SLOVENSKÁ TECHNICKÁ
UNIVERZITA V BRATISLAVE
FAKULTA ELEKTROTECHNIKY
A INFORMATIKY

Ing. Stanislav Marček

Autoreferát dizertačnej práce

Využitie kaskádovej klasifikácie pre detekciu chýb, systémových zlyhaní a sieťových prienikov

na získanie akademickej hodnosti philosophie doctor, PhD.
v doktorandskom študijnom programe
9.2.9 Aplikovaná informatika

Bratislava 2016

Dizertačná práca bola vypracovaná v dennej forme doktorandského štúdia

Školiace pracovisko: Ústav informatiky a matematiky

Predkladateľ: Ing. Stanislav Marček
Fakulta elektrotechniky a informatiky
Slovenská technická univerzita v Bratislave

Školiteľ: Dr. rer. nat. Martin Drozda
Fakulta elektrotechniky a informatiky
Slovenská technická univerzita v Bratislave

Oponenti: doc. Ing. Zoltán Balogh, PhD.
Katedra informatiky
Fakulta prírodných vied
Univerzity Konštantína Filozofa v Nitre
Tr. A. Hlinku 1, 949 74 Nitra

doc. Ing. Ivan Kotuliak, PhD.
Ústav počítačového inžinierstva a aplikovanej informatiky
Fakulta informatiky a informačných technológií
Slovenskej technickej univerzity v Bratislave
Ilkovičova 2, 842 16 Bratislava

Autoreferát bol rozoslaný dňa: .....

Obhajoba dizertačnej práce sa koná: 31.8.2016 o 13-tej hodine

na Fakulte elektrotechniky a informatiky STU, Ilkovičova 3, 812 19 Bratislava, v miestnosti C502.

prof. Dr. Ing. Miloš Oravec
Dekan FEI STU v Bratislave

Obsah

Úvod

Napredovanie techniky v poslednom čase naberá smer k socio-technickým systémom ako je internet vecí (IoT), rôzne sociálne siete, komunity a blogy. K socio-technickým systémom patrí nielen mobilný telefón, ale aj počítač, tablet, pomocou ktorých sa snažíme nepretržite komunikovať so svetom. Na jednej strane je snaha používateľa mať chránené súkromie a na druhej strane socializácia, teda zdieľanie informácií a pocitov s ostatnými, kedykoľvek a kdekoľvek. Práve tento protiklad alebo inak povedané nejasná hranica medzi nimi je to, čo sa vo veľkej miere útočníci snažia využiť. Práve útoky na mobilné zariadenia a webové útoky v poslednom čase narástli.

Kapacita batérie prenosných zariadení sa zvyšuje, avšak kapacitu batérie limitujú ako rozmery zariadenia, tak aj samotná jej hmotnosť. Energeticky efektívna klasifikácia je využiteľná práve v takýchto zariadeniach, ale aj systémoch s obmedzenými zdrojmi ako sú pamäť, výpočtový výkon a komunikácia.

Efektívnosť v rozhodnutiach sa dá vyjadriť **nákladmi** spojených s meraním, učením sa, a samotným rozhodnutím. Energetická efektívnosť s nízkym počtom falošných oznámení je zložitá úloha. Zložitejšou sa stáva, ak sa aplikuje na reálne namerané údaje. Kaskádová klasifikácia to do určitej miery vie znížiť falošné hlásenia. Vysoká detekcia a nízka chybovosť sú spriahnuté parametre a v ideálnom prípade je detekcia 100% a chybovosť na nule. No v reálnych podmienkach platí, že zvýšenie detekcie útokov má za následok zvýšenie chybovosti normálneho správania. Platí to aj opačne, zvýšenie detekcie normálneho správania zvyšuje ohrozenie úspešného útoku. Od kaskádovej klasifikácie sa očakáva síce zníženie detekcie, teda aby bolo čo najmenšie zníženie detekcie, ale zároveň veľkú redukciu chybovosti. Ide o všeobecnú ochranu zariadenia pre používateľa.

Podľa správy od Symantec [10], z roku 2009, bolo v roku 2008 1,6 milióna kódov označených za škodlivých. Čo pre Symantec znamenalo pre rok 2009, vytvorenie dvojnásobnej databázy ako za dosiaľ monitorované roky od 1992, odkedy začali vytvárať anti-vírusový softvér.

Zero-day vulnerabilities resp. po slovensky zraniteľnosti nultého dňa sú prioritou pri odhaľovaní škodlivých kódov už od 2009. Zraniteľnosti nultého dňa predstavujú nové, dosiaľ nepoznané útoky. Od roku 2006 do 2012 predstavovali len do 15 nových útokov za rok. V 2013 a 2014 zraniteľnosti nultého dňa vzrástli na počet 24 resp. 25 a v roku 2015 bol nárast až na 54 nových útokov [28]. Pri absencii záplat v operačnom systéme sú tieto kódy hrozbou, pretože sa často vyhnú detekcii založenej na čisto známych znakoch. Za rok 2015 uvádza viac ako 430 miliónov nových škodlivých kódov. A práve boj proti zraniteľnostiam nultého dňa považujú za kritické v ostatnom roku. Naopak počet softvérových robotov klesá od roku 2013. Veľmi lákavé pre kriminálnikov sú aj nové služby poskytované priamo výrobcami ako Apple Pay, Android Pay či Samsung Pay. Okrem priamo ohrozujúceho škodlivého kódu, Symantec poukazuje aj na nárast v oblasti obťažujúcich správ vo forme reklám v rôznych formách. Od úpravy fotoalbumu, cez kaledár až k zámene zvonenia za reklamu.

Webové servery sú podľa Symantec tiež v ohrození. Útoky na webové servery sa množia, vývojári operačných systémov síce záplaty poskytujú, avšak vlastníci serverov ich nepoužívajú a tieto záplaty neaktualizujú. To platí aj pre jednotlivé softvérové produkty, knižnice ako je napríklad rozosielanie automatizovanej pošty alebo SSH server (Secure Shell protocol). V posledných 3 rokoch viac ako tri štvrtiny webových stránok neboli aktualizované, pričom každá siedma mala kritickú zraniteľnosť. Zraniteľnosť nekončí iba na serveri, ale aj v rozšíreniach prehliadačov, najmä Adobe Plug-ins. Internet vecí patrí k rozvíjajúcej sa technológie a netreba podceňovať jeho bezpečnosť. K úniku informácie môže dôjsť napríklad z elektronického záznamu pacienta, kde je nutná zvýšená pozornosť na ochranu a bezpečnosť siete. Do internetu vecí patria aj domáce spotrebiče napojené na internetovú sieť, ako je napríklad chladnička s automatizovaným objednávaním chýbajúcej potraviny, či televízor, práčka, osvetlenie priestorov, monitorovanie energetickosti budovy.

Dostupnosť a **spoľahlivosť** sú dva najvýznamnejšie merané parametre služieb informačné systému. To však vyžaduje rýchlu a presnú detekciu narušení, prienikov, či anomálií. Pre tieto dva parametre je dôležité zotavenie sa, t.j. obnova celého softvéru. Tá sa deje následne po zistení anomálie, či útoku. Správne zotavenie má vplyv na správnu funkčnosť a teda i na dostupnosť a spolahlivosť informačného systému. Schopnosť rýchlo a správne reagovať na potreby systému, zotavenie, blokovanie komunikácie, vedie k potrebe hľadania správneho algoritmu.

Premenná *uptime*, čo je čas vyjadrujúci trvanie nepretržitej prevádzky a *downtime* trvanie bez služby, potom dostupnosť t_d je časová miera systému, aplikácie pracujúcej alebo schopnej pracovať a poskytovať služby, stiahnutá na obdobie, napríklad mesiac, rok a pod. Recipročne, nedostupnosť t_n je čas, kedy je server, aplikácia nedostupná.

$$t_d = \frac{\sum uptime}{\sum(uptime + downtime)} \quad , \qquad t_n = \frac{\sum downtime}{\sum(uptime + downtime)}$$

Spoľahlivosť sa môže definovať pomocou rôznych štatistických parametrov vyhodnotení klasifikátorov.

Algoritmy strojového učenia

K zautomatizovaniu dolovaniu znalostí sa využívajú algoritmy strojového učenia a atribúty záznamov. Algoritmy strojového učenia by sme mohli rozdeliť na klasifikáciu, predikciu alebo zhlukovú analýzu.

Všetky tri hľadajú značku pre údaj tvorený viacerými parametrami. Napríklad pri predpovedi počasia by to bola relatívna vlhkosť, priemerná teplota či smer a rýchlosť vetra. Cieľom klasifikácie na rozdiel od predikcie je nájsť diskrétne hodnoty značiek. Napríklad slnečno, dážď a pod. Pri predikcii sú používané funkcie pre nájdenie spojitej hodnoty. Napríklad predpokladaná teplota na zajtra $28,9\,^{\circ}\mathrm{C}$. Zhluková analýza v porovnaní s klasifikáciou nepozná výsledné značkovanie, ale zoskupuje blízke údaje do zhlukov alebo klastrov. Počet je daný implicitne alebo sa algoritmom sám snaží nájsť optimálny počet. Medzi hlavné vlastnosti,

vo väčšine implementačných metód, patrí efektívnosť a škálovateľnosť.

Všeobecne sa klasifikácia a predikcia nazýva ako kontrolované učenie a zhluková analýza ako nekontrolované učenie podľa toho, či množina obsahuje alebo neobsahuje značku. Najjednoduchším spôsobom klasifikácie je triedenie do dvoch skupín, ale počet skupín môže byť väčší, avšak známy a konečný pre proces klasifikácie [4].

Zo štatistických vyhodnocovaní klasifikátorov nad množinami boli použité nasledovné vzťahy.

Presnosť Acc je miera, podiel správne zaklasifikovaných záznamov zo všetkých.

Detekcia $DetR$ je podiel správne klasifikovanej hypotézy k všetkým skutočne pozitívnym.

Chybovosť FPr je miera nesprávne klasifikovanej opačnej hypotézy.

$$Acc = \frac{\sum_j c_{j,j}}{\sum_{i,j} c_{i,j}},$$

$$DetR_j = \frac{c_{j,j}}{\sum_i c_{i,j}},$$

$$FPr_j = \frac{\sum_k c_{j,k}}{\sum_{i,k} c_{k,i}}, \quad resp. \;\; FNr_j = 1 - DetR_j,$$

kde $i, j, k \in \{1, 2, ..., m\}$; $k \neq j$ sú indexi kategórii.

Detekcia je zároveň schopnosť správne zadeliť hľadaný záznam do triedy. To znamená, že pri hľadaní útokov $DetR$ vyjadruje mieru nájdenia triedy, teda pre prípad detekčného systému prienikov opísaný v práci, ako sa vedelo detegovať normálne správanie. Pre detekčný systém, ako antivírový softvér, chyba II. druhu má za následok obťažujúce hlásenia chýb, vo firewall-e ukončenie chceného spojenia. V praxi má chybovosť za následok napadnutie počítača alebo až pád systému.

Pre účely kaskádovej klasifikácie je použitý vzťah nezaklasifikovaných záznamov kategórie UN_j a celkovú mieru UNr

$$UNr = \frac{\sum_j UN_j}{\sum_{i,j} c_{i,j}} \quad .$$

Optimalizačné metódy strojového učenia

Ako bolo spomenuté náklady na rozhodovanie sú dôležitou súčasťou, no správne rozhodnutia tiež. K efektívnosti patrí vedieť nielen správne klasifikovať, ale aj rýchlo sa naučiť, či rýchlo klasifikovať, teda mať čo najjednoduchšie pravidlá. Samotné strojové učenia obsahujú heuristiky k efektívnejšej práci, ako napríklad zlučovanie pravidiel a vetiev rozhodovacieho stromu.

Optimalizačné metódy sú spojené buď s predspracovaním údajov pred alebo priamo v procese učenia. Poznáme veľa techník predspracovania údajov [13]. *Čistenie údajov*, na odstránenie šumu. *Spájanie údajov* z rôznych databáz. *Transformácia údajov*, ako je normalizácia. *Redukcia údajov*, sem patri zoskupovania, mazanie, zhlukovanie na zníženie počtu údajov. Predspraco-

vanie pomáha zefektívniť proces učenia zlepšením kvality atribútov a teda vzorov, ktoré majú byť učené.

Posledne menovaná technika *redukcia údajov* zmenšuje množinu údajov, pričom zachováva integritu pôvodnej.

Stratégie redukcie pozostávajú z:

- Agregácie záznamov
- Redukcia početnosti záznamov neparametrickými metódami
- Výber podmnožiny atribútov
- Redukcia dimenzionality pri zachovaní informácií záznamov
- Diskretizácia a vytváranie hierarchie

Ponechanie irelevantných, či redundantných atribútov môže viesť k degradácii učenia (viď nameraný výsledok z obr. 3, žltý graf, pre RBFN).

Dopredný výber atribútov (ďalej iba FS, z angl. Forward feature Selection) postupne pridáva jeden za druhým vybraný najlepší atribút. Výber nastáva podľa vyhodnocovacej podmienky, napríklad najlepšia presnosť z atribútov alebo najmenšia chybovosť. Terminácia pridávania atribútov nastáva tiež zvolením stratégie. Problém môže nastať, ak sa v inkrementačných krokoch nájde iba lokálne maximum. To sa rieši pridaním možnosti odskúšania ďalších krokov. FS je časovo náročný algoritmus pretože, v každom kroku prejde všetkými vyberanými atribútmi, ktoré vyhodnotí pomocou modelu a vyberie najúspešnejší. FS patrí aj medzi závislé spôsoby merania výberu vhodnosti atribútov. Nakoľko sa vždy vytvára model, ktorý sa testuje. Zároveň je možné, že vybrané atribúty nie sú najvhodnejšie. Ďalej v práci sa bude zapisovať použitie ako:

$$FS(MLA),$$

kde MLA bude jeden z klasifikačných modelov.

Spätný výber alebo inak eliminácia atribútov (ďalej iba BE, z ang. Backward feature Elimination) je opačný proces k FS, teda z celkového počtu atribútov sa vyhodnocovaním kritéria najhorší atribút odstráni. Ďalej v práci sa bude zapisovať použitie ako:

$$BE(MLA).$$

Korelačná váha prepočítava korelačné koeficienty medzi párom atribútov a tento vektor prepočíta na váhu v intervale [0;1], kde 1 znamená najväčšia váha pre výber atribútu. **mRMR** je metóda od Peng a kol. [21] pre optimálny výber atribútov. Štatisticky prepočítava maximálnu závislosť počítaného atribútu na parameter značky. Ďalším výpočtom je maximálna relevancia počítanou vzájomnou výmenou informácie a minimálnej redundancie. Všetky tri parametre sa skombinujú pre dosiahnutie usporiadaného listu.

"Neexistuje žiadny filter, či obálkovač na výber optimálnych atribútov, nezávislý či od učenia či na metrike výkonnosti modelu.", konštatoval Aliferis [3]. Kedže sa nedokáže nájsť univerzálne optimálny, nedá sa špecifikovať, ktorý by bol výhodnejší.

1 Kaskádová klasifikácia

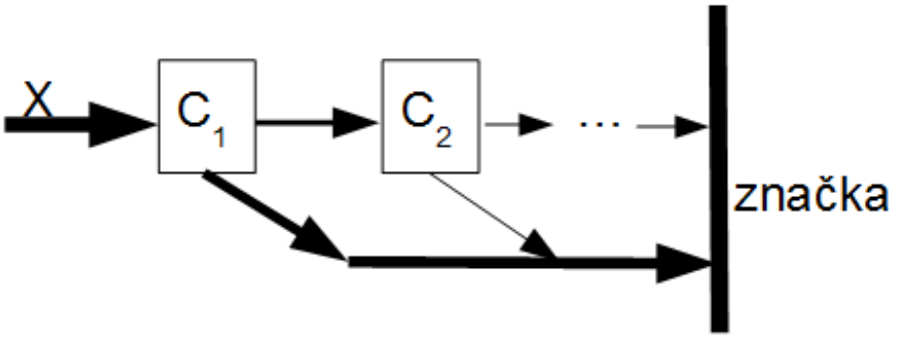
Cieľom je využiť túto metodológiu ako energeticky efektívny mechanizmus pri zachovaní prípadne zlepšení presnosti klasifikácie alebo zníženia chybovosti, hlavne II. druhu t.j. zbytočné obťažovanie detekčným systémom. Senzorové siete sú špeciálny prípad, ktoré majú obmedzené zdroje a teda nie je vhodné neefektívne využívanie batérie. Zložité algoritmy strojového učenia vyžadujú vysoké nároky na výpočet (pamäť, procesor). Posielanie informácii na server, anomálie administrátorovi, z ktorého sa vráti použiteľný model, má tiež vysoké nároky na zdroje. Vysielanie a prijímanie spotrebuváva rádovo viac energie ako použitie procesora, či meranie zo senzorov. Z tohto dôvodu je nutný kompromis medzi výpočtom a vysielaní v záujme efetívnejšieho využitia zdrojov [8].

Kaskádové klasifikátory boli študované Gama-om a Brazdil-om v článku [11]. Píšu o využití sekvenčných kaskád pre zvýšenie presnosti *Acc* vyhodnotenia. Porovnávali pritom tri algoritmy strojového učenia, vo všetkých kombináciach a na rôznych množinách údajov. Lineárny diskriminant, rozhodovací strom a naívne Bayesov algoritmus. V každej iterácii sa vytvoril model zo základného klasifikátora. Následne pridaním nových atribútov sa vytvárali nové pravidlá.

Kaskádova viacstupňová klasifikácia, opísaná v Kaynak a Alpaydin [15], definuje princíp sekvenčných stupňov (viď obr. 1) tak, aby *"... v ďalšom stupni, použijeme nákladnejší klasifikátor, zostavíme zložitejšie pravidlo na pokrytie ešte nepokrytých vzorov z predchádzajúceho stupňa"*, teda nasledovali vzostupne podľa nákladovej náročnosti a zložitosti algoritmov strojového učenia.

Tento krok je zdôvodniteľný pri veľkých počtoch záznamov, kde aj jednoduchším algoritmom môže byť záznam správne zatriedený. Kaskádová metodológia vo formálnom výraze je reprezentovaná jednotlivými stupňami strojového učenia označených ako C_1, C_2 ... Ako prvý sa aplikuje stupeň s najnižším indexom (1), aby kategorizoval záznamy. V tomto prípade C_1 zastupuje najbližší sused získaný pomocou algoritmu E2LSH, viď kapitolu 2. Záznamy, ktoré neboli zatriedené v C_1, sú poslané do ďalšieho stupňa C_2. V prezentácii výsledkov nižšie sa použili iba dva stupne, no tento prístup sa dá zovšeobecniť na väčší počet algoritmov. V príkladoch sa používa notácia

$$C_1 \odot C_2 = C_2(C_1(X))$$



Obrázok 1: Princíp kaskádovej klasifikácie, s klasifikátormi C_1 a C_2 prideľujúcimi značku.

.

Pri kaskádovej klasifikácii sa uplatňuje pravidlo — najprv triedi najlacnejší algoritmus. Ak tento nevie s určitosťou zatriediť záznam, ten je postúpený do ďalších stupňov, kde sa dostávame k zložitejším algoritmom, ale aj k presnejším výsledkom. Prvý klasifikátor má byť nastavený tak, aby majoritné správanie bolo čo najskôr označené. Ako je naznačené hrúbkou šípiek na obrázku 1.

Použitie prvého stupňa E2LSH bolo logickou voľbou nakoľko dokáže rýchlo nájsť najbližších susedov, podľa ktorých sa pridelí značka záznamu [5]. Výhodou je aj pripravenosť algoritmu na jednoduchú aktualizáciu, pridanie záznamu do trénovacej množiny. Ďalšou výhodou je práca na veľkom súbore záznamov. Kedže väčšinové monitorovanie v sieti je normálne, korektné správanie, jej identifikácia by mala byť najefetívnejšia. Preto aktualizácia prvého stupňa by mala byť vyvážená a mať skôr za následok považovať všetko za korektné správanie. Podružnou výskumnou úlohou v práci je nájdenie optimálnych parametrov E2LSH, ktoré spočívajú práve v (ne)nájdení najbližších susedov, v rámci postačujúcej efektivity. Nájdenie rovnováhy vo výpočtovej zložitosti zmenou parametrov polomeru R a pravdepodobnosťou P. Ostatné parametre sú dopočítateľné. Tretím parametrom testovaný v tejto práci je počet k a zhodnosť najbližších susedov. V tomto prípade sa uvažuje o týchto situáciach:

(i) hľadanie k=1 susedov,

(ii) hľadanie k=2 najbližších susedov.

V oboch situáciach, ak je vzdialenosť l=0, teda bol nájdený totožný záznam, klasifikuje sa priamo podľa neho. Ak pri k=2, najbližší susedia nemajú rovnaké značky, sú označené ako nezaklasifikovaný záznam. V rámci práce sa venovalo aj hľadaniu algoritmu pre ďalšie stupne v kaskádovej klasifikácie. Ich porovnávanie je popísane nižšie. Pri testovaní a vyhodnocovaní sa použili programy E2LSH od Andoniho [5], nástroj Rapidminer [23] s rozšíreniami hlavne Weka.

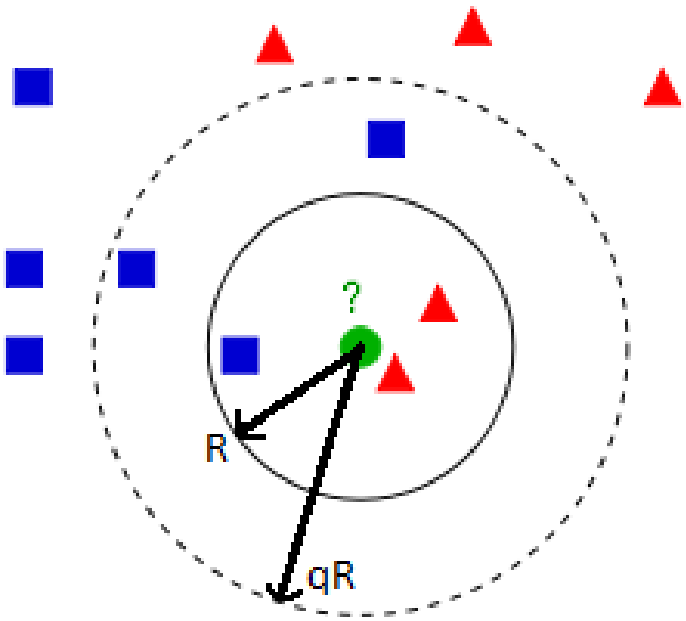
2 Hašovanie na základe blízkosti v Euklidovom priestore

Metóda hľadania najbližšieho suseda patrí medzi tzv. "lenivé učenie", pretože model je hľadaný až pri dotazovaní sa na kategorizovaný záznam. Je to protiklad, kde učenie prebieha skôr a model je už známy pri dotazovaní sa na záznam, teda proces učenia odpadá z nákladov klasifikácie. Výhodou takéhoto prístupu je možnosť súčasného riešenia viacerých problémov a úspešné aktualizácie problémov. Nevýhodou je veľká pamäťová náročnosť, veľký vplyv šumu na rozhodovanie a obyčajne sú tieto metódy označované za pomalšie.

Pre metódu hľadania suseda platí, že k dotazovanej vzorke sa vypočítajú najbližší susedia obr.2 na základe vzdialenosti. Vyhodnotenie kategórie má rôzne parametre ako metrika v priestore (Hammingová, Euklidová, Manhattanská), počet vyšetrovaných najbližších susedov k i samotné vyšetrovanie majoritou, minoritou alebo inou prioritou funkciou. Táto metóda je dobre rozpracovaná a má veľa heuristík na zlepšenie výkonnosti, ako napríklad redukcia dimenzionality.

Hašovanie na základe blízkosti bolo predstavené Indykom a kol. [14]. Uplatnenie hľadali v rýchlosti nájdenia najbližšieho suseda meraním počtu prístupov do pamäte. Ich cieľom bolo nájsť v dosahu o polomere R a s pravdepodobnosťou P susedov pre bod záujmu $\widehat{X}$, t.j. záznam. Princíp spočival v tom, že hašovacia funkcia blízkym bodom, s polomerom menším ako $l(X, \widehat{X}) < R$, priraďovala rovnakú hodnotu, zatiaľ čo ďalekým bodom rozdielnu. Ďalekým bodom definovali ako vzdialenosť $l(X, \widehat{X}) > q \cdot R$, pričom platí, že $q > 1$ je konštanta, ktorá kvantifikuje medzeru medzi blízkosťou a ďalekosťou viď obr.2.

Množina hašovacích funkcii $\mathcal{H} = \{h : \mathbb{R}^d \to \mathbb{N}\}$ mapuje d-dimenzionálny vektor záznamu



Obrázok 2: Najbližší susedia. Dotazovaný bod zelený, ostatné farby predstavujú kategórie v trénovacej množine. (zdroj wikipedia)

X na prirodzené čísla. $\mathcal{H}$ je nazvaná $(R, qR, P1, P2)$-citlivá, ak pre ľubovoľné dva body $X, \widehat{X}$ platí:

- ak $l(X,) \leq R$ potom $Pr[h(\widehat{X}) = h(X)] \geq P_1$,
- ak $l(X, \widehat{X}) \geq qR$ potom $Pr[h(\widehat{X}) = h(X)] \leq P_2$.

Nutnou podmienkou pre hašovaciu funkciu je zabezpečiť, aby pravdepodobnosť $P_1 > P_2$.

Algoritmus E2LSH (Exact Euclidean Locality Sensitive Hashing) je navrhnutý, aby redukoval dimenziu priestoru pôvodného záznamu X vytvorením množiny funkcii $\mathcal{G} = \{g : \mathbb{R}^d \to \mathbb{N}^K\}$, pozostávajúcich z hašovacích funkcii $\mathcal{H}$, $g(X) = (h_1(X), ..., h_K(X))$. Pre zadané parametre R, P sa vypočíta parameter L, počet rovnomerne a nezávislo náhodne vybraných funkcii $g_1, ..., g_L$ z $\mathcal{G}$. Pri predspracovaní trénovacej množiny sa uloží každé X do balíkov. Spracovanie dotazu $\widehat{X}$, znamená vypočítať jednotlivé $g_i(\widehat{X})$ a ak platí pre aspoň jedno $i = \{1, ..., L\}$ také, že $g_i(X) = g_i(\widehat{X})$, potom z balíkov $g_i(X)$ sa prechádza bodmi X a vrátia sa tie, pre ktoré platí $l(X, \widehat{X}) \leq R$. Zároveň platí, že počet počet vrátených bodov pre $\widehat{X}$, kde $l(X, \widehat{X}) \leq qR$, je menšia ako $3L$.

Datar a kol. [7] navrhli hašovaciu funkciu

$$h_{a,b}(X) = \frac{a.X + b}{w},$$

kde a je vektor náhodne vybraný z Cauchy alebo Gaussovho rozdelenia, $b \in \mathbb{R}$ je náhodne číslo z rovnomerného rozdelenia a z intervalu $[0, w]$. $(a.X)$ predstavuje skalárny súčin a w je číslo zobrazenia, ktoré rozdeľuje intervaly na rovnaké segmenty.

Implementáciu E2LSH podľa článku [5] naprogramovali Andoni a Indyk [6]. Citát z krátkej odpovedi na otazku, čo je E2LSH:*"...balík umožňujúci na mnohodimenzionálnom Euklidovom l^2 priestore nájsť najbližšieho suseda. Po predspracovaní údajov, E2LSH odpovedá na dotazy typicky v podlineárnom čase. Každý sused je ohlásený s určitou pravdepodobnosťou..."*. Táto metóda na rozdiel od k-najbližšieho suseda, nehľadá počet k susedov, ale všetkých susedov v polomere R s pravdepodobnosťou P, zapísuje ako (R, P)-najbližšieho suseda, čo je len intuitívnejší zápis (R, q)-najbližšieho suseda popisovaného vyššie. Zároveň by sa dalo povedať, že transformuje dimenzionalitu
$dim(X)=d \ \rightarrow \ dim(X)=L$.

Program vyžaduje tri základné parametre K, L, M, tie sú však vypočítane optimalizáciou na odhad času odpovede, založenej na počte kolízii buď z náhodne vytvorenéj množiny, alebo množiny určenej používateľom. Parametre optimalizácie sú polomer, pravdepodobnosť, dostupná pamäť z dôvodu možného stránkovania a počet záznamov trénovacej množiny s ich dimenziou. Odhad sa začína parametrom K, pričom sa minimalizuje suma časov

(i) potrebných na výpočet L funkcii pre bod $\widehat{X}$ a

(ii) na výpočet vzdialeností vrátených z hašovacích funkcii.

Intuitívne tieto časy sú k parametru K recipročné. Kým prvý čas stúpa druhý klesá.

3 Vyhodnotenie kaskádovej klasifikácie

Káskádová klasifikácia bola aplikovaná na dve hlavné množiny KDD99 a DAP14. Množiny tvorili reálne údaje z útokov na sieť a používania mobilných zariadení.

KDD99, KDD CUP 1999 (Knowledge Discovery and Data Mining 1999) bola súťažou pre detekčné systémy prienikov. Oficiálnou úlohou súťaže [24] bolo nájsť predpovedný model schopný odlíšiť oprávnené a neoprávnené spojenie v počítačovej sieti. Inak povedané, vytvoriť detekčný systém rozpoznávajúci prieniky na:

- 'dobré' a 'zlé' spojenia, t.j. normálnu prevádzku siete a útok na sieť;
- normálnu prevádzku a 4 skupiny útokov;
- normálnu prevádzku a konkrétny typ útoku.

DAP14, Device Analyzer Project 2014 [26, 27] je časová-séria záznamov s skladajúca sa z približne 300 rôznych udalostí. Záznamy boli tvorené na mobilných zariadeniach s operačným systémom Android vo verzii 2.1 a vyšších.

Ďalšími porovnávacími množinami boli množina Yahoo a množina Madelon. Yahoo! WEBSPAM-UK2007 [2], je množina z prehľadávania .uk domén z mája 2007. Obsahuje 114529 stránok, z ktorých 6479 je označených buď ako spam, nespam alebo nerozhodnutý.

Madelon [1] je umelá množina určená pre výzvu výberu atribútov na NIPS 2003. Množina je zoskupená do 32 zhlukov v päťsto dimenzionálnom priestore. Množina obsahuje 2600 označených záznamov.

3.1 KDD99

Množina údajov pre učenie pozostávala z 494 021 záznamov a testovacia množina z 311 029. Trénovacia množina bola upravená množina z pôvodnej päť miliónovej množiny.

Lincolnove laboratória po dobu 9 týždňov zbierali záznamy o komunikácii z miestnej siete, pričom simulovali typickú komunikáciu amerických vzdušných síl. Do sieti vkladali rôzne druhy útokov. Trénovacia množina meraná počas 7 týždňov bola spracovaná do približne 5 miliónov záznamov o spojení. Podobne bola testovacia množina meraná počas 2 týždňov a spracovaná do 2 miliónov záznamov. Každý záznam, definovaný komunikáciou dvoch uzlov, bol označený buď **normal** alebo tým konkrétnym špecifickým útokom nižšie. Záznam bol tvorený 41 atribútmi, z čoho 34 je numerických, spojitých hodnôt a 7 symbolických, nominálnych hodnôt.

Pre objasnenie zastúpenia jednotlivých vzoriek, ktorá sa dá stiahnuť, sa vykonala analýza, ktorej výsledky sú uvedené v tabuľke 1. **Plná** trénovacia množina obsahovala skoro 5 miliónov záznamov, z ktorej autori vytvorili **10%-nú** čiastkovú trénovaciu množinu s počtom záznamov necelých 500 tísíc. **Testovacia** množina obsahovala viac ako 300 tisíc označených dotazov. V testovacej množine okrem známych útokov boli aj nové typy útokov, v počte 17, zadelené do pôvodných 4 skupín. Dva známe typy sa už nevyskytovali. Záznamy z testovacej množiny nemajú rovnakú distribúciu rozdelenia ako trénovacia množina.

McHugh v článku [19] sumarizoval kritiku z KDD99. Odporúčal vhodnejšie merania výkon-

nosti, lepšiu charakterizáciu a validáciu prevádzky, rozšírenie experimentu ku komerčným systémom, zriadenie normatívnej databázy útokov pre podporu výskumu do budúcnosti.

Tavallaee a kol. vykonali štatistickú analýzu dát zo súbora KDD99 [25]. Našli sa dva hlavné nedostatky. Veľké množstvo redundantných údajov, 78% v trénovacej a 75% v testovacej množine. Druhý nedostatok vyšiel z analýzy zložitosti zatriedenia údajov, teda možnosť odlíšenia medzi normálnym správaním a útokom. Zistili len malé rozdiely medzi jednoduchými a zložitými algoritmami. Skúmali nasledovné algoritmy: rozhodovaci strom J48, Naive Bayes, Random Forest, Random Tree, Multi-layer Perceptron, Support Vector Machine), čo uvádza i Elkan [9]. Vytvorili novú množinu údajov nazvanú NSL-KDD odvodenú z pôvodnej verzie KDD99, i keď i v tejto množine zostávajú niektoré problémy opísané u McHugha [19].

3.2 Výsledky KDD99

Nakoľko použitie E2LSH vyžaduje striktne numerické atribúty, tie nie numerické atribúty boli transformované. Ako prvé sme odskúšali premapovávať texty na monotónne sa zvyšujúce celé čísla. Následne boli atribúty normalizované na interval $[0;1]$. Z analýzy vyplynulo, že čím viac údajov je k dispozícii, tým je E2LSH výkonnejšie. Zväčšením polomeru výrazne znížime nerozhodnuteľnosť UNr. Parameter P od hodnoty 0,9 na výsledky nemal, pričom sa predpokladalo použitie na rozsiahlych údajov, čo minimalizuje vplyv tohto parametra.

Dôsledkom bolo fixné zmeny polomeru $R = 0,1$ a nastavenie kolízneho parametra $P = 0,9$, čo predpokladá 10% pravdepodobnosť, že sa nenájde sused a úpravu transformácie nominálnych hodnôt [18] nasledovne.

k-simplex metóda mapuje nominálne hodnoty atribútu do bodu s dimenzionalitou rovnou práve počtu unikátnych nominálnych hodnôt. Následne tento podpriestor je normalizovaný na medzi bodovú vzdialenosť $[0;1]$ *v záujme zachovania integrity metriky.*

Týmto predspracovaním narástla dimenzionalita z pôvodných 41 na 116 resp. 121 v prípade spracovania celej množiny KDD99, ako trénovacej tak aj testovacej.

Následne boli pôvodné trénovacie množiny podrobené analýze, kde sa zistila redundancia záznamov. Pozorované rozdelenie je zobrazené v tabuľke 1. Z pôvodných 5 miliónov záznamov sa trénovacia množina zmenšila 5-násobne (označenie **Rx**).

Tabuľka 1: Percentuálne zastúpenie skupín v množine KDD99 .

	plný ($\mathbf{p_x}$) $\rightarrow$ RD ($\mathbf{R_x}$)		čiastkový ($\mathbf{p_1}$) $\rightarrow$ RD ($\mathbf{R_1}$)		test $\rightarrow$ RD ($\mathbf{R_D}$)	
normal	19,859	75,612	19,691	60,331	19,481	62,005
sondáž	0,839	1,288	0,831	1,463	1,339	3,467
DOS	79,278	23,002	79,239	37,484	73,901	30,522
U2R	0,001	0,005	0,011	0,036	0,073	0,278
R2L	0,023	0,093	0,228	0,686	5,205	3,727
počet vzoriek	4 898 431	1 074 974	494 021	145 584	311 029	77 216

Na zlepšenie presnosti klasifikácie sa aplikoval dopredný i spätný výber atribútov (FS, BE) z **Rx** trénovacej množiny. Výber sa uskutočnil v krížovej validácií vo vybraných algoritmoch uvedených v tabuľke 2.

V spomenutej tabuľke sú uvedené výsledky použitia algoritmov ako víťaza súťaže KDD99 [22], 9. miesto najbližšieho suseda [9], či nedávno publikovanej doktorandskej práce [16]. Poradie sa určovalo koeficientom ceny ξ danej vzťahom:

$$\xi = \frac{\sum_{i,j} c_{i,j} b_{i,j}}{\sum_{i,j} c_{i,j}}$$

Celkovo vzaté, výsledky so základom aproximáčného prístupu neboli významne lepšie ako víťaz KDD99 (na základe použitia rozhodovacích stromov C5), no zníženie výpočtovej zložitosti v tomto prístupe je vďaka, najmä aplikovaniu hašovania na základe blízkosti. **Významnou úlohou** je práve správna detekcia majoritných vzoriek, z reálneho sveta označovaných ako normal a DOS útokov z hľadiska množiny KDD99.

3.3 DAP14

DAP14 je množina súborových záznamov v pološtrukturovanej forme činností na mobilnom zariadení. Na zariadeniach si používateľ nainštaloval aplikáciu Device Analyzer, ktorá zaznamenáva stavy v čase alebo pri udalostiach do svojej internej databázy a následne ich poslal na server.

Následne uverejnili všeobecnú výzvu na dolovanie informácii uverejnenú v roku 2014. Účastníci si mohli stiahnuť najprv verejne dostupnú malú vzorku, a následne zvyšnú množinu cez aplikáciu podľa preferencii a potrieb z analýzy prvej vzorky. Analýza DAP14 Wagnera a kol. bola zameraná hlavne na pripojiteľnosť, presun používateľov, správu batérie, volania a správy. Zo záverov vyplynulo, že v priemere sa európan presúva menej ako ind či američan. Každý 6. používateľ nabíja telefón raz denne. Vybitie batérie sa stávalo v priemere do 11.-teho dňa u polovice používateľov. Nabíjanie sa vo väčšine udialo do jednej hodiny $t_d = 92.4\%$.

V stiahnutej množine sa vyskytovalo približne 1TB údajov v komprimovanej forme z približne 17000 mobilnych telefonov. Údaje boli zbierané od 25.9.2010 do 1.9.2014. Stiahnutých bolo 16830 súborov. Záznamy boli vo formáte CSV. Riadok predstavuje záznam s ´;´ oddelenými hodnotami. Išlo o nasledovné údaje:

- identifikátor riadka ID, kontrola z dôvodu rozdeľovania súbora;
- *uptime* čas od posledného zapnutia;
- predpokladaný čas s dátumom;
- kľúč k atribútu, vo formáte hierarchickej štruktúry oddelenej čiarou '|'.
- hodnota pre daný atribút

3.4 Výsledky DAP14

Cieľom bolo detekovať abnormálne správanie sa mobilného zariadenia. Abnormálne správanie je definované používateľom alebo operačným systémom, teda obsluhujúcim programom. Používateľ zistí chybu, keď sa aplikácia nespráva tak, ako očakával. Prestane pracovať, odosiela

Tabuľka 2: Porovnanie výkonností metód použitých na množine KDD99. *Pozn.: skr. DT — rozhodovcacie stromy;*

	DetR [%]					FPr [%]	Acc [%]	ξ	Poznámka
	normal	sondáž	DOS	U2R	R2L	normal			
1.miesto KDD99, C5 DT	99,45	83,32	97,12	13,16	8,40	8,19	92,71($\pm$0,825)	0,2331	B. Pfahringer [22]
9.miesto KDD99, najbližší sused	99,55	75,01	97,29	3,51	0,59	9,12	92,33($\pm$0,815)	0,2523	Ch. Elkan [9]
HNB_PKI_INT	98,04	61,28	99,60	2,63	3,69	6,28	93,72	0,2224	Koc a spol. [16]
XMeans^{R1}	56,49	59,19	97,33	0	11,89	0,566	84,34	0,2571	
XMeans^{Rx}	55,75	58,33	97,15	0	12,44	2,44	84,09	0,2776	
Weka J48^{R1}	99,49	74,70	97,31	3,95	5,84	0,51	92,60	0,2401	
FS(WAODE)^{Rx}	99,36	58,43	97,00	0	0,02	0,64	91,83	0,2648	
FS(HNB)^{Rx}	99,34	60,71	96,97	0	0	0,66	91,83	0,2639	
FS(NaB)^{Rx}	99,14	55,88	96,58	0,00	4,16	9,52	99,14	0,2634	
BE(NaB)^{Rx}	97,45	63,27	96,92	14,04	0,06	9,37	91,47	0,2719	
FS(RBFN)^{Rx}	99,32	64,09	97,00	0,00	0,02	9,35	99,32	0,2637	
E2LSH^{p1} *R=0,1* $\odot$ Weka J48	99,48	77,03	97,15	3,95	6,31	9,01	92,54	0,2433	
E2LSH-2^{p1} *R=0,1* $\odot$ Weka J48	99,51	76,98	97,16	3,51	6,14	9,02	92,54	0,2434	
E2LSH^{R1} *R=0,1* $\odot$ XMeans^{R1}	97,17	68,89	97,27	0,00	10,38	4,76	92,28	0,2077	
E2LSH-2^{R1} *R=0,1* $\odot$ XMeans^{R1}	97,15	68,72	97,27	0,00	10,40	4,76	92,27	0,2079	
E2LSH^{R1} *R=0,1* $\odot$ FS(WAODE)^{R1}	99,76	65,43	97,34	0,00	3,29	8,87	92,41	0,2500	z 2 atribútov
E2LSH-2^{R1} *R=0,1* $\odot$ FS(WAODE)^{R1}	99,77	64,95	97,33	0,00	0,02	9,08	92,24	0,2570	z 2 atribútov
E2LSH^{R1} *R=0,1* $\odot$ FS(HBN)^{R1}	99,74	65,96	97,33	0,00	3,27	8,89	92,41	0,2493	z 4 atribútov
E2LSH-2^{R1} *R=0,1* $\odot$ FS(HBN)^{R1}	99,75	65,72	97,32	0,00	0,00	9,11	92,24	0,2562	z 4 atribútov
E2LSH^{R1} *R=0,1* $\odot$ FS(NaB)^{R1}	96,65	81,57	96,58	38,16	3,35	6,3	91,5	0,2309	z 7 atribútov
E2LSH-2^{R1} *R=0,1* $\odot$ FS(NaB)^{R1}	96,63	81,59	96,54	38,60	0,09	6,3	91,29	0,2348	z 7 atribútov
E2LSH^{R1} *R=0,1* $\odot$ BE(NaB)^{R1}	98,21	68,67	97,30	14,04	3,33	8,82	92,14	0,2554	z 12 atribútov
E2LSH-2^{R1} *R=0,1* $\odot$ BE(NaB)^{R1}	98,23	68,24	97,29	14,04	0,06	9,04	91,96	0,2624	z 12 atribútov
E2LSH^{R1} *R=0,1* $\odot$ FS(RBFN3)^{R1}	99,72	71,10	97,34	0,00	3,29	8,83	92,48	0,2488	z 3 atribútov
E2LSH-2^{R1} *R=0,1* $\odot$ FS(RBFN3)^{R1}	99,73	70,62	97,33	0,00	0,02	9,04	92,31	0,2558	z 3 atribútov

údaje bez vyzvania, no najobťažujúcom správaním považujúcim za abnormálne je "zmrznutie" systému.

Počas analýzy údajov nebolo známe, ktoré z predošlých kľúčov súviseli z ich vypnutím. Niektoré parametre z kľúčov neboli zaradené do hodnotenia, no mohli mať zásadný vplyv pri získavaní znalostí.

Parametre na predikciu boli inšpirované z článkov pamäť, procesor [12], batéria, podsvietenie [20, 27, 17]. V množine DAP14 je možstvo zaznamenaných hodnôt, ktoré medzi sebou súvisia alebo sú celkom nezávislé.

V obvyklom správani sa hľadalo včasné upozornenia, ktoré by indikoval vypnutie systému. Každá aplikácia spotrebováva energiu batérie rôzne, a teda upozornenia na slabú výdrž batérie by mal predchádzať adaptívnej stratégie využívania týchto zdrojov. Za korektné správanie možno považovať, keď sa zariadenie používalo a na noc sa vyplo, ale aj to, že po nainštalovaní aplikácie bolo nutné zariadenie reštartovať. Za nekorektné správanie sa považuje, keď sa zariadenie vyplo vyčerpaním batérie, vynutením hardvérového vypnutia (vo väčšine zariadení je to dlhé stlačenie tlačidla zapnúť/vypnúť), či vybratím batérie s následkom vypnutia zariadenia. Za nekorektným správaním sa hľadali rôzne sledy udalosti, ktoré ich mohli spôsobiť; chybná aplikácia, vírus v zariadení, či iné nesprávne používanie zariadenia. Skoro každé zariadenie má už vstavané upozornenia na nízky stav batérie, a teda pri dalšom používaní sa systém vypne sám, čo patrí tiež k správnemu používaniu.

Kľúč *crash* ako značka nebol zastúpený, a teda *zlyhanie* bolo určené podľa toho, či sa nachádza alebo nenachádza vypínacia sekvencia *shutdown* medzi dvoma zavedeniami systému, t.j. *startup*, alebo *time|bootup*.

Atribúty boli získané zachyteným posledným stavom vybraných parametrov pred vypnutím resp. novým zavedením.

Ďalej sú uvedené niektoré fakty o spracovaní údajov DAP14. Pravidelný zber údajov bol pozorovaný u 60% používateľov, pričom 4% z nich využili možnosť zastavenia zberu údajov, čo preruší kontinuálnosť merania a následne i predpoklad zlyhania systému.

Zdravie batérie patrí k dôležitým atribútom mobilného zariadenia, no v DAP14 sa zriedka vyskytujú iné hodnoty ako: 'neznámy' a 'dobrý' stav; 'neznámy' stav nie je problém, vyskytuje sa aj u iných atribútov. 'nefunkčný' stav mal za následok zlyhanie, ale bol to len jeden záznam, tak ako aj stav 'studenej' batérie. Preťaženie či prehriatie so sporadickým výskytom neznamenalo zlyhanie ani vo väčšine prípadov.

Z tabuľky 3 E2LSH s R=0,1 vyplýva, že sa nezaklasifikuje až 66% pre prvého najbližšieho suseda a resp. 75% pre prvých dvoch. Zvýšením polomeru na R=0,3 sa zníži nezaklasifikovanie na 13% resp. 32%. Ďalším zväčšením na R=0,9 klesne pod 1% resp. na 21%. Priemer R=2,5 bol vybratý, aby sa našiel aspoň jeden sused, a zároveň slúži ako hranica preporovnanie dvoch najbližších susedov na nezaklasifikovanie. Výsledkom je 20% záznamov, ktoré putovali do druhej kaskády. Zároveň možno pozorovať presnosť 80% klasifikácie aproximatívnym najbl. susedom.

Filtráciou atribútov sa zlepšili výkonnosti všetkých algoritmov. V prípade E2LSH nie je

Tabuľka 3: Vyhodnotenie výkonnosti na množine DAP14 s E2LSH pre rôzne hodnoty parametra R

	Acc[%]	$DetR$[%]	TP	FP	FN	TN	UN_T	UN_F
E2LSH R=0,1	85,70±0,79	88,11±1,04	6834	920	926	4239	15655	9888
E2LSH R=0,3	81,32±0,55	84,72±0,55	17538	3165	3094	9699	2783	2183
E2LSH R=0,9	80,29±0,64	83,69±0,59	19575	3817	3707	11082	133	148
E2LSH R=2,5	80,24±0,65	83,59±0,58	19675	3862	3740	11185	0	0
E2LSH-2 R=0,1	90,06±0,66	91,84±1,10	4726	425	514	3785	18175	10837
E2LSH-2 R=0,3	87,88±0,57	89,36±0,57	14610	1742	1414	8267	7391	5038
E2LSH-2 R=0,9	87,35±0,65	88,50±0,70	17202	2236	1596	9252	4617	3559
E2LSH-2 R=2,5	87,30±0,62	88,39±0,64	17347	2279	1601	9321	4467	3447

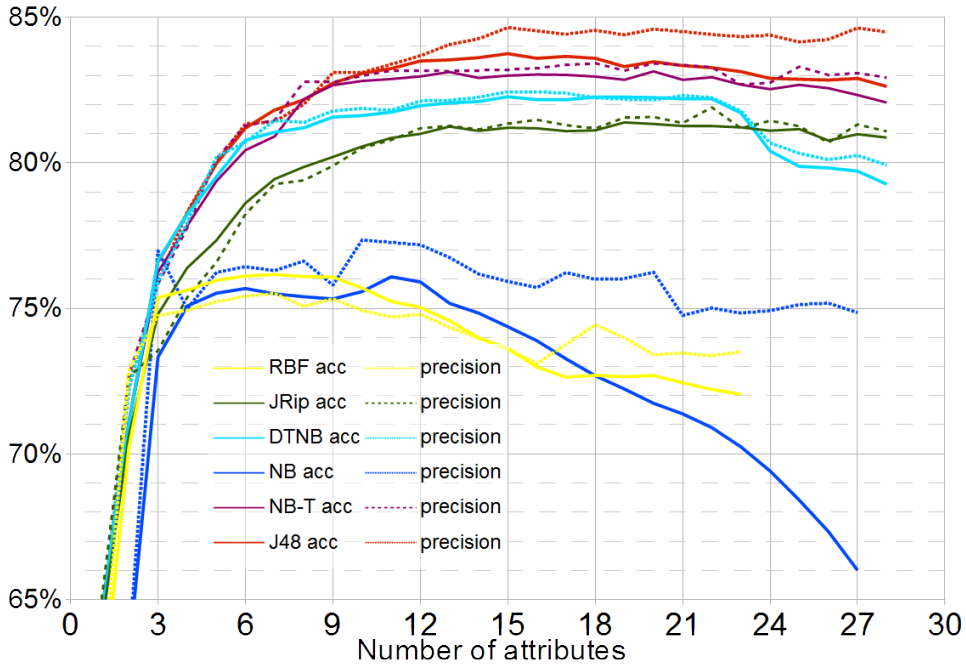
žiadúce vyberať vhodné atribúty, nakoľko spracovanie má byť čo najjednoduchšie a zmena v jednom atribúte pri normalizácii sa neprejaví natoľko, aby vážne ovplyvnil výsledok. Na obrázku 3 sú znázornené ako postupným pridávaním kulminuje presnosť a detegovateľnosť pre spomínaných 6 algoritmov. Na obrázku je badať ako niektoré algoritmy s pribúdajúcim počtom atribútov degradujú výkonnosť (NaB, RBFN, a neskôr aj DTNB).

Nakoľko samotné FS alebo BE trvá dlho, hlavne pri rozhodovacích stromoch a pravidlách, boli odskúšané aj iné postupy nájdenia optimálnych atribútov ako výpočet korelačných váh alebo nájdenia minimálnych relevantných atribútov. V týchto prípadoch je výsledkom zoradenie atribútov. Nakoľko FS ukázalo, že pri 12 atribútoch už žiaden algoritmus výrazne na presnosti nestúpa, rozhodlo sa ich viacej nevyberať. Ďalším kritériom bolo aj samotné vyhodnotenie poradia s hodnotou váhy alebo skóre. Bližšou analýzou výsledných hodnôt korelačnej funkcie medzi 8.,9. a 10. atribútom boli veľmi rozdielne. Zároveň pre postup mRMR sa naopak hodnoty po 10. atribúte vyrovnali. Preto bol výsledný počet výberu atribútov z týchto postupov nastavený na 9.

Porovnaním jednotlivých optimalizačných postupov nie je jednoznačne dokázané, ktorý je lepší. Pre RBFN je výpočet korelácie lepšie ako skóre z mRMR, naopak u J48. Porovnávať tieto postupy nezávislé od algoritmov nie je vhodné, nakoľko trvanie výpočtu je kratšie ako opakované trénovanie na algoritmoch.

Z výsledku analýzy FS je možné vyčítať tie najrelevantnejšie atribúty ako *O_display, GSM_srvS, B_lvl*, *C_num, S_eMem* je tiež v každej FS(MLA) okrem jednej. Na druhej strane sa s atribútmi *A_sMin, A_priv, A_num* nepočíta. Využitie CPU nemá dobrú reprezentáciu *O_cpu* atribútom v množine a je nutné bližšie preskúmanie.

Tabuľka 4 ukazuje spoločné výsledky kaskádovej klasifikácie. Najlepší výsledok je dosiahnutý s E2LSH-2 $\odot$ Weka J48 s presnosťou 83.78% a chybovosťou 24.57%. Pre porovnanie sú v treťom a štvrtom stĺpci výsledky aplikácie výberu atribútov s mRMR a korelačnou funkciou. V porovnaní s najlepším dosiahnutým výsledkom, Weka J48, s presnosťou 83.75% a chybovosťou 25.28% badať porovnateľné výsledky.



Obrázok 3: Vyjadrenie detekcie a presnosti pre jednotlivý iteračný krok FS(MLA)

Tabuľka 4: Hodnoty presnosti $Acc[\%]$ a použití kaskádovej klasifikácie na množine DAP14 s 1. stupňom E2LSH a rôznymi MLA v 2. stupni.

MLA		pre všetky atribúty	FS	mRMR	korel. váhy
E2LSH R=0,1	RBFN	73,92±0,45	78,51±0,54	76,90±0,44	77,15±0,79
E2LSH-2 R=0,1	RBFN	73,53±0,46	78,67±0,57	75,92±0,50	77,24±0,83
E2LSH-2 R=2,5	⊙ RBFN	81,57±0,51	82,41±0,49	81,38±0,77	82,19±0,53
E2LSH R=0,1	JRip	80,88±0,51	80,94±0,58	79,98±0,48	78,90±0,67
E2LSH-2 R=0,1	⊙ JRip	81,36±0,51	81,43±0,58	80,14±0,43	79,20±0,86
E2LSH-2 R=2,5	⊙ JRip	83,18±0,54	83,19±0,67	82,73±0,64	82,71±0,64
E2LSH R=0,1	DTNB	79,86±0,62	82,04±0,71	81,10±0,70	80,01±0,57
E2LSH-2 R=0,1	⊙ DTNB	80,10±0,70	82,46±0,70	81,11±0,75	80,35±0,66
E2LSH-2 R=2,5	⊙ DTNB	82,91±0,59	83,41±0,67	82,85±0,65	82,99±0,61
E2LSH R=0,1	NaB	73,09±0,88	78,12±0,51	75,74±0,30	76,59±0,68
E2LSH-2 R=0,1	⊙ NaB	71,28±1,10	78,16±0,56	74,18±0,48	76,38±0,80
E2LSH-2 R=2,5	⊙ NaB	80,66±0,52	82,31±0,63	80,98±0,62	82,13±0,61
E2LSH R=0,1	NBT	81,61±0,67	82,37±0,52	80,52±0,65	80,05±0,59
E2LSH-2 R=0,1	⊙ NBT	82,19±0,70	82,92±0,49	80,30±0,73	80,41±0,69
E2LSH-2 R=2,5	⊙ NBT	**83,36±0,64**	83,50±0,66	82,29±0,66	82,89±0,51
E2LSH R=0,1	J48	81,86±0,44	82,66±0,39	81,36±0,62	80,01±0,46
E2LSH-2 R=0,1	⊙ J48	82,34±0,41	83,31±0,37	81,72±0,63	80,41±0,56
E2LSH-2 R=2,5	⊙ J48	83,25±0,55	**83,78±0,42**	**83,09±0,70**	**83,07±0,56**

Zoznam použitej literatúry

[1] Machine learning repository.

[2] Yahoo! webspam-uk2007.

[3] Aliferis, C. F., Statnikov, A., Tsamardinos, I., Mani, S., and Koutsoukos, X. D. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *The Journal of Machine Learning Research 11* (2010), 171–234.

[4] Alpaydin, Ethem. *Introduction to Machine Learning*. The MIT Press.

[5] Andoni, A., and Indyk, P. *E2LSH 0.1 user manual*, 2005. Accessed: Dec. 24, 2013.

[6] Andoni, A., and Indyk, P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proceedings of 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)* (2006), pp. 459–468.

[7] Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry* (2004), ACM New York, NY, USA, pp. 253–262.

[8] Drozda, M., Bate, I., and Timmis, J. Bio-inspired error detection for complex systems. In *Proceedings of 17th IEEE Pacific Rim International Symposium on Dependable Computing (PRDC)* (2011), pp. 154–163.

[9] Elkan, C. Results of the KDD'99 classifier learning. *ACM SIGKDD Explorations Newsletter 1*, 2 (2000), 63–64.

[10] Fossi, M. e. a. Symantec global internet security threat report trends of 2009, 2009.

[11] Gama, J., and Brazdil, P. Cascade generalization. *Machine Learning 41*, 3 (2000), 315–343.

[12] Guo, J., Li, W., Song, X., Zhang, B., and Wang, Y. Software rejuvenation strategy based on components. In *Second World Congress on Software Engineering (WCSE)* (2010), vol. 2, IEEE, pp. 80–83.

[13] Han, Jiawei and Kamber Micheline. *Data Mining: Concepts and Techniques*. San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[14] Indyk, P., and Motwani, R. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1998), STOC '98, ACM, pp. 604–613.

[15] Kaynak, C., and Alpaydin, E. Multistage cascading of multiple classifiers: One man's noise is another man's data. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)* (San Francisco, CA, USA, 2000), Morgan Kaufmann Publishers Inc., pp. 455–462.

[16] Koc, L., Mazzuchi, T. A., and Sarkani, S. A network intrusion detection system based on a hidden naïve bayes multiclass classifier. *Expert Syst. Appl. 39*, 18 (Dec. 2012), 13492–13500.

[17] Li, Q., Zhu, K., Cheng, Y., and Zhang, J. Constrained Elements Based Software Rejuvenation Policy in Embedded Environment. *Journal of Computational Information Systems 9*, 16 (2013), 6391–6398.

[18] Marcek, S., and Drozda, M. Network intrusion detection with cascading classification. In *Proceedings of 5th International Conference on Intelligent Systems, Modelling, and Simulation (ISMS 2014)* (2014).

[19] McHugh, J. Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Trans. Inf. Syst. Secur. 3*, 4 (Nov. 2000), 262–294.

[20] Oliver, E. A., and Keshav, S. An empirical approach to smartphone energy level prediction. In *Proceedings of the 13th International Conference on Ubiquitous Computing* (New York, NY, USA, 2011), UbiComp '11, ACM, pp. 345–354.

[21] Peng, H., Long, F., and Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 8 (2005), 1226–1238.

[22] Pfahringer, B. Winning the KDD99 classification cup: Bagged boosting. *SIGKDD Explorations Newsletter 1*, 2 (Jan. 2000), 65–66.

[23] Rapidminer. Rapidminer tool, November 2013.

[24] SIGKDD. Kdd-cup 1999, 1999.

[25] Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A.-A. A detailed analysis of the kdd cup 99 data set. In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009* (2009).

[26] Wagner, D., Rice, A., and Beresford, A. Device analyzer, januar 2014.

[27] Wagner, D., Rice, A., and Beresford, A. Device Analyzer: Large-scale Mobile Data Collection. *SIGMETRICS Perform. Eval. Rev. 41*, 4 (april 2014), 53–56.

[28] Wood, P. e. a. Symantec global internet security threat report trends of 2016, 2016.

4 Publikácie

Publikované výsledky dizertačnej práce

Marcek, S., Drozda, M., Juhas, G., and Lehocki, F. Network intrusion detection in high dimensional space. In Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on (2009), IEEE, pp. 1–7.

Marcek, S., and Drozda, M. Network intrusion detection with cascading clas- sification. In Proceedings of 5th International Conference on Intelligent Systems, Modelling, and Simulation (ISMS 2014) (2014).

Marcek, S., and Drozda, M. Predicting system failures on mobile devices. In Proceedings of the Mediterranean Conference on Information and Communication Technologies 2015 (2016), Springer, pp. 499–508.

Ostatné

Gallo, Ondrej - Marček, Stanislav: Application of optical sensors in biomedical engineering. In: Meditech - Proceedings of the ESF Project Conference : Innovative Program of Modern Biomedical Technologies. Project No. SORO/JPD-26/2005. Bratislava, Slovakia, 26.5.2008. - Bratislava : STU v Bratislave, 2008. - ISBN 978-80-227-2881-2. - S. 159-164.

Marček, Stanislav - Gallo, Ondrej: Non-Invasive Patient Monitoring. In: Meditech - Proceedings of the ESF Project Conference : Innovative Program of Modern Biomedical Technologies. Project No. SORO/JPD-26/2005. Bratislava, Slovakia, 26.5.2008. - Bratislava : STU v Bratislave, 2008. - ISBN 978-80-227-2881-2. - S. 153-158.

Summary

The goal of this thesis is application and evaluation of the cascade detection in some practical scenario. Cascade detection is applicable for faults detection in wireless sensor networks, as an network intrusion detection system or failure detection in operating system. The benefits of cascade classification are in use on system with limited resources such as memory, battery or computation power. Its importance lies in the possibility significantly reduce false positive ratio.

Cascade classification consists of sequence of classifiers, that on earlier stage eliminates majority normal samples. For KDD99 dataset also most of the DOS attack, which is a majority of dataset. It is crucial for cascade detection that suspicious behaviour is evaluated on the next stage of cascade that is more specific and costlier algorithm.

In this work we have shown that the proposed approach gets comparable results to other researchers methodologies in detection rate. As a first stage in the cascade we have chosen an approximate nearest neighbour based algorithm, the E2LSH. The time/space complexity of E2LSH is polynomial in dimension.

We have applied this approach on several datasets. One of them was dataset from challenge to classify network intrusions, the KDD99. The winner of the KDD99 was an approach with the cost χ=0.2331. This result has been long not overcome. However, we applied our $C_1 \odot C_2$ and achieved minimal cost χ=0.20773. Generally, the best results were achieved by using C4.5 decision tree as a second stage on the cascade.

Application of the cascade classification on software rejuvenation dataset need more investigation in preprocessing stage, but it seems to be a good approach.