Analýza ML metód a algoritmov pre detekciu pokročilých útokov (APT)

Projekt INNOvative SOC/INNOSOC

Obsah

Úvod 3
Experimenty a výsledky 6
Datasety 6
DAPT2020 6
Použité algoritmy a metódy 9
Random forest 9
SVM 13
Logistická regresia 14
Nastavenia a parametre zvolených metód 15
Logistická regresia 15
Random forest 15
SVM 16
Vyhodnocovanie experimentov 16
Konfúzna matica 17
Accuracy 18
Precision 18
Recall 19
F1-Score 19
Výsledky 19
Zhrnutie 23

Úvod

Pokročilá perzistentná hrozba APT (Advanced Persistent Threat) je forma kybernetickej hrozby, ktorú predstavujú dobre financované organizácie, často s cieľom získať kľúčové informácie od cieľovej organizácie. APT hrozba sa líši od konvenčných a viac-stupňových útokov a je definovaná kombináciou troch slovíčok:

- **Pokročilá** – Advanced – Pokročilý aktér hrozieb je technicky zdatný, skúsený a schopný objavovať alebo zavádzať nové zraniteľnosti do systému a vyvíjať nové spôsoby, ako tieto zraniteľnosti zneužiť. Útočníci dokonca siahajú po rôznych útokoch ako napr. zero-day exploit, planted insiders, alebo supply chain exploitation, keď to obrana cieľa vyžaduje.
- **Pretrvávajúca** – Persistent – Pretrvávajúca hrozba je ktokoľvek, kto prejavuje odhodlanie útočiť až do dosiahnutia cieľa, je zručný a skúsený v prispôsobovaní sa nepriateľskému územiu, aby zostal neodhalený, a má zdroje na vývoj svojich TTP na porazenie obranných strategických alebo taktických protiopatrení. Tieto zdroje zahŕňajú rôzne aktíva a nie sú špecificky obmedzené len na finančné prostriedky. Ľudský kapitál je napríklad veľmi relevantným zdrojom. Pretrvávajúca hrozba bude mať prístup k personálnym kapacitám pre každú operáciu, ktorú iniciuje. Pracujú prevažne v tímoch, keďže rozsah a škála ich činností sú dostatočne rozsiahle. Ďalšími aktívami sú infraštruktúra, čas a finančné zdroje.
- **Hrozba** – Threat – Hrozba označuje jednotlivcov (napr. osoby, skupiny a organizácie), ktorí majú motiváciu, schopnosti a príležitosť zapojiť sa do škodlivých činov. Kľúčovou definujúcou črtou hrozby je jej zlomyseľný úmysel, ktorý je hnacou silou jej motivácie.

Pri uvažovaní o APT je dôležité si uvedomiť, že sa jedná o ľudí, keďže tento pojem bol špeciálne navrhnutý tak, aby ich odlíšil od jednoduchého malvéru. Ako tvrdí Richard Bejtlich: "Hrozba znamená, že protivník nie je kus bezduchého kódu." Technicky neexistuje spôsob, ako zachytiť ľudskú kreativitu alebo predstavivosť. Preto nemôžeme predpokladať, že vieme predpovedať, aký útok APT vykoná. Výskum často považuje APT útoky za cielené viacstupňové útoky. Avšak obmedzenie, že viacstupňový útok musí pozostávať aspoň z dvoch atomických alebo drobných udalostí, nie je pre APT útok povinné. Ak útočník potrebuje na dosiahnutie cieľa len jednu akciu, použije ju. Cesta najmenšieho odporu, ako časť pojmu Pokročilá v definícii, opisuje ich schopnosť využívať celé spektrum narušení. Napríklad, ak existuje známa zraniteľnosť s príslušným zneužívaným materiálom (*exploit material*), APT ju *môže* a *bude* využívať na dosiahnutie svojich cieľov.

Viacstupňový útok môže tiež zasiahnuť jeden konkrétny cieľ. Preto majú detekčné systémy problém rozlíšiť hrozby stojace za útokom.

APT sa snažia zostať neodhalené a preto používajú techniky skrývania (*concealment*), pričom jednou z najčastejšie používaných je technika *living-off-the-land*. Tieto techniky zahŕňajú maskovanie škodlivej činnosti za zdanlivo neškodné programy, ktoré využívajú nástroje už na počítačoch používateľov. Tieto nástroje sú často súčasťou operačného systému alebo iného softvéru inštalovaného používateľom. Tento typ útoku je obzvlášť náročné odhaliť, pretože útočníci zvyčajne nevytvárajú nové škodlivé súbory na cieľových počítačoch a bežné antivírusové skeny ich nemusia identifikovať.

Tieto techniky sa v konvenčnom spravodajstve kybernetických hrozieb CTI (Cyber Threat Iintelligence) nezohľadňujú, pretože vyžadujú vyššiu úroveň sofistikovanosti od útočníka a nemožno ich porovnávať so známymi postupmi. Tento fakt zdôrazňuje dôležitosť zohľadnenia týchto pokročilých taktík v bezpečnostných opatreniach a neustáleho aktualizovania CTI, aby sme boli o krok vpred s meniacim sa prostredím hrozieb.

Napriek tomu APT môže a bude používať známe vzory útokov, ako sú opakované útoky (*replay attacks*) alebo vloženie kódu (*injection*), prostredníctvom známych zraniteľností. Zavedené systémy založené na pravidlách a dátami riadené prístupy pre konkrétne TTP sú čiastočným riešením.

V najhoršom prejave možno APT opísať ako komplexného nepriateľa so širokým spektrom útokov charakterizovaného DoD (Department of Defense) a niekedy nazývaného integrované hrozby. Táto vlastnosť znamená, že nevyhľadávajú príležitosti na úder - na umožnenie svojich operácií využívajú politické, vojenské, spravodajské alebo ekonomické zdroje.

Historické príklady útokov APT, ktoré sú obzvlášť zaujímavé, sú útok Stuxnet, ktorý bol použitý na zničenie centrifúg na obohacovanie uránu v Natanzi v Iráne, Industroyer, ktorý spôsobil výpadok elektriny na Ukrajine v roku 2016, a TRITON, ktorý zaútočil na bezpečnostné prístrojové systémy ropného priemyslu. Medzi ďalšími útokmi APT predstavuje významný pilier vo vývoji APT prechod od prevažne špionáže k deštruktívnym útokom.

Okrem kradnutia informácií a manipulácie či dokonca ničenia cieľov môže APT tiež usilovať o inštaláciu *backdoor* softvéru, aby získal jednoduchý prístup v strategicky výhodných časoch. Okrem využití rôznych taktík, techník a procedúr TTP (*Tactics, Techniques, and Procedures*) sa typy útokov líšia aj podľa oblastí útočníka, cieľa, účelu a prístupu. Tabuľka nižšie zobrazuje rozlíšenie medzi typmi útokov.

	Konvečný útok	Viac-stupňový útok	APT útok
Útočník	prevažne jednotlivec	kriminálne motivované skupiny	Vysoko organizovaná, sofistikovaná, odhodlaná a dobre financovaná skupina
Cieľ útoku	Nešpecifikované, väčšinou individuálne systémy	Organizácie špecifické pre dané oblasti	Konkrétne ciele so strategickou hodnotou
Účel útoku	Finančný zisk, preukázanie schopností	Finančný zisk, preukázanie schopností	Konkurenčné výhody, strategické výhody
Prístup k útoku	Jednorazová, krátkotrvajúca operácia typu *smash-and-grab* (rozbi a vezmi)	Jednorazová, nápadná a neutajená operácia trvajúca od niekoľkých hodín až po týždne	Opakované pokusy, nízky a pomalý (*low-and-slow*) prístup, prispôsobenie sa na dlhodobý odpor obrany a časté skrývanie, kým zadné vrátka (*backdoor*) nemajú strategické využitie

Tab. 1: Porovnanie konvečných, viac-stupňových a APT útokov

V oblasti detekcie je možné použiť rôzne metodológie, pričom najbežnejším prístupom je diferenciácia nástrojov podľa ich cieľov detekcie. V zásade rozlišuje nasledovné metódy:

- Metódy založené na **pravidlách** (*rule-based*): Ide o metódu detekcie útokov, ktorá využíva vopred definované pravidlá a vzory na identifikáciu podozrivej aktivity. Tieto pravidlá a vzory sú zvyčajne založené na znalostiach známych útokov alebo vzorcov správania, ktoré sú v systéme považované za normálne. Preto sa táto kategória nazýva aj detekcia založená na signatúrach alebo detekcia založená na znalostiach, ktorú možno ďalej rozdeliť na porovnávanie signatúr, analýzu prechodov stavov a expertné systémy.
- Metódy založené na **anomáliách** (*anomaly-based*): Detekcia anomálií identifikuje nezvyčajné vzory alebo prípady v dátach, ktoré nezodpovedajú očakávanému správaniu. Kľúčovým rozdielom je, že detekcia anomálií nerozlišuje medzi známymi a neznámymi útokmi. Keďže detekcia anomálií sa zvyčajne spolieha na správanie systému na detekciu odchýlok, tieto systémy sú menej rozšírené ako systémy založené na pravidlách, najmä keď vezmeme do úvahy, že proces učenia musí predchádzať variantom strojového učenia.
- **Štatistické** metódy: Táto skupina zahŕňa všetky metódy založené na štatistickej analýze pre účely predikcie a detekcie. Väčšina štatistických metód detekcie útokov využíva pravdepodobnostné modely a trénovacie dáta na sledovanie konkrétneho správania sa.
- **Klasifikačné** metódy: Klasifikačná skupina zahŕňa metódy, ktoré tiež používajú techniky strojového učenia, ale s riadeným učením, pri ktorom sa učia neškodné a škodlivé správanie a potom označujú nové aktivity podľa naučených tried. Okrem toho môže viactriedne označovanie (*labelling*) dosiahnuť presnejšie

detekcie, ak je trénovací dataset primerane označený a obsahuje reprezentatívnu vzorku útokov z každej triedy. Nové správanie, ktoré nepripomína ani predchádzajúce neškodné, ani predchádzajúce škodlivé vzorce, nemožno primerane riešiť.

- **Ľudské znalosti**: Detekčné systémy sú zaradené do kategórie ľudských znalostí, keď využívajú silné odborné znalosti v danej oblasti alebo sa spoliehajú na CTI.

Experimenty a výsledky

Datasety

DAPT2020

Dataset DAPT2020 (Dataset for Advanced Persistent Threats (APT) 2020) je súbor dát kybernetickej bezpečnosti špeciálne navrhnutý na výskum v oblasti detekcie a analýzy útokov pokročilých pretrvávajúcich hrozieb (APT). Bol vytvorený a zverejnený v roku 2020 laboratóriom Cyber Threat Intelligence (CTI) na Univerzite Nového Južného Walesu (UNSW) v Canberre v Austrálii.

Primárnym cieľom bolo riešiť nedostatok realistických, aktuálnych a vysokokvalitných datasetov, ktoré by zachytili viacstupňovú, nenápadnú a dlhodobú povahu APT útokov. Na rozdiel od súborov dát obsahujúcich jednoduché zneužívania alebo krátkodobé útoky, DAPT2020 sa zameriava na simuláciu celého reťazca kybernetického útoku *cyber kill chain*, ktorý predstavuje sofistikované APT kampane a poskytuje údaje na vývoj detekčných systémov pre laterálny pohyb (angl. *lateral movement*), eskaláciu privilégií (*privilege escalation*), exfiltráciu údajov (*data exfiltration*) a aktivity velenia a riadenia C2 (*command & control*).

Súbor dát bol vygenerovaný v kontrolovanom testovacom prostredí typu sandbox, ktoré bolo navrhnuté tak, aby emulovalo skutočnú podnikovú sieť, nie zo živého produkčného systému.

- **Topológia siete**: Simulovala sa malá podniková sieť s viacerými operačnými systémami (Windows 10, Windows Server, Ubuntu) a so štandardnými službami (Active Directory, webové servery, zdieľané súbory).
- **Simulácia útoku**: Výskumníci spustili skutočný malvér a útočné nástroje APT podľa zdokumentovaných taktík, techník a postupov TTP (Tactics, Techniques, and Procedures) známych APT skupín.
- **Neškodná prevádzka na pozadí**: Na vytvorenie realistickej zmesi boli počas niekoľkých dní generované aj bežné aktivity používateľov (prehliadanie webu, e-mail, prenos súborov atď.) pomocou automatizovaných skriptov a dobrovoľníkov.

Dataset bol vytvorený simuláciou správania sa APT útokov v cloudovej sieti počas piatich dní:

- **1.deň**: Normálna prevádzka - ping, prehliadanie webu, operácie so súbormi, ...
- **2. deň**: Prieskum (*reconnaissance*) - nmap, sqlmap, skenovanie zraniteľností (*vulnerability scanning*), ...
- **3. deň**: Vybudovanie základne (*foothold*) - reverzné shelly (*reverse shells*), zneužívanie SQL (*SQL exploitation*), obchádzanie autentifikácie (*authentication bypass*)
- **4. deň**: Laterálny pohyb (*lateral movement*) - skenovanie internej siete, zneužívanie zraniteľností (*exploiting vulnerabilities*)
- **5. deň**: Exfiltrácia údajov (*data exfiltration*) - prenosy FTP, nahrávanie na Google Drive, komunikácia C&C

Dataset obsahuje 79 parametrov/príznakov (features) sieťového toku vrátane trvania toku, veľkostí paketov, časov medzi príchodmi, príznakov TCP, dĺžok hlavičiek, rýchlostí prenosu bajtov/paketov a rôznych štatistických ukazovateľov.

Dataset je získaný z viacerých zdrojov, čo poskytuje holistický pohľad, ktorý je kľúčový pre detekciu APT. Súbor dát DAPT2020 zahŕňa:

- **Úplné zachytenie paketov (PCAP)**: Kompletná sieťová prevádzka z testovacieho prostredia, ktorá umožňuje hĺbkovú kontrolu protokolov.
- **Údaje o toku siete (NetFlow)**: Zhrnuté protokoly pripojení, ktoré sú menej podrobné, ale efektívne pre analýzu.
- **Systémové protokoly**:
    - Protokoly/logy udalostí systému Windows: Protokoly zabezpečenia, systému a aplikácií, ktoré sú kľúčové na detekciu laterálneho pohybu (*lateral movement*), eskalácie privilégií (*privilege escalation*) a pretrvávanie (*persistence*).
    - Sysmon protokoly/logy: Podrobné udalosti vytvárania procesov, sieťového pripojenia a zmeny súborov (neoceniteľné pre forenzné sledovanie).
- **Artefakty hostiteľa** (*host-based artifacts*):
    - Výpisy pamäte: Pre analýzu škodlivého softvéru.
    - Obrazy diskov: Na forenznú investigatívu.
    - Výpisy procesov a snímky registrov.
- **Protokoly/logy upozornení a detekcií**: Výstupy z bezpečnostných nástrojov ako Suricata (IDS) a Elastic Security (SIEM) nasadených počas experimentu.

Dataset obsahuje dve odlišné, prepojené APT kampane, ktoré boli inšpirované skupinami z reálneho sveta (napr. podobné APT29/Cozy Bear alebo APT32/OceanLotus):

1. **Kampaň 1**: Zameriava sa na krádež a únik údajov. Zahŕňa počiatočný kompromitačný útok, laterálny pohyb na databázovom serveri, umiestnenie údajov a únik prostredníctvom skrytých kanálov.
2. **Kampaň 2**: Zameriava sa na nasadenie ransomvéru. Zahŕňa vytvorenie základne (foothold), eskaláciu privilégií (*escalating privileges*), deaktiváciu obrany (*disabling defenses*) a nasadenie ransomvéru v sieti.

Tieto kampane zahŕňajú fázy ako prieskum (*reconnaissance*), počiatočný prístup (napr. *spear-phishing*), vykonanie (*execution*), pretrvávanie (*persistence*), laterálny pohyb (*lateral movement*) a exfiltrácia (*exfiltration*).

Dataset má nasledujúce kľúčové vlastnosti:

- Používa moderné TTP a vzorky malvéru z časového rámca 2019 – 2020.
- Zachytáva útoky, ktoré sa odohrávajú v priebehu dní, nie sekúnd.
- Útoky sú starostlivo označené presnými časovými pečiatkami a namapované na rámec MITRE ATT&CK, vďaka čomu sú vynikajúce pre kontrolované strojové učenie a výskum založený na TTP.
- Synchrónny zber sieťových (network) a hostiteľských (host) záznamov umožňuje koreláciu medzi zdrojmi, čo je nevyhnutnosť pre lov APT (APT hunting).

Dataset je verejne dostupný pre výskumné účely.

https://www.kaggle.com/datasets/sowmyamyneni/dapt2020

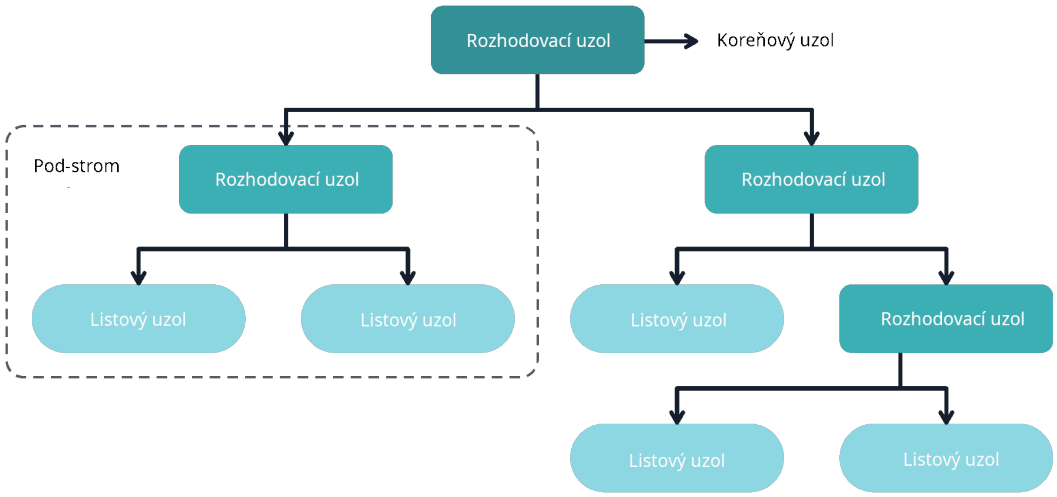
Detailnejšie informácie ohľadom datasetu možno nájsť v podkladovom dokumente DAPT_at_MLHat2020.pdf.

Použité algoritmy a metódy

Random forest

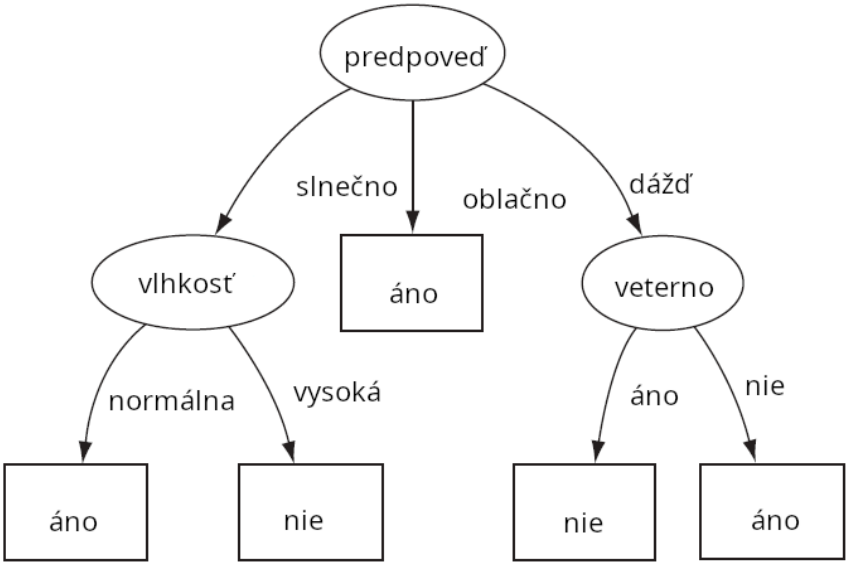
Rozhodovacie stromy

Rozhodovací strom je neparametrický algoritmus, ktorý sa používa na klasifikačné aj regresné úlohy. Má hierarchickú stromovú štruktúru, ktorá pozostáva z koreňového uzla, vetiev, vnútorných uzlov a listových uzlov. Ako možno vidieť z nižšie uvedeného diagramu, rozhodovací strom začína koreňovým uzlom, ktorý nemá žiadne prichádzajúce vetvy.



Obr. 1: Rozhodovací strom

Odchádzajúce vetvy z koreňového uzla sa potom privádzajú do interných uzlov, známych aj ako rozhodovacie uzly. Na základe dostupných funkcií oba typy uzlov vykonávajú hodnotenia s cieľom vytvoriť homogénne podmnožiny, ktoré sú označené listovými uzlami alebo koncovými uzlami. Listové uzly predstavujú všetky možné výsledky v rámci súboru údajov. Ako príklad si možno predstaviť, že ste sa snažíme posúdiť, či by sme mali alebo nemali ísť hrať golf, môžeme na výber použiť nasledujúce pravidlá rozhodovania:

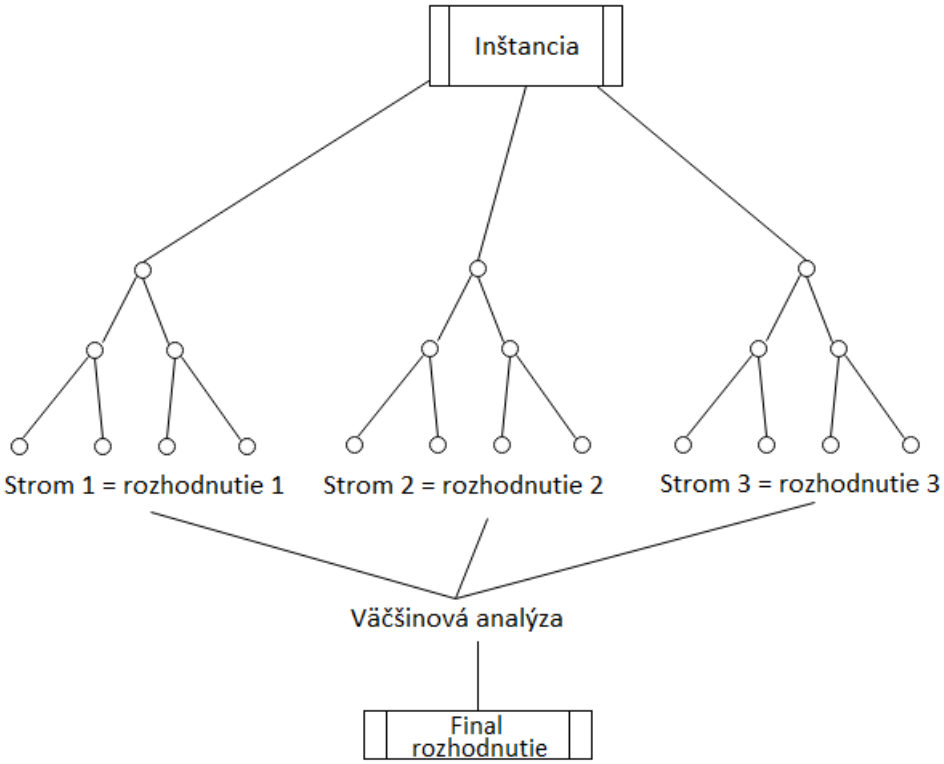


Obr. 2: Rozhodovací strom pre dataset Golf

Tento typ štruktúry vývojového diagramu tiež vytvára ľahko stráviteľnú reprezentáciu rozhodovania, čo umožňuje rôznym skupinám v celej organizácii lepšie pochopiť, prečo bolo rozhodnutie prijaté. Učenie rozhodovacieho stromu využíva stratégiu rozdeľuj a panuj tým, že vykonáva chamtivé vyhľadávanie s cieľom identifikovať optimálne body rozdelenia v strome. Tento proces rozdelenia sa potom opakuje zhora nadol, rekurzívnym spôsobom, kým všetky alebo väčšina záznamov nie sú klasifikované pod špecifickými štítkami tried. To, či sú všetky dátové body klasifikované ako homogénne množiny, do značnej miery závisí od zložitosti rozhodovacieho stromu. Menšie stromy sú ľahšie schopné dosiahnuť čisté listové uzly – t. j. dátové body v jednej triede. Ako však strom rastie, je čoraz ťažšie udržať túto čistotu a zvyčajne to vedie k tomu, že do daného pod-stromu spadá príliš málo údajov. Keď k tomu dôjde, je tento jav známy ako fragmentácia údajov a často to môže viesť k nadmernému prispôsobeniu. Výsledkom je, že rozhodovacie stromy uprednostňujú malé stromy, čo je v súlade so zásadou úspornosti podľa princípu „Occamova britva“, čo znamená, že "entity by sa nemali množiť pokiaľ to nie je nevyhnutné". Inak povedané, rozhodovacie stromy by mali zvyšovať svoju zložitosť iba v prípade potreby, pretože najjednoduchšie vysvetlenie je často najlepšie. Aby sa znížila zložitosť a zabránilo sa nadmernému prispôsobeniu, zvyčajne sa používa prerezávanie; Toto je proces, ktorý odstraňuje vetvy, ktoré sa rozdeľujú na prvkoch s nízkou dôležitosťou. Prispôsobenie modelu sa potom môže vyhodnotiť prostredníctvom procesu krížovej validácie.

Random Forest

Náhodné lesy (*random forest*) sú metódy súborového učenia (*ensamble learning*), ktoré sa používajú buď na klasifikáciu, alebo na regresné úlohy. Náhodné lesy sa skladajú z viacerých rozhodovacích stromov, ktoré sa kombinujú do jednomyseľného rozhodnutia alebo klasifikácie (obrázok nižšie).



Obr. 3: Random forest – princíp činnosti

Náhodné lesy sú lepšie ako bežné rozhodovacie stromy, pretože nespôsobujú nadmerné prispôsobovanie dát a následné pretrénovanie modelu. Toto sa dosiahnuť vďaka použitiu metód, ktoré využívajú náhodné podpriestory (napríklad *bagging*) na zníženie korelácie medzi jednotlivými stromami. Elementárne algoritmy/stromy sú trénované na rôznych podmnožinách príznakového priestoru, ktoré sú tiež náhodne vyberané. Súbor modelov rozhodovacích stromov pomocou metódy náhodného podpriestoru možno zostaviť pomocou nasledujúceho algoritmu.

Pre N trénovacích objektov a počet príznakov D , postupujeme nasledovne:

1. Zvolí sa parameter L ako počet jednotlivých stromov v súbore (*ensemble*).
2. Pre každý jednotlivý l-tý strom sa vyberie $d_l < D$ ako počet príznakov pre l. Typicky sa pre všetky stromy používa len jedna hodnota.
3. Pre každý strom sa vytvorí tréningovú podmnožinu X_n pomocou bootstrap metódy.

Vytvorenie rozhodovacích stromov z X_n vzoriek sa vykoná nasledovne:

1. Vyberú sa náhodné príznaky d_l zo zdroja, potom optimálne rozdelenie trénovacej množiny obmedzí svoje vyhľadávanie na ne.
2. Podľa daného kritéria sa vyberie najlepší atribút a podľa neho sa urobí rozdelenie v strome.
3. Strom sa vytvára, kým v každom liste nezostane viac ako n_{min} objektov, kým sa nedosiahne určitú výšku stromu, alebo kým nie je vyčerpaná tréningová sada.

Aby bolo možné aplikovať súbor modelu (ensemble) na nový dátový objekt, je potrebné skombinovať výsledky jednotlivých modelov väčšinovým hlasovaním alebo kombinovaním *a posteriori* pravdepodobností. Príklad finálneho klasifikátora je nasledovný:

$$a(x) = \frac{1}{L}\sum_{i=1}^{L} l_i(x)$$

Medzi základné parametre metódy *random forest* patria:

- počet stromov: Čím viac stromov, tým je lepšia kvalita modelu, ale čas trénovania a pracovná záťaž algoritmu sa tiež proporcionálne zvyšujú. Často, s rastúcim počtom stromov, kvalita trénovacej množiny stúpa (môže dokonca dosiahnuť 100% presnosť), ale kvalita testovacej množiny je asymptotická (takže možno odhadnúť minimálny požadovaný počet stromov).
- počet príznakov pre selekciu rozdelenie stromu: S rastúcim počtom príznakov sa predlžuje aj čas vytvárania náhodného lesa a stromy sú jednotnejšie než predtým. Často sa v klasifikačných problémoch volí počet atribútov ako hodnota $\sqrt{D}$ a pre regresné problémy ako hodnota $D/3$.
- maximálna hĺbka stromu: Čím je hĺbka menšia, tým rýchlejšie sa algoritmus vytvorí a bude fungovať. S rastúcou hĺbkou sa dramaticky zvyšuje kvalita počas tréningu. Kvalita sa môže tiež zvýšiť na testovacej sade. Odporúča sa používať maximálnu hĺbku (okrem prípadov, keď je príliš veľa tréningových objektov a získavame veľmi hlboké stromy, ktorých stavba trvá značný čas). Pri používaní plytkých stromov zmena parametrov spojených s obmedzením počtu objektov v liste a pri rozdeľovaní nevedie k výraznému efektu (listy sú už veľké). Používanie plytkých stromov sa odporúča pri úlohách s veľkým počtom šumových (*noisy*) alebo mimo-ležiacich objektov (*outliers*), ktoré predstavujú náhodné alebo irelevantné dáta.
- kritérium nečistoty (*impurity criterion*): Toto kritérium slúži na výber príznaku (rozhodovacieho pravidla) pre vetvenie. Pre regresné problémy sa zvyčajne používa stredná kvadratická chyba MSE (*Mean Squared Error*) alebo stredná absolútna chyba MAE (Mean Absolute Error). Pri klasifikačných problémoch je to Giniho kritérium, entropia alebo chyba klasifikácie. Rovnováha a hĺbka stromov sa môžu líšiť v závislosti od konkrétnej funkcie nečistoty, ktorá je zvolená.

SVM

Metóda podporných vektorov je klasifikačnou a regresnou metódou, ktorú navrhol v 90-tych rokoch V. Vapnik a ktorá v podstate určuje hraničnú krivku oddeľujúcu 2 rôzne triedy. Hlavnou myšlienkou je nájdenie takej nadroviny, ktorá by optimálne rozdelila v príznakovom priestore vstupné dáta. Pre danú množinu trénovacích dát patriacich do 2 tried $(x_1, y_1), \ldots, (x_l, y_l)$, kde $x_i \in R_n$ a $y_i \in \{-1, 1\}$ sa teda snažíme nájsť takú nadrovinu $wx + b = 0$, aby sme mohli separovať dáta. Zo všetkých nadrovín určených w a b je len jedna taká, ktorá maximalizuje vzdialenosť medzi najbližšími bodmi z týchto tried a nadrovinou - optimálna nadrovina a platí:

$$d(w,b) = \min_{\{x_i | y_i = 1\}} \frac{wx_i + b}{|w|} - \max_{\{x_i | y_i = -1\}} \frac{wx_i + b}{|w|}$$

a zodpovedajúca minimálna a maximálna hodnota je -1, resp. +1

$$d(w,b) = \left| \frac{1}{\mathbf{w}} - \frac{-1}{\mathbf{w}} \right| = \frac{2}{|\mathbf{w}|} .$$

Minimalizáciou výrazu $|w|^2 / 2$ v závislosti na podmienke $y_i (wx_i + b) \geq 1$ dostaneme pre daný pár nadrovín maximálnu vzdialenosť 2 tried. Riešenie optimalizačného problému SVM je dané sedlovým bodom (*saddle point*) Lagrangeovej funkcie:

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i \left[y_i \left(wx_i + b \right) - 1 \right],$$

kde α_i je Lagrangeov multiplikátor. Zatiaľ čo $L(w, b, \alpha)$ sa zmenšuje s w a b, derivácie $L(w, b, \alpha)$ sú pre všetky $\alpha_i \geq 0$ rovné nule. Riešenie je dané

$$\overline{w} = \sum_{i=1}^{l} \overline{\alpha}_i y_i x_i \qquad \overline{b} = -\frac{1}{2} \overline{w} \left[x_r + x_s \right],$$

kde x_r a x_s sú podporné vektory, ktoré patria do triedy $y_r = 1$, resp. $y_s = -1$. Dáta v pôvodnom príznakovom priestore nie sú separovateľné žiadnou lineárnou nadrovinou, ale môžu byť lineárne separovateľné v nelineárne mapovanom priestore príznakov, ktorý je odlišný od toho pôvodného a zvyčajne má vyššiu dimenziu. Toto nelineárne mapovanie $K(.,.)$ sa uskutočňuje pomocou jadrových funkcií (*kernel functions*):

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{l} \overline{\alpha}_i y_i K(x_i, x) + b \right) \sum_{i=1}^{N} \alpha_i t_i K(x, x_i) + b,$$

kde yi sú cieľové hodnoty, $\sum_{i=1}^{l} \overline{\alpha}_i y_i = 0$ a $\overline{\alpha}_i > 0$. Jadro $K(.,.)$ sa vytvára tak, aby spĺňalo špecifické vlastnosti v závislosti na danom probléme a možno napísať:

$$K(x, y) = \Phi(x)\Phi(y) \ ,$$

kde Φ(x) je mapovanie vstupného príznakového priestoru do nového priestoru s potenciálne nekonečnou dimenziou. Existujú 3 jadrové funkcie zabezpečujúce nelineárne mapovanie:

- polynomiálna $K(x, y) = (xy + 1)^z$ – parameter z predstavuje stupeň polynómu
- Gaussove radiálne bázové funkcie (Radial Basis Functions RBF) $K(x, y) = exp\{-[(x - y)^2 / 2\sigma^2]\}$ – kde parameter σ je štandardná odchýlka gaussovskej funkcie
- MLP funkcia $K(x, y) = tanh(scale(xy) - offset)$ – *scale* a *offset* sú dané parametre
- Sigmoidálna $K(x, y) = tanh(\beta xy + \theta) - \beta, \theta$ sú dané parametre.

Logistická regresia

Logistická regresia, tiež známa ako logistická regresná analýza, je metóda štatistickej analýzy, ktorá predpovedá zodpovedajúce binárne výsledky na základe vzťahu medzi nezávislou premennou a závislou premennou, a preto ju možno použiť na klasifikáciu dvoch tried. Klasifikácia je v podstate regresný problém s diskrétnym výstupom - algoritmus logistickej regresie odráža túto povahu.

Nech X označuje vstupný priestor klasifikácie a $C = \{0, 1\}$ označuje výstupný priestor dvojtriednej klasifikácie. Pre vstupné dáta $\boldsymbol{x} \in X$ je logistická regresia $P(c|\boldsymbol{x})$ pre predikciu ich kategórie ako $c \in C$ daná nasledovne:

$$P(c|x) = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{1}{1 + \exp(-z)} \ ,$$

kde $\sigma(\cdot)$ sa označuje ako logistická funkcia, známa aj ako sigmoidná alebo sigmoidálna funkcia, ktorá sa tiež používa ako jedna z aktivačných funkcií v umelých neurónových sieťach.

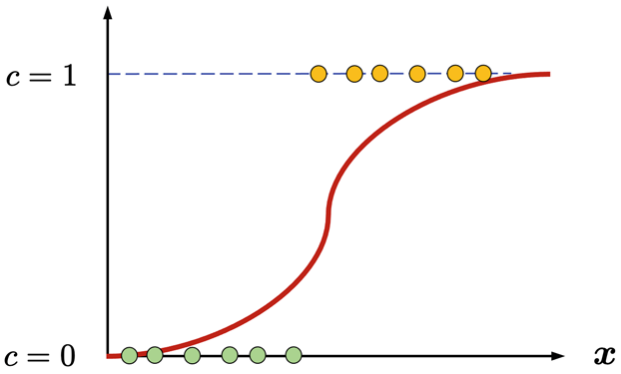
Logistická regresia má nasledujúcu vlastnosť, t. j. $P(c = 0|\boldsymbol{x}) = 1 - P(c = 1|\boldsymbol{x})$. Je to preto, že rozdelenie pravdepodobnosti dvojtriednej klasifikácie by malo spĺňať podmienku $P(c = 0|\boldsymbol{x}) + P(c = 1|\boldsymbol{x}) = 1$. Tiež nech $z = \boldsymbol{w}^T\boldsymbol{x} - b$, z čoho sa získa nasledovné:

$$P(c = 1|x) = \sigma\left(\boldsymbol{w}^T x - b\right) = \frac{1}{1 + \exp\left(-\left(\boldsymbol{w}^T x - b\right)\right)},$$

$$P(c = 0|x) = 1 - \sigma\left(\boldsymbol{w}^T x - b\right) = 1 - \frac{1}{1 + \exp\left(-\left(\boldsymbol{w}^T x - b\right)\right)}$$

$$= \frac{\exp\left(-\left(\boldsymbol{w}^T x - b\right)\right)}{1 + \exp\left(-\left(\boldsymbol{w}^T x - b\right)\right)}.$$

Schematický diagram algoritmu klasifikácie logistickej regresie je znázornený na nasledovnom obrázku.



Obr. 4: Princíp algoritmu klasifikácie logistickej regresie

Vyššie uvedená metóda logistickej regresie je použiteľná len pre klasifikáciu dvoch tried. Neskôr sa objavili niektoré rozšírené verzie logistickej regresie pre viactriedne klasifikačné problémy, vrátane multi-normálnej logistickej regresie, usporiadanej logistickej regresie a podmienenej logistickej regresie.

Nastavenia a parametre zvolených metód

Logistická regresia

- C:1.0
- class_weight:None
- dual:False
- fit_intercept:True
- intercept_scaling:1
- l1_ratio:None
- max_iter:1000
- multi_class:deprecated
- n_jobs:None
- penalty:l2
- random_state:42
- solver:lbfgs
- tol:0.0001
- verbose:0
- warm_start:False

Random forest

- bootstrap:True
- ccp_alpha:0.0
- class_weight:None
- criterion:gini
- max_depth:None

- max_features:sqrt
- max_leaf_nodes:None
- max_samples:None
- min_impurity_decrease:0.0
- min_samples_leaf:1
- min_samples_split:2
- min_weight_fraction_leaf:0.0
- monotonic_cst:None
- n_estimators:100
- n_jobs:None
- oob_score:False
- random_state:42
- verbose:0
- warm_start:False

SVM

- C:1.0
- break_ties:False
- cache_size:200
- class_weight:None
- coef0:0.0
- decision_function_shape:ovr
- degree:3
- gamma:scale
- kernel:rbf
- max_iter:-1
- probability:False
- random_state:42
- shrinking:True
- tol:0.001
- verbose:False

Vyhodnocovanie experimentov

Viacero techník bolo v priebehu času navrhnutých ako kvantitatívne miery pre hodnotenie alebo evaluáciu:

- Pokrytie (*Coverage*) môže označovať pokrytie zraniteľností alebo pokrytie útokov, ktoré využíva existujúce zoznamy možných útokov alebo zraniteľností na určenie, ktoré sú detegované a ktoré nie.

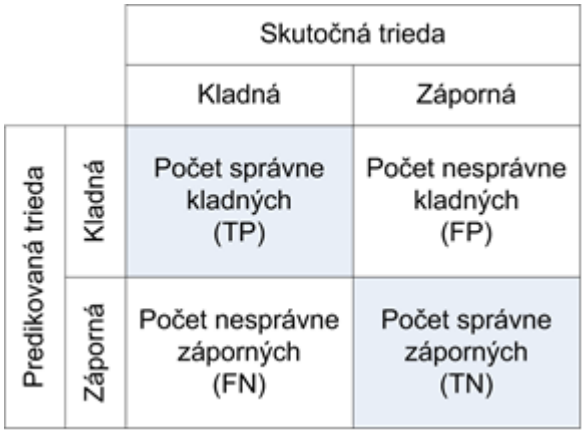
- Pravdepodobnosť detekcie a falošné poplachy (*Probability of detection and false alarms*) označujú falošné a skutočné poplachy vytvorené v danom časovom rámci. Skutočné poplachy označujú poplachy, ktoré sú pripísateľné útoku, zatiaľ čo falošné poplachy nie sú spôsobené útokom. Môže byť zobrazená ako prevádzková krivka prijímača ROC (*Receiver Operating Curve*) ako agregácia. Percento skutočných a falošných poplachov tak kontrastuje a predstavuje detekčný systém v rôznych režimoch prevádzky.
- Odolnosť (*Resistance*) voči útokom namiereným na detekčné systémy narušenia IDS (I*ntrusion Detection Systems*): Podľa tejto metriky sa merajú rôzne útoky na detekčný systém a kontroluje sa, či detekčný systém naďalej funguje správne. Tieto útoky môžu byť napríklad zjavne úmyselné útoky na narušenie. Do tejto kategórie patrí aj šírka prenosového pásma, ale uvádza, aké množstvo záťaže je všeobecne prijateľné.
- Schopnosť korelovať udalosti opisuje schopnosť detekčného systému korelovať útoky z rôznych zdrojov. Predstaviteľné kategórie sú korelácia zdrojových údajov, korelácia alarmov a akákoľvek korelácia.
- Schopnosť identifikovať útok znamená, ako dobre detekčný systém priradí poplachu označenie.
- Schopnosť určiť úspešnosť útoku: Ďalším rozdielom je to, ako dobre dokáže detekčný systém rozlíšiť úspešné a neúspešné útoky.
- Ďalšie metriky (nesúvisiace s výkonom detekcie) sú jednoduchosť použitia, údržba, problémy s nastavením, požiadavky na zdroje, dostupnosť a kvalita podpory.

Pre vyhodnocovanie experimentov boli použité nasledovné metriky.

Konfúzna matica

Konfúzna matica je jedným z najpopulárnejších spôsobov vyhodnocovania akéhokoľvek klasifikačného modelu. Hoci matica sama o sebe nie je metrikou, maticovú reprezentáciu možno použiť na definovanie rôznych metrík, z ktorých všetky sa stanú dôležitými v nejakom konkrétnom prípade alebo scenári. Konfúznu maticu je možné vytvoriť pre binárnu klasifikáciu, rovnako ako aj pre klasifikačný model s viacerými triedami.

Konfúzna matica sa vytvorí porovnaním predikovaného označenia triedy dátového bodu s jeho skutočným označením triedy. Toto porovnanie sa zopakuje pre celý súbor dát (dataset) a výsledky tohto porovnania sa zostavia v maticovom alebo tabuľkovom formáte. Táto výsledná matica predstavuje konfúznu maticu (obr. nižšie).



Obr. 5: Konfúzna matica

Vo všeobecnosti máme zvyčajne 2 triedy – kladnú/pozitívnu a zápornú/negatívnu triedu. Na základe tejto štruktúry môžeme jasne vidieť štyri pojmy dôležitosti.

• True Positive (TP) – Počet správne kladných: Toto je suma celkového počtu príkladov (dátových bodov z datasetu) z pozitívnej triedy, kde sa skutočné označenie triedy rovnalo predpovedanému označeniu triedy, t. j. celkovým počtom prípadov, v ktorých sme správne predpovedali označenie pozitívnej triedy pomocou nášho modelu.

• False positive (FP) – Počet nesprávne kladných: Toto je počet celkových prípadov z negatívnej triedy, kde ich model nesprávne klasifikoval tým, že ich predpovedal ako pozitívne. Odtiaľ pochádza názov, falošne pozitívny.

• True Negative (FN) – Počet nesprávne záporných: Toto je počet celkového počtu príkladov z negatívnej triedy, kde sa skutočné označenie triedy rovnalo predpovedanému označeniu triedy, t. j. celkový počet prípadov, v ktorých sme správne predpovedali označenie negatívnej triedy pomocou nášho modelu.

• False negatívne (FN) – Počet nesprávne záporných: Toto je počet celkového počtu prípadov z pozitívnej triedy, kde ich náš model nesprávne klasifikoval tým, že ich predpovedal ako negatívne. Odtiaľ pochádza názov, falošne negatívny.

Accuracy

Táto metrika je jednou z najpopulárnejších spôsobov merania resp. vyhodnocovania efektívnosti ľubovoľného klasifikátora. Je definovaná ako celková presnosť alebo podiel správnych predpovedí modelu. Vzťah pre výpočet z konfúznej matice je nasledovný:

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

Precision

Precision (presnosť), známa aj ako kladná prediktívna hodnota, je ďalšou metrikou, ktorú možno odvodiť z konfúznej matice. Táto metrika je definovaná ako počet predikcií, ktoré

sú skutočne správne alebo relevantné zo všetkých predikcií založených na kladnej triede. Vzorec pre túto metriku je nasledovný:

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

Model s vysokou hodnotou presnosti identifikuje vyšší podiel kladnej triedy v porovnaní s modelom s nižšou presnosťou. Táto metrika sa stáva dôležitou v prípadoch, keď sa viac zaujímame o nájdenie maximálneho počtu kladných tried, aj keď sa celková presnosť zníži.

Recall

Recall (úplnosť) je metrika modelu na identifikáciu percenta relevantných dátových bodov. Je definovaná ako počet inštancií kladnej triedy, ktoré boli správne predikované. Označuje sa aj ako miera pokrytia alebo citlivosť. Vzťah pre vypočítanie tejto metriky je nasledovný:

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

Táto metrika sa stáva dôležitým ukazovateľom efektívnosti klasifikátora v scenároch, v ktorých chceme zachytiť čo najväčší počet prípadov konkrétnej triedy, aj keď to spôsobí zvýšenie falošných detekcií/predikcií. Ako príklad možno uviesť prípad bankového podvodu, model s vysokou hodnotou metriky recall nám poskytne vyšší počet prípadov potenciálnych podvodov. Rovnako to však zvýši vyvolanie poplachov pre väčšinu podozrivých prípadov.

F1-Score

Existujú prípady, v ktorých chceme dosiahnuť vyváženú optimalizáciu presnosti a úplnosti. F1-Score je metrika, ktorá je harmonickým priemerom metrík precision a recall a pomáha nám optimalizovať klasifikátor pre vyváženú presnosť a úplnosť predikcie. Vzorec pre výpočet F1-Score je nasledovný:

$$F = \frac{2.P.R}{P + R}$$

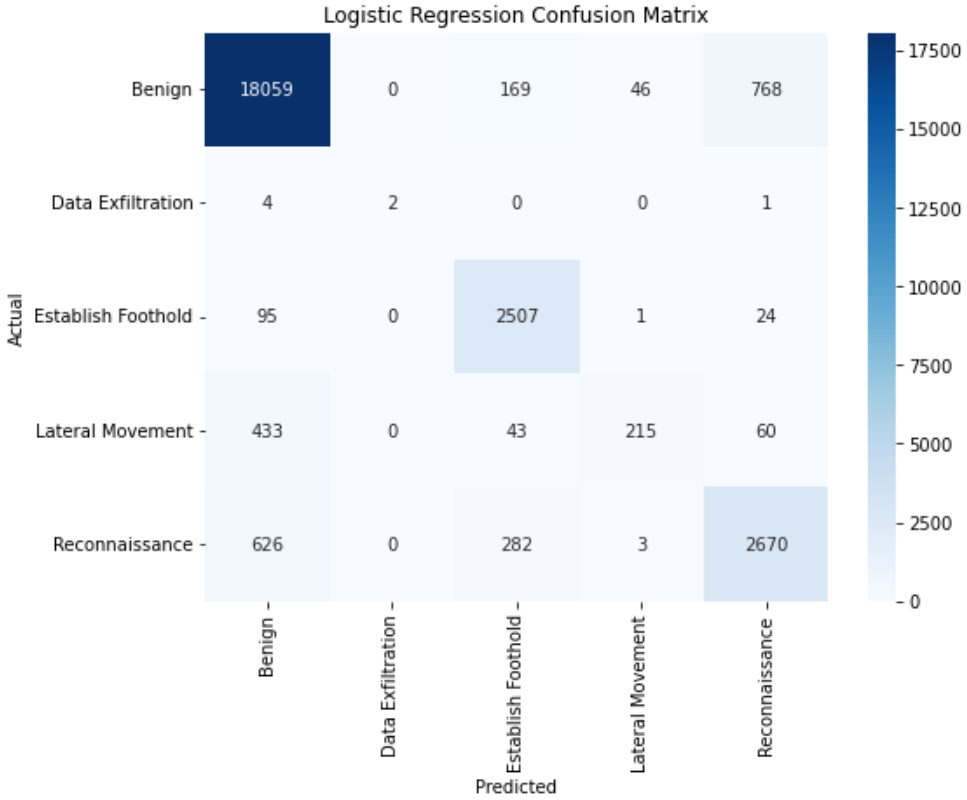
Výsledky

Tabuľka 3 prehľadne zobrazuje výsledky testovania pre zvolené klasifikačné metódy.

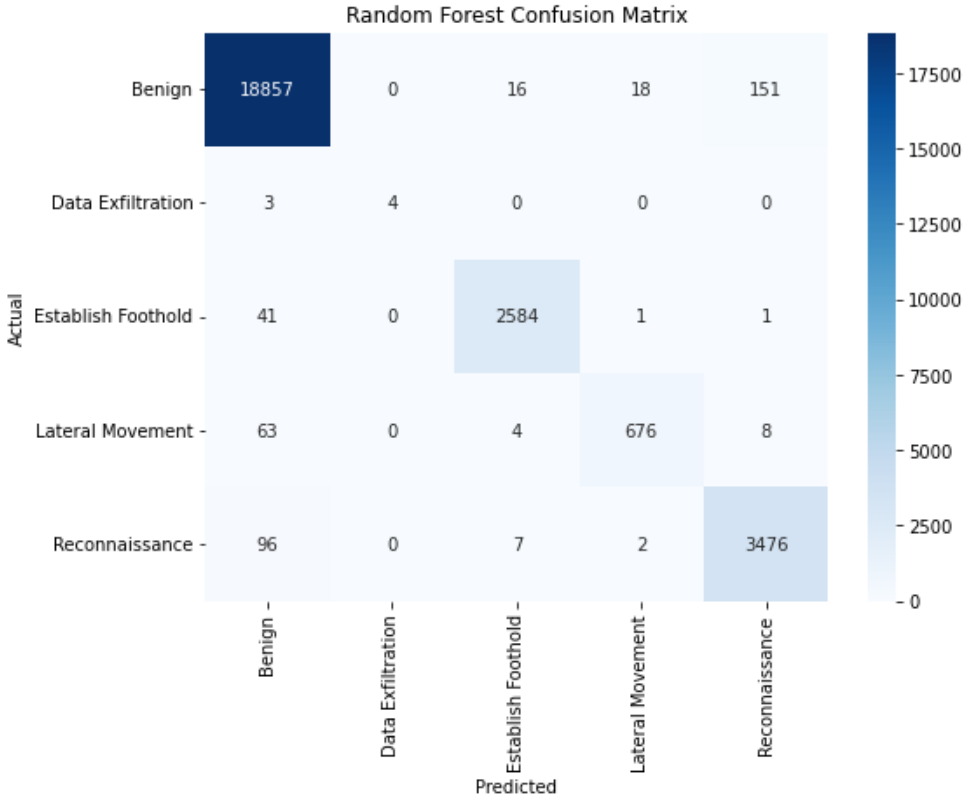
Metóda	Accuracy	Precision	Recall	F1-score	Processing time (sec.)
Log.Regression	0.9018	0.9005	0.9018	0.8970	0.17
Random Forest	**0.9842**	**0.9842**	**0.9842**	**0.9842**	0.19
SVM	0.9269	0.9312	0.9269	0.9212	48.91

Tab. 3: Porovnanie jednotlivých klasifikačných metód

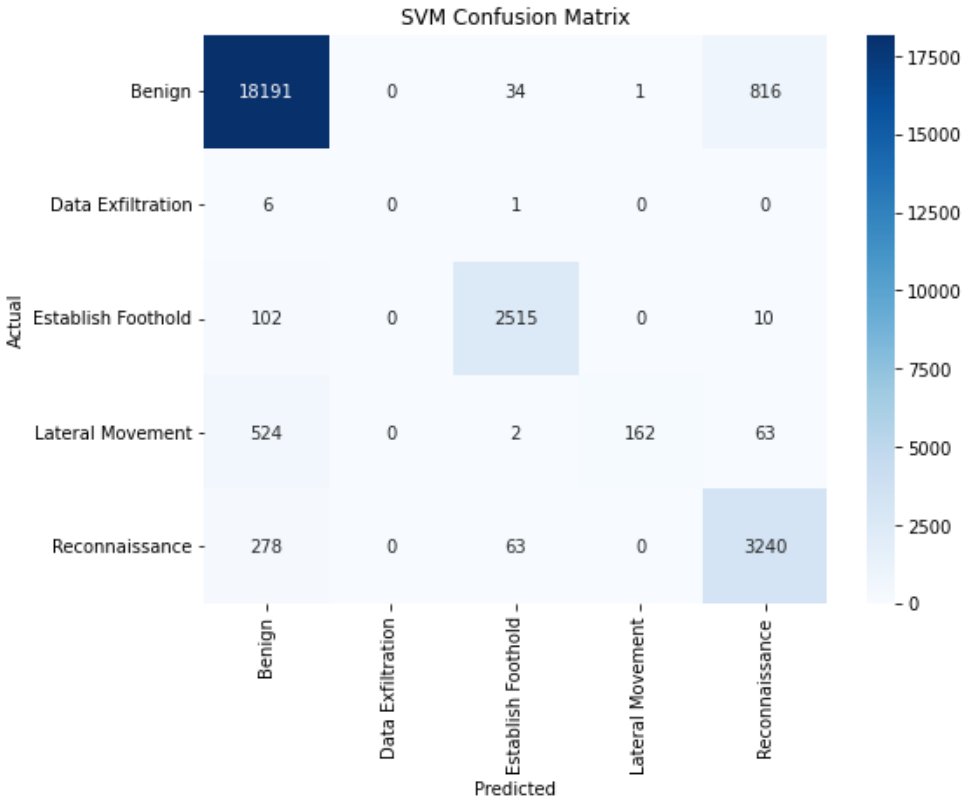
Nasledujúce obrázky zobrazujú konfúznu maticu pre jednotlivé testované metódy.



Obr. 6: Konfúzna matica pre logistickú regresiu



Obr. 7: Konfúzna matica pre náhodný les



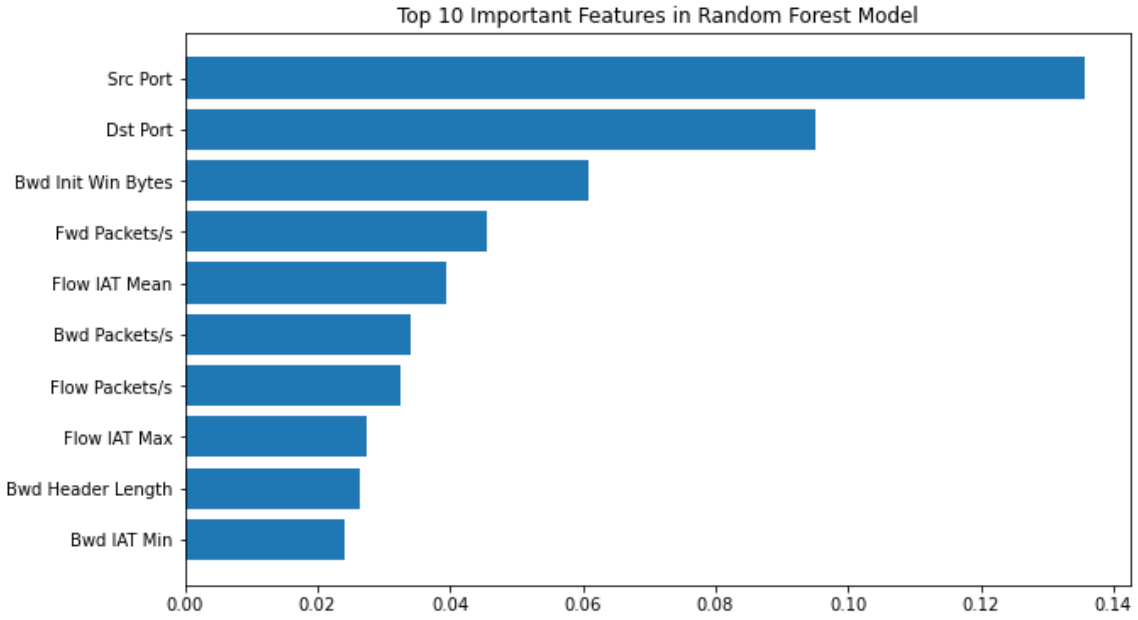
Obr. 8: Konfúzna matica pre SVM

Na nasledujúcom obrázku možno vidieť grafické zobrazenie korelačného koeficientu pre vybrané príznaky v podobe tepelnej mapy, pričom je vidieť, že viacero z nich je vzájomne previazaných.



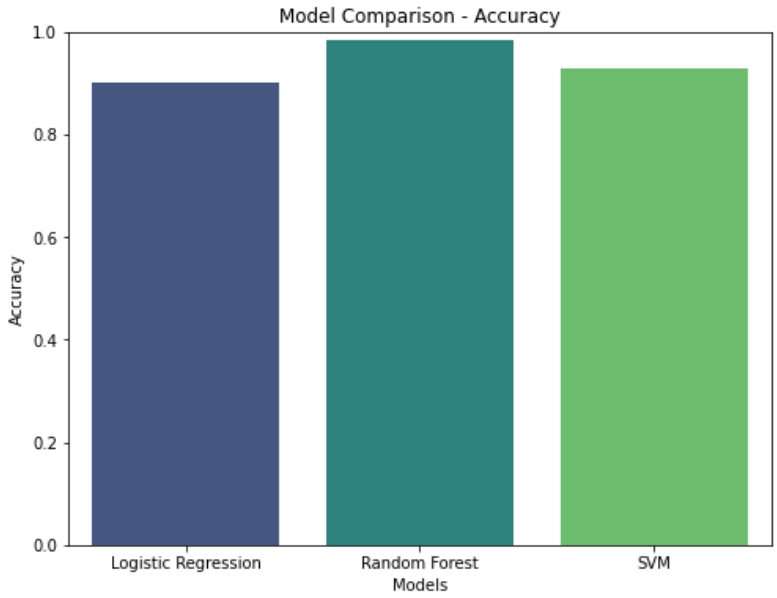
Obr. 9: Heat-mapa korelačného koeficientu pre vybrané príznaky

Na nasledujúcom obrázku je znázornených 10 najrelevantnejších príznakov pre detekciu APT útokov v modeli náhodného lesa vyhodnotených na základe nečistôt (angl. impurity-based).



Obr.10: 10 najdôležitejších príznakov v modeli náhodného lesa

Posledný obrázok vizualizuje úspešnosť (parameter Accuracy) jednotlivých zvolených klasifikačných metód.



Obr. 11: Porovnanie úspešnosti modelov

Zhrnutie

Na základe vykonaných experimentov možno zhodnotiť, že najvhodnejšia metóda pre detekciu pokročilých APT útokov je najvhodnejšia metóda *Random forest*. Na datasete DAPT2020, ktorý je jedným z najkomplexnejších a najčastejšie používaných súborov dát pre detekciu APT útokov, sme dosiahli úspešnosť detekcie 98%. Táto hodnota je takmer identická pre metriky Precision aj Recall, takže je veľmi dobre pokrytá detekcia falošne pozitívnych aj falošne negatívnych detekcií. V poradí druhá najúspešnejšia metóda s úspešnosťou detekcie na úrovni 92% je metóda *SVM*, pričom metóda *logistickej regresie* dosahuje porovnateľnú úspešnosť detekciu s hodnotou ~90%. Z hľadiska času potrebného na testovanie dát dosahujú metódy *Random forest* a *logistická regresia* veľmi porovnateľné výsledky na úrovni desiatok milisekúnd, takže možno nasadiť obe metódy pre real-time aplikácie. *SVM* metóda potrebuje pri testovaní výrazne dlhší čas na detekciu na úrovni desiatok sekúnd, čo predstavuje viac ako 100-násobne dlhší čas.

Na základe vykonaných experimentov možno teda sumarizovať, že najvhodnejšia metóda pre detekciu pokročilých APT útokov je najvhodnejšia metóda **Random Forest.**