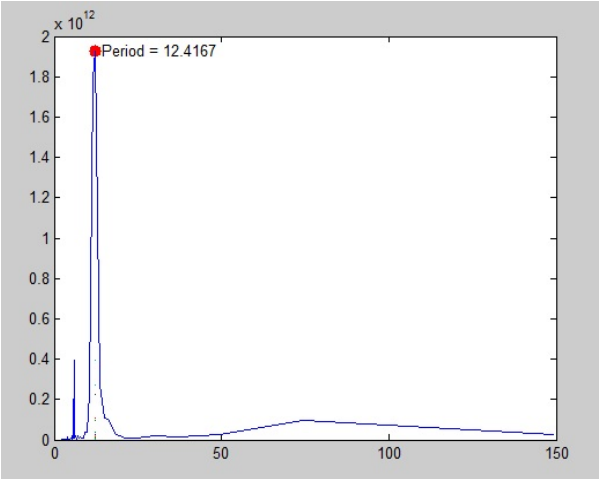
Obr. 49 Týždenný (vľavo) a mesačný graf sieťovej prevádzky

4.1.2 Analýza periodickosti sieťovej prevádzky

Jednou z možností využitia fourierových transformácií pri spracovávaní dát sieťovej prevádzky je vyhľadávanie periodickosti v relatívne neperiodických vstupných dátach. Tieto štatistické dáta sa dajú totiž brať v danom kontexte ako vlnová funkcia, ktorá sa vďaka určitej periodickosti dá analyzovať a pomocou transformácie určiť jej vlastnosti. Jednou z týchto vlastností je aj frekvencia periodickosti, ktorej určenie je matematicky veľmi presné a algoritmus výpočtu pomocou rýchlej fourierovej transformácie je časovo nenáročný. Príklad výpočtu je uvedený na Obr. 50 a Obr. 51.

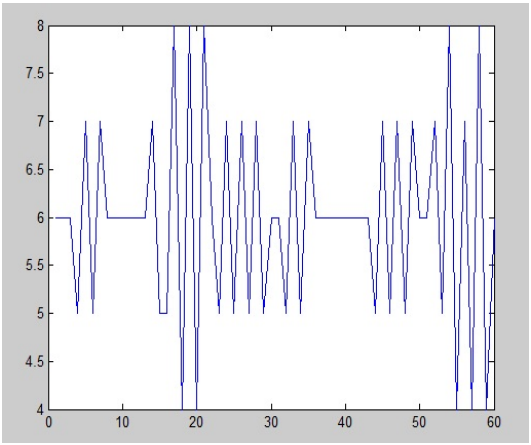


Obr. 50 Graf periodickej sieťovej prevádzky

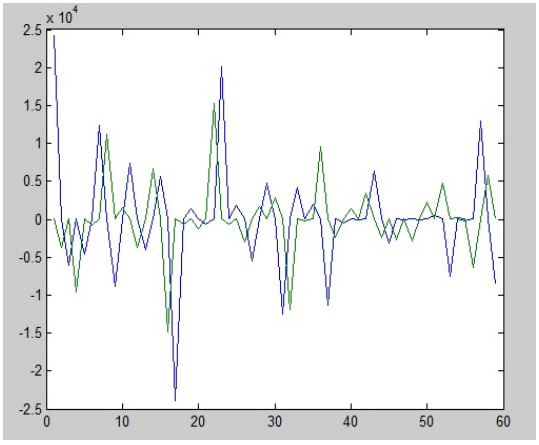


Obr. 51 Určenie frekvencie periodickosti

Ďalšie vlastnosti danej vlnovej funkcie, ktoré sa dajú následne určiť pri periodických dátach sú odchýlka šírky iterácie a odchýlka vrcholov iterácií danej vlnovej funkcie. Príklady analýzy sú uvedené na obrázkoch Obr. 52 a Obr. 53. Na Obr. 53 sú zobrazené odchýlky maximálnych a minimálnych vrcholov od priemernej hodnoty.



Obr. 52 Graf odchýlky šírky iterácií



Obr. 53 Graf odchýlok max a min hodnôt

4.2 Stredná úroveň

Na strednej úrovni sa práca zaoberá analýzou požiadaviek a odpovedí v komunikácií medzi používateľom na lokálnej sieti a web serverom na internete. Na tejto úrovni je možné ako v predchádzajúcom prípade analyzovať napríklad periodickosť spomínanej komunikácie, analyzovať návštevnosť zvolených kategórií web stránok, prípadne pomocou v predchádzajúcich kapitolách popísaných metód vytvoriť systém web odporúčaní.

Vďaka analýze periodickosti sieťovej prevádzky je možné oddeliť periodickú sieťovú komunikáciu od kvázi náhodnej. Separácia periodickej sieťovej prevádzky je dôležitá najmä z hľadiska bezpečnosti užívateľa a operačného systému. Keďže periodickosť sieťovej prevádzky s presne opakujúcich sa intervaloch môže byť vytvorená len aplikáciou, ktorá využíva sieťovú komunikáciu v určitých intervaloch, vzniká potreba hlbšej identifikácie týchto dát z hľadiska možného napadnutia operačného systému rôznymi typmi útokov. Príkladom takéhoto správania môže byť DoS útok, komunikácia škodlivého softvéru typu spyware, adware, prípadne sieťová komunikácia vírusov. Na analýzu a odhalenie periodickosti v sieťovej prevádzke je možné využiť niektorý z typov fourierových transformácií a výpočet entropie, pričom algoritmus sa zameriava napríklad na čas a frekvenciu HTTP alebo iných požiadaviek ktoré sú známe a často sa vyskytujúce v sieťovej prevádzke na internete ako aj v lokálnej sieti.

Analýzou neperiodickej sieťovej komunikácie vytvorenej samotnou činnosťou používateľov ako je prehliadanie stránok je možné využitím rôznych matematických metód navrhnúť a vytvoriť systém web odporúčaní. Tento systém môže pomocou vopred zachytenej a analyzovanej web komunikácie navrhnúť každému nasledujúcemu používateľovi odkaz s web stránkou určitej kategórie, prípadne mu môže ponúknuť reklamný odkaz s určenou dĺžkou zobrazenia, ktorá korešponduje s predpokladanou dĺžkou zotrvania používateľa na danej stránke.

4.2.1 Systémy web odporúčaní

Na internete vždy bolo a je správanie užívateľov založené na určitých odporúčaniach [9]. Jedná sa buď o osobné odporúčania medzi používateľmi, o reklamu alebo vyhľadávače, ktoré používateľovi podľa zadaných parametrov odporučia samotnú

stránku alebo doménu s požadovanými informáciami. S exponenciálnym rastom informačnej domény na internete je opodstatnenosť vhodných web odporúčaní stále viac aktuálna.

Je preto v záujme prevádzkovateľov domén a webov vedieť vytvoriť osobný profil každého užívateľa, prípadne ho zaradiť už do vytvoreného profilu a ponúknuť mu informácie alebo odporúčania, ktoré sa čo najviac zhodujú s jeho preferenciami. Toto je možné dosiahnuť dvoma spôsobmi a to explicitne (napr. pomocou dotazníka na užívateľa) alebo implicitne (sledovaním predchádzajúcej činnosti užívateľa).

Použitím implicitného prístupu sú vo všeobecnosti dosahované lepšie a pravdivejšie výsledky, pričom spôsobov implicitného spracovania dát o užívateľoch je viacero. Vo všeobecnosti je však prístup rovnaký a je založený na vytvorení databázy profilov správania pomocou hľadania podobností. Každému užívateľovi systému je následne priradený určitý profil, podľa ktorého mu sú ponúknuté predvolené odporúčania. Správanie užívateľov je identifikované pomocou tzv. clickstreamu, ktorý určuje aké web stránky prípadne domény užívateľ navštevuje.

4.2.2 Clickstream

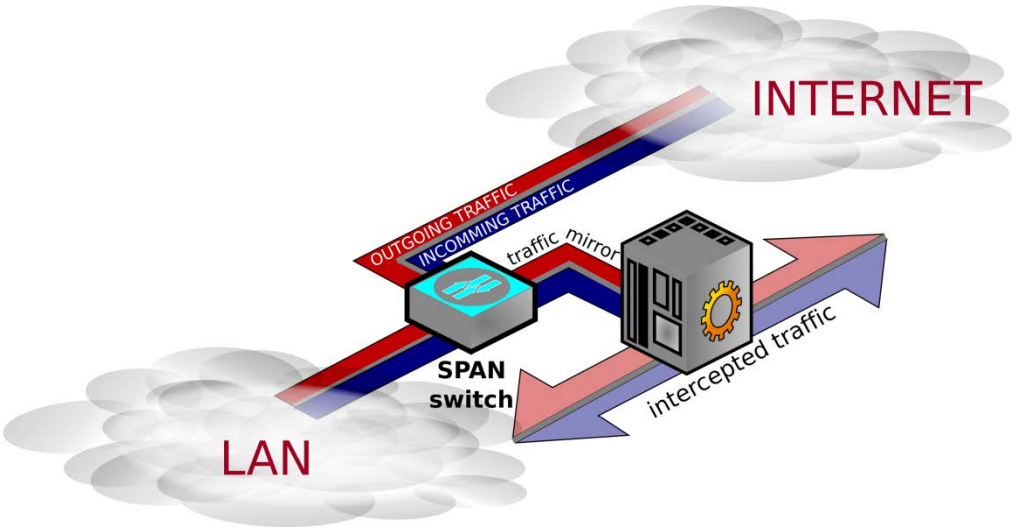
Clickstream v zmysle webových technológií je záznam o činnosti používateľa, hlavne o prezeraní webových stránok. Clickstream je možné odchytiť na strane web server, web prehliadača, alebo na hociktorom bode komunikácie (port mirroring na smerovači). Clickstream analýza je užitočná pre analýzu aktivity na určitom webe, testovanie softvéru alebo výskum trhu. [7]

Najjednoduchšia a najmenej účinná je metóda získavania clickstreamu z log údajov na strane servera. Túto metódu je možné použiť len v rámci jedného servera prípadne domény, pričom informácie o ďalšom pôsobení užívateľa sú nedostupné. Tieto informácie majú teda len lokálny charakter pre každú doménu osobitne. [10]

Pri ďalšej v súčasnosti najviac používanej metóde vzniká problém komunikácie a odovzdávania informácií z webového prehliadača. Problém je rovnaký ako pri predošlej metóde, a to že prevádzkovateľ určitej domény vie získať informácie len o činnosti užívateľa na danej doméne. Čiastočné riešenie tohto problému je možné použitím jednoduchých javascript aplikácií umiestnených na stránkach rôznych domén, pričom kód daného javascriptu môže zapisovať a využívať súbory cookies a zároveň

ukladať alebo odosielať rôzne informácie o používateľovi. Využitím takéhoto systému javascript aplikácií je možné po umiestnení ich dostatočne veľkého počtu na rôznych stránkach výrazne zvýšiť rozsah získaných informácií. Pomocou informácií uložených týmto spôsobom je mnohokrát možné zostaviť rozsiahlu históriu navštívených stránok užívateľa. Samotný javascript kód pritom môže byť nenápadne implementovaný do reklamy alebo neškodne vyzerajúceho odkazu na sociálnu sieť (ako napr. Facebook Like odkaz). Aj tento postup však môže vo viacerých prípadoch zlyhať. Ak má napríklad používateľ zakázanú podporu cookies, prípadne ak používa antivírový softvér, ktorý považuje sledovacie cookies aplikácie za nebezpečné a odstraňuje ich.

V tejto práci sa zameriame hlavne na využitie tretej a najúčinnejšej metódy získavania clickstreamu užívateľov. Na jeho získanie je použitá technika takzvaného port mirroringu, pričom touto metódou je možné odchytiť väčšinu HTTP komunikácie a jej analýzou zostaviť samotný clickstream každého užívateľa siete. Túto metódu je možné využívať aj v rámci internetu, avšak nato sú potrebné ďalšie metódy, ako napríklad použitie proxy servera. Na naše účely však postačuje testovacia LAN sieť s rôznymi užívateľmi, ktorá má reálne pripojenie do internetu. Model získania clickstreamu pomocou port mirroringu na prepínači je zobrazený na Obr. 54.



Obr. 54 Získanie clickstreamu pomocou port mirroringu

4.2.3 Vytvorenie clickstreamu

Komunikácia používateľov na internete pozostáva z veľkého množstva rôznych protokolov, správ a širokého spektra prenášaných typov dát. Táto kapitola sa zameriava hlavne na analýzu HTTP komunikácie medzi koncovým používateľom a internetovými servermi. Z veľkého množstva prenášaných dát je preto potrebné selektovať údaje o HTTP komunikácií a to konkrétne požiadavky (GET/POST) od používateľov a následné odpovede od web serverov. Na odchytávanie a počiatočnú selekciu a analýzu dát bol použitý nástroj tshark, ktorý je konzolovou verziou jedného z najznámejších open-source analyzátorov sieťovej prevádzky. Pomocou tohto nástroja je možné nielen odchytiť sieťovú prevádzku pomocou port mirroringu, ale využiť jeho široké možnosti analýzy na získanie potrebných údajov ako napríklad zostaviť kompletné URL požiadavky, získať HTTP REFERER, získať zdrojovú, cieľovú adresu odkazu, sekvenčné čísla TCP spojení, prípadne získať dekódovaný obsah tela HTTP balíčkov. Pomocou týchto informácií je následne možné podľa požadovaných kritérií zostaviť pre každého užívateľa samotný clikstream. Pred samotným zostavením clickstreamu roztriedime komunikáciu podľa zdrojových a cieľových IP adries a ďalej už pracujeme s dátami pre každého používateľa osobitne.

Pri zostavovaní clickstreamu je v prvom rade je potrebné vytvoriť páry správ typu požiadavka/odpoveď. Zostavenie párov je možné vďaka analýze TCP sekvenčných čísel, pričom mechanizmus poradových čísel zabezpečuje možnosť presného priradenia odpovede ku požiadavke. Po vytvorení daných párov sa nadbytočná komunikácia môže zahodiť, keďže je pre ďalšiu analýzu nepotrebná.

Ďalším krokom v zostavení clickstreamu je vytvorenie relácií. Relácia v danom kontexte znamená zoznam dvojíc (požiadavka/odpoveď), ktoré na seba nadväzujú podľa určitých pravidiel. Rozdelenie clickstreamu na relácie je dôležité kvôli analýze samotného clickstreamu, respektíve porovnávaniu správania používateľov. Jedna relácia by mala vo všeobecnosti predstavovať súvislú činnosť používateľa, ktorá je prerušená maximálne na čas, ktorý je určený ako priemerný najdlhší čas potrebný na prečítanie obsahu bežnej web stránky. Všetka komunikácia nasledujúca po danom maximálnom časovom intervale je považovaná za ďalšiu reláciu. Takto vytvorené relácie môžeme ďalej analyzovať, porovnávať a slúžia ako vstupné dáta pre vyhodnocovanie web odporúčaní.

Pri zostavení clicstreamu je ďalej potrebné odstránenie nepotrebných HTTP požiadaviek a odpovedí, ktoré vznikajú napríklad pri načítavaní reklám, pohyblivých rámcov, obrázkov, počítadiel a podobne. Analýza clickstreamu totiž vyžaduje získanie iba ozajstných manuálne otvorených stránok, na ktoré užívateľ sám klikol, prípadne manuálne zadal URL adresu danej domény (stránky).

Na odstránenie nadbytočných dvojíc z relácie môžeme využiť kombináciu dvoch postupov. Prvým postupom je sledovanie poľa, ktoré je obsiahnuté v hlavičke HTTP balíčka s názvom HTTP REFERER. Toto pole sa nachádza v hlavičke každej HTTP požiadavky a je vyplnené automaticky webovým prehliadačom. Jeho význam je odkazovanie na pôvodnú HTTP URL, na ktorej sa nachádzal či už hypertextový alebo iný odkaz, ktorý vygeneroval danú požiadavku. V reálnom využití to znamená, že po zadaní URL do prehliadača sa pošle požiadavka obsahujúca prázdne pole HTTP REFERER. Každá nasledujúca stránka na ktorú klikneme, prípadne každá ďalšia požiadavka na stiahnutie dodatočného obsahu stránky (obrázok, reklama) bude v poli HTTP REFERER obsahovať URL stánky pôvodu daného odkazu. [13]

Využitie len tejto metódy filtrovania obsahu je však značne nedostatočné, pretože aj požiadavka na každú ďalšiu regulárnu stránku na ktorej odkaz užívateľ klikne bude v poli HTTP REFERER obsahovať URL predchádzajúcej stránky. Takto by sme filtráciou stratili možnosť zachytiť každú regulárnu stránku okrem tej, ktorej URL bola užívateľom manuálne zadaná do prehliadača. Preto využijeme druhú možnosť filtrovania HTTP komunikácie a to sledovanie vybranej časti obsahu HTTP odpovedí, pričom sa zameriavame len na hypertextové a iné druhy odkazov. Pomocou počtu týchto odkazov v texte HTTP odpovede je možné určiť či sa jedná o regulárnu stránku, alebo len o súčasť stránky načítanej dodatočne. Experimentálne bola minimálna hodnota počtu odkazov na regulárnej stránke stanovená na 20. Pomocou tejto metódy vieme následne nielen vyselektovať iba regulárne stránky z množstva HTTP požiadaviek a odpovedí, ale využitím poľa HTTP REFERER a získaných odkazov z obsahu HTTP odpovedí vieme vytvoriť relácie, logické spojenia popisujúce následnosť aktivity používateľa.

4.2.4 Analýza clickstreamu

Pri analýze samotného clickstreamu spôsobuje najväčší problém škálovateľnosť obsahu. Čo v reálnom systéme znamená že stále pribúdajú nové web stránky, prípadne

užívateľ navštívi doménu na ktorej nikdy predtým nebol. Tento problém vieme čiastočne vyriešiť rozdelením celej informačnej domény na 2 časti a to dôveryhodné a nedôveryhodné domény a tie následne riešiť samostatne.

Základom analýzy clickstreamu v systémoch web odporúčaní je porovnávanie clikstreamu užívateľov medzi sebou, buď online alebo offline pomocou pred pripravenej vzorky. Rozdiel medzi online a offline analýzou je ten, že v online analýze sa zbierajú a vyhodnocujú dáta priamo od užívateľov v reálnom čase, a v offline režime prebieha analýza nad vopred zozbieranými dátami.

Keďže pomocou port mirroringu je možné zozbierať najväčšie množstvo údajov potrebných pre analýzu, samotná rôznorodosť týchto dát spôsobuje najväčší problém analýzy. V našom prípade bol problém rôznorodosti obsahu kompenzovaný použitím kategorizácie, pričom v možnostiach tejto práce bolo kategorizovať iba veľmi malú časť celkového obsahu web komunikácie skúmanej LAN siete.

Kategorizácia web stránok prebiehala v troch fázach. Prvou fázou bolo overenie, či sa daná stránka (doména) nachádza v oficiálnom zozname vydanom spoločnosťou Google Top1000 [31]. Tento zoznam je tvorený 1000 najvyhľadávanejšími stránkami na svete, pričom ak daná doména spadá do tohto zoznamu, je jej priradená kategória reprezentovaná daným číslom. Ak doména nespadá do daného zoznamu, je zaradená do listu nekategorizovaných stránok. V druhej fáze bolo manuálne kategorizovaných 70 najviac navštevovaných stránok (zväčša české a slovenské domény). Pri reálnom nasadení systému by bolo potrebné automatizovať kategorizáciu stránok buď pomocou rôznych vyhľadávačov, prípadne sa zamerať len na komerčné využitie a kategorizovať len zadané stránky klientov. V našom systéme sú teda všetky nekategorizované stránky reprezentované číslom kategórie 0. Príklad kategorizácie je zobrazený v Tab. 2 a Tab. 3. Číselné označovanie kategórií je potrebné kvôli neskoršiemu výpočtu entropie medzi postupnosťou domén v clickstreame užívateľa. Spolu obsahuje navrhnutý systém viac ako 1070 domén v 232 kategóriách.

Výsledkom doterajšej analýzy je teda vytvorenie clickstreamu pre každého užívateľa, pričom tento clikstream pozostáva z relácií, ktoré obsahujú na seba nadväzujúce dvojice HTTP požiadaviek a odpovedí, pričom v každej relácií sú zachytené aj informácie o dĺžke trvania, počtu klikou užívateľa, ich frekvencií, počtu navštívených stránok a kategóriách daných stránok.

Tab. 2 Kategorizácia stránok

Doména	Kategória
facebook.com	Social Networks
youtube.com	Online Video
yahoo.com	Web Portals
live.com	Search Engines
msn.com	Web Portals
wikipedia.org	Dictionaries & Encyclopedias
sme.sk	News
tuke.sk	Education

Tab. 3 Číselné označenie kategórií

Index	Kategória
1	Web Portals
2	Social Networks
3	File Sharing & Hosting
4	News
5	Sports News
6	Movie Reference

4.2.5 Analýza clikstreamu pomocou výpočtu entropie

V predchádzajúcich kapitolách sme ukázali, že výpočet entropie môže byť veľmi užitočný pri grafickej analýze rôznych druhov dát. Preto sme použili tento druh analýzy aj pri analýze zostaveného clicstreamu používateľov skúmanej lokálnej siete. Pri analýze clickstreamu predstavuje entropia mieru podobnosti medzi užívateľmi, ktorý vystupujú v procese generovania webových odporúčaní [9]. Keďže sa jedná o diskrétne hodnoty je entropia vyjadrená pomocou nasledujúcej rovnice

$$H(D(U_t,U_x)) = -\sum_{i=1}^{n} p(d_i)\log_2 p(d_i)$$

kde, U_x reprezentuje reláciu clickstreamu používateľa, ktorá bola odchytená skôr a slúži ako vzor, a U_t predstavuje reláciu clicstreamu užívateľa, ktorému chceme odporučiť cieľovú kategóriu. Potom $D(U_t,U_x)$ je rozdiel medzi ohodnoteniami cieľového užívateľa U_t a užívateľa U_x pre n unikátnych kategórií a $p(d_i)$ je funkcia pravdepodobnosti rozdielu ohodnotení. Tieto pravdepodobnosti popisujú do akej miery sú ciele užívateľa U_t podobné cieľom užívateľa U_x. Nižšia hladina entropie znamená logicky vyšší stupeň podobnosti cieľov užívateľov.

Finálna analýza pozostáva z troch krokov:

a.) Príprava dát: v prvom kroku sa vyššie popísaným postupom pre každého užívateľa zostaví z clickstreamu pole relácií.

b.) Výber dôveryhodných relácií vhodných pre vytvorenie odporúčaní: tento krok je možné rozdeliť do dvoch častí. V prvej časti je potrebné vypočítať entropiu medzi reláciou cieľového používateľa a reláciami ostatných používateľov. Hodnota entropie je vypočítaná pomocou vzorca

$$H(D(U_t,U_x)) = -\sum_{i=1}^{n} p(d_i)\log_2 p(d_i)$$

Výpočtom entropie medzi týmito reláciami dostaneme pole obsahujúce celú reláciu a hodnotu entropie v rozmedzí od 0 po 1, pričom 0 je hodnota najnižšej entropie, čiže najvyššej usporiadanosti a naopak hodnota 1 predstavuje úplnú neusporiadanosť, čiže v našom prípade minimálny predpoklad podobnosti daných relácií. Z tohto poľa následne vyberieme všetky relácie, ktorých vypočítaná hodnota entropie je nižšia ako 0,64. Tento koeficient bol stanovený experimentálne ako najvyššia povolená hodnota entropie, pričom všetky relácie s vyššou entropiou už nemajú dostatočnú podobnosť s reláciou cieľového používateľa. Pri hodnote entropie vyššej ako 0.64 sa porovnávané relácie už zhodujú maximálne len v jednej kategórií a daná dvojica relácií je pre účely odporúčania nepoužiteľná. V druhej časti je potrebné zostaviť z ostávajúceho poľa relácií návrhy odporúčaných stránok, pričom každý návrh je ohodnotený váhou, ktorá je vypočítaná ako

$$D(R(U_X)) = \sum_{1}^{n}(1 - \mathrm{Ent}(U_X, U_T))$$

,kde $R(U_X)$ predstavuje nasledujúcu kategóriu vhodnú na odporúčanie, $D(R(U_X))$ je stupeň dôležitosti kategórie, n je počet výskytov nasledujúcej kategórie $R(U_X)$ a $Ent(U_x,U_t)$ je hodnota entropie vypočítaná medzi reláciou cieľového užívateľa a reláciou užívateľa U_x . Výsledkom tejto rovnice je pole odporúčaných kategórií s príslušnou hodnotou stupňa dôležitosti.

c.) Výber najvhodnejších stránok podľa stupňa dôležitosti“ Z vytvoreného zoznamu v bode b.) sa vytvorí zoznam odporúčaní, ktoré sú následne podľa stupňa dôležitosti ponúknuté cieľovému užívateľovi. Stupeň dôležitosti je určený podľa frekvencie výskytu a podmienok ako napríklad či už bola daná stránka cieľovým užívateľom v minulosti navštívená alebo nie.

4.2.6 Štatistické metódy analýzy web komunikácie

Pri analýze clickstreamu používateľov bolo získané okrem samotných relácií aj veľké množstvo ďalších údajov ako sú čas, ktorý strávil používateľ na danej stránke, počet a frekvencia kliknutí užívateľa, počet stránok načítaných z jednej domény a podobné detailné informácie. Tieto údaje je možné využiť pri návrhu dodatočnej funkcionality systému web odporúčaní. Použitím štatistickej metódy logickej regresie bol navrhnutý systém, ktorý by bol schopný pomocou týchto údajov napríklad odporučiť reklamnú upútavku, ktorej trvanie by bolo určené a smerované tak, aby čo najlepšie vystihovalo správanie používateľa a zároveň ho čo najmenej daná reklama obmedzovala.

Za základnú podmienku aby daný reklamný odkaz čo najmenej obmedzoval užívateľa sme zvolili odhad pravdepodobnosti, či užívateľ ostane alebo neostane na danej stránke dlhšie ako 1 minútu. Na výpočet pravdepodobnosti tohto odhadu boli použité vo finálnej rovnici použité nasledujúce hodnoty získané analýzou clickstreamu. Prvou hodnotou je dĺžka trvania celej relácie (ďalej označovaná CAS), druhou závislou premennou je celkový počet zobrazených stránok užívateľa počas trvania relácie (ďalej označovaná CLIK) a poslednou závislou premennou je celkový počet navštívených domén počas relácie (ďalej označovaná DOM). [32]

Pomocou týchto premenných bol pomocou logickej regresie z veľkého množstva odchytených dát reálnej komunikácie navrhnutý model, ktorý určí pravdepodobnosť, či užívateľ zostane na danej stránke viac ako minútu alebo nie.

Rovnica odhadu log je nasledovná

$$logit(\pi_i) = -0.40307 + 0.000233 * \text{CAS} + 0.077324 * \text{CLIK} - 0.184449 * \text{DOM}$$

Z danej rovnice vyplýva, že pravdepodobnosť že užívateľ zostane ostane viac ako 1min je určená vzorcom

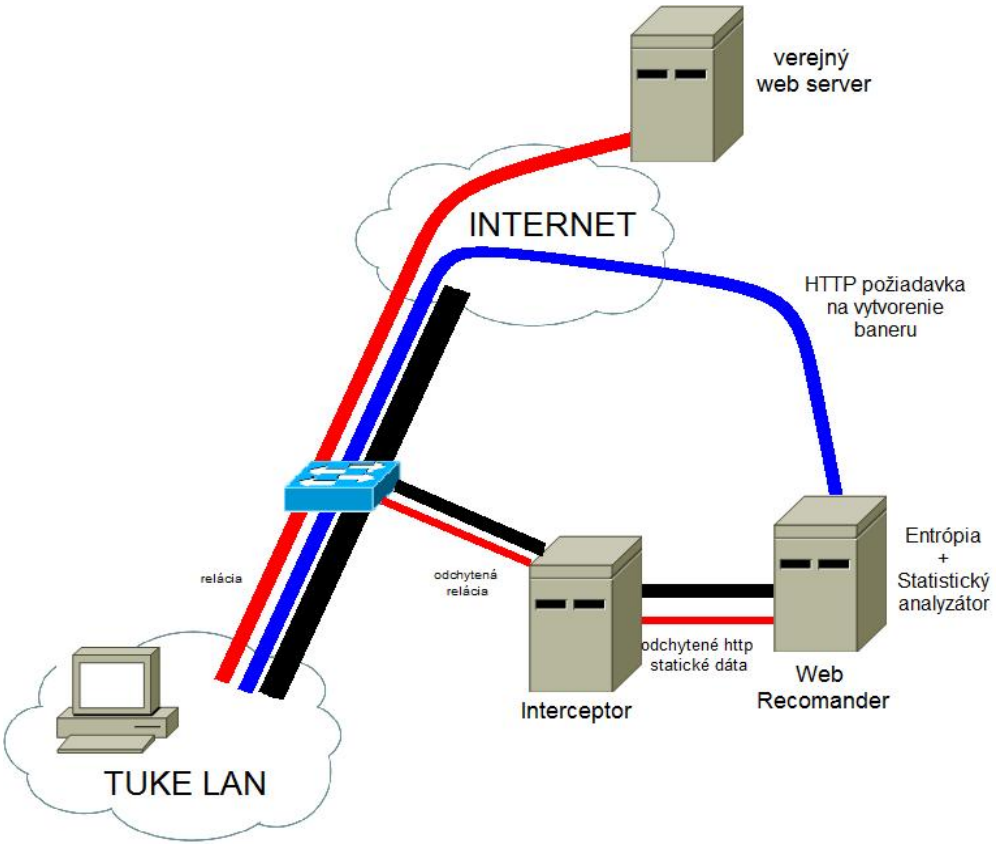
$$\pi_i = \frac{\exp\{-0.40307 + 0.000233 * \text{CAS} + 0.077324 * \text{CLIK} - 0.184449 * \text{DOM}\}}{1 + \exp\{-0.40307 + 0.000233 * \text{CAS} + 0.077324 * \text{CLIK} - 0.184449 * \text{DOM}\}}$$

Použitím tohto vytvoreného modelu na ďalšie nezávislé vzorky clickstreamu užívateľov však bolo jeho výsledkom 96% zamietnutí hypotézy, že používateľ zostane na danej stránke viac ako 1 minútu. Presnosť daného modelu pri odhadovaní bola stanovená na 65%, čo znamená že daný model nemôžeme vyhlásiť za dostatočne vierohodný. [32]

Pri takomto vysokom počte zamietnutí hypotézy je na danej vzorke možno povedať, že aplikácia tohto modelu s využitím daných premenných pre danú hypotézu nemá v tomto prípade opodstatnenie. Logicky teda môžeme z tejto analýzy vysloviť záver, že pomocou daných hodnôt nie je možné zostaviť model, ktorý by v požadovanej miere učil dĺžku zotrvania užívateľa na danej stránke.

4.2.7 Implementácia systému web odporúčaní

V predchádzajúcich podkapitolách sme ukázali ako pomocou výpočtu entropie a použitia štatistických metód môžeme pre vybraného používateľa vygenerovať odporúčanie podľa jeho predchádzajúcej aktivity. Po odchytení komunikácie pomocou interceptora a port mirroringu je možné v určitom časovom intervale vygenerovať odporúčania pre každého užívateľa, ktorý splnil určité podmienky (pri vytváraní relácií).Výsledná topológia zapojenia systému web odporúčaní je zobrazená na Obr. 55.



Obr. 55 Topológia implementácie systému web odporúčaní

Načrtnutá topológia pozostáva z dvoch sietí (LAN, WAN), na ktorých rozhraní sa nachádza aktívne sieťové zariadenie mirrorujúce oba smery sieťovej prevádzky. Túto prevádzku následne odchytáva a analyzuje interceptor, ktorý v pravidelných intervaloch odosiela z časti analyzované dáta samostatnému serveru na vytvorenie web odporúčaní. Takýmto spôsobom je možné za určitých podmienok vytvoriť pre užívateľov práve komunikujúcich na internete odporúčania najvhodnejších stránok (kategórií).

Po vytvorení odporúčaní je následne potrebné tieto odporúčania ponúknuť danému používateľovi. Jedna z možností ako ponúknuť danému používateľovi odporúčanie s danou stránkou je využitie reklamného prvku umiestneného na stránke verejného web servera. Po načítaní stránky daného web servera sa vytvorí spojenie medzi počítačom a verejným web serverom (spojenie označené červenou farbou), zároveň s ním sa však vytvorí ďalšie spojenie, ktoré načíta obsah reklamného prvku, avšak tento prvok sa nenachádza na verejnom web serveri, ale na servery web odporúčaní, ktorý vygeneruje obsah s daným web odporúčaním a ponúkne ho tak používateľovi. Relácia pre načítanie reklamného prvku je zobrazená modrou farbou.

4.2.8 Výsledky analýzy web komunikácie

Pri zostavení systému web odporúčaní bolo prezentované, že použitím vybraných grafických a štatistických metód analýzy dát je možné pre každého užívateľa vytvoriť odporúčanie najvhodnejšej nasledujúcej stránky. Keďže na získanie reálnych výsledkov využitia systému web odporúčaní by bolo potrebné okrem analýzy reálnej web prevádzky využiť aj dynamicky generovaný odkaz umiestnený na často navštevovanej web stránke. Bez spätnej väzby, ktorá by bola vytvorená kliknutím užívateľa na odkaz navrhovanej internetovej stránky je vyslovenie záveru o percentuálnej úspešnosti vytvorených návrhov neobjektívny a nemôžeme taktiež vysloviť záver o vhodnosti použitia navrhnutého systému. V rámci rozsahu tejto práce bolo prezentovanie možností použitia grafických metód analýzy dát, čo bolo ich použitím v plnom rozsahu splnené.

Taktiež bolo prezentované, že použitím štatistickej metódy logistickej regresie je možné vytvoriť model, ktorý s určitou pravdepodobnosťou určí podľa vstupných parametrov možný výsledok zadanej hypotézy. V tejto práci uvedená hypotéza a na nej vytvorený model má síce požadovanú presnosť, avšak nespĺňa požadované podmienky použitia. Preto by bola potrebná ďalšia analýza možností použitia tejto metódy s využitím ďalších premenných získaných analýzou web komunikácie.

5 Záver

Hoci je dnešný celosvetový trend rýchleho rozvoja informačných technológií považovaný za pozitívny, tempo akým vznikajú nové možnosti vývoja ako hardvéru tak hlavne softvéru so sebou prináša rovnako rýchlo vznikajúce nové typy hrozieb, ktorých detekcia a eliminácia je taktiež čoraz náročnejšia a zložitejšia. Takmer každodenne vyskytujúce sa nové typy hrozieb využívajú čoraz prepracovanejšie metódy znemožnenia ich odhalenia a vývojári zabezpečovacích systémov musia stále siahať po nových metódach, ktoré sú schopné tieto hrozby eliminovať.

Táto práca poskytuje teoretické východiská a zároveň prakticky prezentuje možnosti využitia grafických a matematických metód, ktoré poskytujú dosiaľ málo preskúmané možnosti v oblasti bezpečnosti informačných technológií. V tomto konkrétnom prípade poskytujú dané metódy nové možnosti v oblasti detekcie škodlivého softvéru ako jednej z najvýznamnejších súčasných bezpečnostných hrozieb. Výsledky prezentované v práci dokazujú veľký potenciál využitia daných metód pri rozpoznávaní typov neznámych súborov. Hlavným prínosom tejto práce je však preukázanie, že využitím metód výpočtu entropie a frekvenčnej analýzy je možné dosiahnuť podobné alebo lepšie výsledky ako použitím konvenčných metód detekcie škodlivého softvéru.

Na druhej strane zložitosť, rôznorodosť a náročnosť použitých metód predstavuje nemalú prekážky reálneho nasadenia spomenutých metód v praxi. V ďalších možných rozšíreniach práce venovaných tejto problematike bude potrebné zamerať sa na vyriešenie rôznych problémov implementácie a navrhnutie nových a účinnejších metód analýzy dát pre dosiahnutie ešte presnejších výsledkov. Medzi jeden z najväčších problémov implementácie patrí potreba manuálnej interakcie vývojára počas určovania samotných podmienok detekcie. V niektorých prípadoch sa jedná o manuálne prechádzanie výsledkov grafickej analýzy niekoľkých tisícov súborov. Tento problém by bolo možné odstrániť pomocou matematických a štatistických metód alebo napríklad využitím neurónových sietí [30]. Medzi ďalšie metódy vhodné pre danú analýzu patrí taktiež využitie výpočtu komplexnosti analyzovaných dát [20], využitie metódy kompresívneho snímania [29], prípadne rozvinutie možností využitia štatistickej analýzy obsahu dát [22].

V poslednej časti práce sú popísané možnosti využitia pokročilých metód grafickej analýzy dát v prostredí počítačových sietí. Viacerými návrhmi a praktickou implementáciou vybraných metód grafickej analýzy bola taktiež ukázaná opodstatnenosť a význam ich použitia v danej oblasti. Práca sa zameriavala hlavne na grafickú analýzu web komunikácie, kde bolo výsledkom implementácie týchto metód zostavenie reálneho systému web odporúčaní.

Zoznam použitej literatúry

[1] OPPENHEIM, Alan V. - SCHAFER, R. W. - BUCK, J. R. Discrete-time signal processing. Upper Saddle River, N.J.: Prentice Hall, 1999, ISBN 0-13-754920-2.

[2] UYSAL, Faruk. Short Time Fourier Transform (STFT) [online]. 2010 [cit 2011-05-25]. Dostupné na internete <http://farukuysal.net/STFT.aspx>

[3] PUNSKAYA, Elena. Fast Fourier transform (FFT) [online]. 2007 [cit 2012-04-25]. Dostupné na internete <http://www-sigproc.eng.cam.ac.uk/~op205/3F3_3_Fast_%20Fourier_Transform.pdf>

[4] IHARA, Shunsuke. Information Theory for Continuous Systems. World Scientific, 1993, s. 2-56. ISBN 9810209851

[5] TANEJA, Inder. 2001. Measures of Uncertainty: Shannon's Entropy. [cit. 2011-05-26]. Dostupné na internete: <http://www.mtm.ufsc.br/~taneja/book/node5.html>. ISSN 88.040-900

[6] PHILLIPS, W.J. 2003. Time-Frequency Analysis. [cit. 2011-05-26]. Dostupné na internete: < http://www.engmath.dal.ca/courses/engm6610/notes/node3.html>.

[7] SKOK, Gavin. Establishing a legitimate expectation of privacy in clickstream data [online]. 2000 [cit 2012-04-25]. Dostupné na internete <http://cyber.law.harvard.edu/privacy/PrivacyInClickstream%28Skok%29.htm>

[8] AKANSU, Ali N – HADDAD, Richard A. Multiresolution Signal Decomposition: Transforms, Subbands, and Wavelets. Academic Press, 2001, s. 391-437. ISBN 0120471418.

[9] MEHTA, Harita, BHATIA, Shveta Kundra, BEDI, Punam 2011. Collaborative Personalized Web Recommender System using Entropy based Similarity Measure. 2011 [cit 2012-01-30]. Dostupné na internete: <http://arxiv.org/ftp/arxiv/papers/1201/1201.4210.pdf>.

[10] GÜNDÜZ, S¸ule. A Web Page Prediction Model Based on ClickStream Tree Representation of User Behavior [online]. 2003 [cit 2012-01-30]. Dostupné na internete: <http://www.cs.uwaterloo.ca/~tozsu/publications/web/kdd03.pdf >.

[11] PHILLIPS, Dr. W. J. Time-Scale Analysis [online]. 2003 [cit 2012-01-30]. Dostupné na internete: < http://www.engmath.dal.ca/courses/engm6610/notes/node4.html >.

[12] DAVIS, Wilbon P. – HALL, Gregory A. Sliding Window Measurement for File Type Identification [online]. 2007 [cit 2012-04-25]. Dostupné na internete: < http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.113.8439&rep=rep1&type=pdf >.

[13] RFC 2616 HTTP/1.1 : 14.36 Referer.

[14] RIECK, Konrad – HOLZ, Thorsten – WILLEMS, Carsten - D¨USSEL, Patrick – LASKOV, Pavel. Learning and Classification of Malware Behavior [online]. 2007

[cit 2012-04-25]. Dostupné na internete: < http://pi1.informatik.uni-mannheim.de/filepool/publications/malware-classification-dimva08.pdf >.

[15] ROUSSEV, Vassil – GARFINKEL, Simson L. File Fragment Classification—The Case for Specialized Approaches [online]. 2010 [cit 2012-04-25]. Dostupné na internete: < http://simson.net/clips/academic/2009.SADFE.Fragments.pdf>.

[16] CONTI, Gregory – BRATUS Sergey – SHUBINA, Anna – SANGSTER, Benjamin – RAGSDALE, Roy – SUPAN, Matthew – LICHTBERG, Andrew – PEREZ-ALEMANY, Robert. Automated mapping of large binary objects using primitive fragment type classification [online]. 2010 [cit 2012-04-25]. Dostupné na internete: < http://www.dfrws.org/2010/proceedings/2010-301.pdf>.

[17] MATHWORKS, Inc. Continuous Wavelet Transform [online]. 2012 [cit 2012-04-25]. Dostupné na internete: < http://www.mathworks.com/help/toolbox/wavelet/gs/f3-1000759.html>.

[18] DAVIS, Tom. Utilizing Entropy to Identify Undetected Malware. Guidance Software, 2009, s. 3-10. Dostupné na internete: < http://image.lifeservant.com/siteuploadfiles/VSYM/99B5C5E7-8B46-4D14-A53EB8FD1CEEB2BC/43C34073-C29A-8FCE-4B653DBE35B934F7.pdf >.

[19] KENDALL, Kris - McMILLAN, Chad. Practical Malware Analysis. Mandiant Intelligent Information Security. Dostupné na internete: <http://www.blackhat.com/presentations/bh-dc-07/Kendall_McMillan/Presentation/bh-dc-07-Kendall_McMillan.pdf >.

[20] NOH, Hanyoung. Complexity-based Packed Executable Classification with High Accuracy. School of Engineering Information and Communications University, 2009. Dostupné na internete: < http://caislab.kaist.ac.kr/publication/thesis_files/2009/Thesis_Hanyoung.pdf >.

[21] LYDA, Robert - HAMROCK, James. Using Entropy Analysis to Find Encrypted and Packed Malware. The IEEE Computer Society, 2007. ISSN 1540-7993/07. Dostupné na internete: < http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.120.9861&rep=rep1&type=pdf >.

[22] TABISH, S. Mornina - SHAFIQ, M. Zubair - FAROOQ, Muddassar. Malware Detection using Statistical Analysis of Byte-Level File Content. National University of Computer & Emerging Sciences, Pakistan 2009. ACM 978-1-60558-669-4. Dostupné na internete: < http://www.cse.msu.edu/~tabishsy/papers/momina_CSIKDD.pdf >.

[23] Microsoft Security Intelligence Report - Volume 11.Microsoft Corporation, 2011. Dostupné na internete: < http://download.microsoft.com/download/0/3/3/0331766E-3FC4-44E5-B1CA-2BDEB58211B8/Microsoft_Security_Intelligence_Report_volume_11_English.pdf >.

[24] CALHOUN, C. William - COLES, Drue. Predicting the Types of File Fragments. Department of Mathematics, Computer Science and Statistics Bloomsburg, University of Pennsylvania Bloomsburg, 2008. Dostupné na internete: < http://www.dfrws.org/2008/proceedings/p14-calhoun_pres.pdf >.

[25] JAVED, Mobin - ASHFAQ, Binte Ayesha - SHAFIQ, M. Zubair - KHAYAM, Ali Syed. On the Inefficient Use of Entropy for Anomaly Detection. National University of Sciences & Technology, Islamabad, Pakistan 2009. Dostupné na internete: < http://wisnet.seecs.nust.edu.pk/publications/2009/Mubeen_RAID09.pdf >.

[26] GRÖNING, Michael. New Approaches on Malware - Detection on Mobile Devices. Hamburg University of Applied Sciences, 2010. Dostupné na internete: < http://inet.cpt.haw-hamburg.de/events/slides/Vortrag-AW1-2.pdf >.

[27] SCHMIDT, C. Thomas - WÄHLISCH, Matthias - GRÖNING, Michael. Context-adaptive Entropy Analysis as a Lightweight Detector of Zero-day Shellcode Intrusion for Mobiles. Hamburg University of Applied Sciences, 2011. Dostupné na internete: <http://www.sigsac.org/wisec/WiSec2011/wisec2011-poster-3-schmidt-mobile-ids.pdf >.

[28] LANDESMAN, Mary. What is a Virus Signature? [online]. 2009 [cit 2011-05-25]. Dostupné na internete < http://antivirus.about.com/od/whatisavirus/a/virussignature.htm>.

[29] DAVENPORT, Mark A. – DUARTE, Marco F. - WALKIN, Michael B. TAKHAR Jason N. Laskar Dharmpal – KELLY, Kevin F. – BARANIUKR Richard G. The Smashed Filter for Compressive Classification and Target Recognition [online]. 2007 [cit 2012-04-25]. Dostupné na internete < http://dsp.rice.edu/sites/dsp.rice.edu/files/cs/spie07-final.pdf >.

[30] STERGIOU, Christos – SIGANOS, Dimitrios. Neural Networks [online]. 1999 [cit 2011-05-25]. Dostupné na internete < http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html>.

[31] GOOGLE, Inc. The 1000 most-visited sites on the web [online]. 2011 [cit. 2012-04-25]. Dostupné na internete < http://www.google.com/adplanner/static/top1000/#>.

[32] HRONČÁK, Ján. Logistická regresia. Bratislava : Univerzita Komenského v Bratislave 2012 [cit. 2012-04-25]. Diplomová práca.

Prílohy

Príloha A: CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe.

Príloha B: Používateľská príručka (Systém detekcie škodlivého softvéru)

Príloha C: Systémová príručka (Systém detekcie škodlivého softvéru)

Príloha D: Používateľská príručka (Systém web odporúčaní)

Príloha E: Systémová príručka (Systém web odporúčaní)