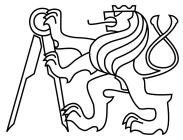
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Analýza leteckých dat a hledání anomálních pasažérů
Student:	Bc. Matúš Tóth
Vedoucí:	doc. Ing. Pavel Kordík, Ph.D.
Studijní program:	Informatika
Studijní obor:	Znalostní inženýrství
Katedra:	Katedra teoretické informatiky
Platnost zadání:	Do konce zimního semestru 2018/19

Pokyny pro vypracování

Prozkoumejte metody detekce anomalit z grafových dat a dat o leteckém provozu. Zpracujte data poskytnutá policií ČR do formy použitelné pro modelování a detekci anomalit. Použijte základní techniky předzpracování dat k přípravě kvalitních atributů. Ve spolupráci s policií formulujte analytické otázky, na které poté odpovíte výsledkem analýz. Analýzu dat proveďte v některém DM nástroji (např. Rapid Miner, nebo h2o.ai). Soustřeďte se zejména na detekci pasažérů podezřelých z pašování lidí, zbraní a chráněných zvířat. Výsledkem budou odpovědi na analytické otázky podpořené datovými reporty.

Seznam odborné literatury

Dodá vedoucí práce.

doc. Ing. Jan Janoušek, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 4. dubna 2017

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra teoretické informatiky

Diplomová práca

Analýza leteckých dát a hľadanie anomálnych pasažierov

Bc. Matúš Tóth

Vedúci práce: Ing. Pavel Kordík, Ph.D.

9. januára 2018

Poďakovanie

V prvom rade by som chcel poďakovať Ing. Pavlovi Kordíkovi, Ph.D. za cenné rady, pomoc a odborné vedenie tejto práce. Ďalej by som chcel poďakovať svojim najbližším a rodine za dôveru a neustálu podporu pri vypracovávaní tejto diplomovej práce.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 9. januára 2018 .....

Odkaz na túto prácu

Tóth, Matúš. *Analýza leteckých dát a hľadanie anomálnych pasažierov.* Diplomová práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

V tejto diplomovej práci sa venujem preskúmaniu možností detekcie anomálií v rôznych typoch dát, spracovaniu leteckých dát poskytnutých Políciou ČR do formy vhodnej pre modelovanie a detekciu spomínaných anomálií. Pre letecké dáta tiež v spolupráci s políciou definujeme analytické otázky, na ktoré následne odpoviem na základe vykonanej analýzy. Analýza bude vykonávaná pomocou pythonovských skriptov a dataminingového nástroja *scikit-learn*.

Klíčová slova Analýza, Detekcia anomálií, Anonymizácia dát, Letecké dáta, Predspracovanie dát, Strojové učenie

Abstract

In this diploma thesis I examine the possibilities of detecting anomalies in different types of data, pre-processing of flight data provided by the Czech Police to a form suitable for modeling and detection of the mentioned anomalies. For flight data, we also define analytical questions in co-operation with the police, which I will then answer according to the results of analysis. The analysis will be performed using Python scripts and the *scikit-learn* datamining tool.

Keywords Analysis, Anomaly detection, Data anonymization, Data pre-processing, Flight data, Machine learning

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

Obranyschopnosť a bezpečnosť štátu a jeho obyvateľov patrí k základným funkciám štátu. Zaisťuje sa tým trvanie a suverenita štátu a je nevyhnutným predpokladom na to, aby občania mohli na území štátu užívať svoje práva a slobody. Je to komplexný pojem zahŕňajúci veľkú množinu rôznorodých činností. Dá sa však rozdeliť na dve hlavné odvetvia a to:

1. Medzinárodná bezpečnosť
2. Vnútroštátna bezpečnosť.

Sem patria aj oblasti, v ktorých je zaistenie bezpečnosti nad sily jednotlivca. Nie je možné individuálne sa brániť proti ozbrojenej agresii, zabrániť šíreniu zbraní hromadného ničenia alebo ovplyvniť ekonomické a environmentálne problémy sveta. Jednotlivec preto dobrovoľne ochranu svojich záujmov posúva na vyššiu organizačnú štruktúru – štát. Ten disponuje vnútornou a vonkajšou suverenitou a prostredníctvom svojich bezpečnostných zložiek (polícia, ozbrojené sily) zaisťuje bezpečnosť svojich občanov. Vnútorná suverenita mu umožňuje vykonávať jurisdikciu v rámci štátneho celku, zatiaľ čo vonkajšia suverenita mu garantuje rovnocenné postavenie v systéme medzinárodných vzťahov a to, že žiadny iný štát nemá právo zasahovať do jeho interných záležitostí.

Medzi oblasti, ktoré musí zastrešovať štát patrí aj oblasť ochrany hraníc. Už samotný pojem ochrany hraníc je nadmieru komplexný, keďže je potrebné identifikovať mnohé druhy hrozieb, od pašovania ľudí, zvierat alebo drog až po nelegálnu imigráciu a terorizmus. Pri ochrane hraníc pojednávame rôzne možnosti dopravy. Pri automobilovej a vlakovej je potrebná fyzická kontrola hraníc a kontrola jednotlivých ľudí. Keďže však letecká doprava poskytuje bohaté informácie o pasažieroch dostupné ešte pred príletom, je možné skúsiť automatizovať identifikáciu podozrivých pasažierov na základe týchto údajov.

Kapitola 1

Základ

1.1 Monitorovanie leteckej dopravy

Letecký dopravce je povinen za účelem zdokonalení hraničních kontrol a boje proti nedovolenému přistěhovalectví předávat útvaru Policie České republiky údaje o cestujících, kteří překročí vnější hranici (pouze přílet) na vybraných pravidelných linkách, u charterových letů na vyžádání.

- směrnice č. 2004/82/ES
- zákon č. 49/1997 Sb., o civilním letectví
- Interní akty řízení PP a ŘSCP

Ako vidíme, letecký dopravca je zo zákona povinný poskytovať dáta o cestujúcich. Tieto poskytované údaje sú (§ 69 z.č. 4/1997):

1. číslo a typ použitého cestovného dokladu
2. štátna príslušnosť
3. meno (mená) a priezvisko
4. dátum narodenia
5. hraničný prechod vstupu na územie členských štátov
6. kódové číslo letu
7. čas odletu a príletu
8. celkový počet osôb prepravovaných uvedeným letom
9. počiatočné miesto nástupu na palubu

1.1.1 Informačný systém *OBZOR*

Tieto dáta tiež musia byť istým spôsobom organizované. Preto Polícia Českej republiky prišla s informačným systémom *OBZOR*, ktorý plne prepojuje leteckých dopravcov s Políciou pomocou siete leteckej dopravy *SITA*. Bol uvedený do prevádzky 1.7.2012 a posiela doň svoje dáta 30 leteckých spoločností na 73 leteckých spojoch.

Medzi jeho funkcie patria:

- vyhodnotenie formátu API správy
- vyhodnotenie správnosti jej obsahu
- základné analýzy rizík
- vykonanie previerky osôb
- prehľadové zobrazenie výsledkov
- štatistické a analytické funkcie (vytváranie profilov cestujúcich)

Ako vidíme, tento informačný systém ponúka rôzne možnosti prehľadu. Ďalšou možnosťou je zobraziť profil cestujúceho. V tomto profile sú zahrnuté všetky lety tohoto pasažiera a aj prípadné problémy.

Isté profily sú rizikové už na základe národnosti cestujúceho, dátumu narodenia, miestom odletu alebo nejakou kombináciou týchto vlastností, *OBZOR* tiež ponúka vytvorenie istých súborov vlastností a cestujúci, ktorí tieto vlastnosti spĺňajú sú označení na ďalšie preskúmanie.

Jednou z najdôležitejších funkcií pre nás je však export dát, pre automatické spracovanie pomocou externých nástrojov.

Kapitola 2

Zadanie

Na základe týchto požiadavkov a momentálnych trendov bolo vytvorené zadanie tejto práce.

2.1 Vstupné dáta

Našimi vstupmi sú dáta z *OBZOR*u 1.1.1. Dáta z obzoru sú organizované do adresárovej štruktúry podľa roku a mesiaca príletu. Jednotlivé lety sú uložené v separátnych súboroch (.csv, .xlsx), kde každý záznam odpovedá jednému pasažierovi. Každý zo záznamov pozostáva z niekoľkých atribútov jednotlivca a to:

1. *FlightNumber* - Číslo letu. Keďže sa jedná o kombináciu písmen a čísel o obmedzenej dĺžke, nie je unikátnym identifikátorom letu, nieto ešte pasažiera.

2. *ScheduledArrival* - Plánovaný čas a dátum príletu. Tiež sa nemusí jednať o jednoznačnú identifikáciu letu, keďže v jeden čas môže pristávať aj viac letov.

3. *Nationality* - Národnosť pasažiera. Zakódovaná v trojpísmenových skratkách štátu (CZE, SVK atp.).

4. *Surname* - Priezvisko pasažiera.

5. *Names* - Všetky zvyšné mená pasažiera.

6. *BirthDate* - Dátum narodenia pasažiera.

7. *Sex* - Pohlavie pasažiera. Nadobúda hodnôt - M pre muža, F pre ženu a U, ktoré nie je definované.

8. *DocumentType* - Pri odlete sa udáva identifikačný dokument pasažiera. Malo by sa jednať o buď pas alebo občiansky preukaz (ak je pasažier občanom členského štátu európskej únie).

9. *DocumentIssued* - Štát, v ktorom bol daný dokument vydaný. Zakódovaný v trojpísmenových skratkách štátu (CZE, SVK atp.).

10. *DocumentNumber* - Číslo tohoto dokumentu.

11. *FlightFrom* - Kód letiska, z ktorého let odlieta.

12. *FlightTo* - Kód letiska, na ktoré let prilieta. Spolu s *FlightNumber, ScheduledArrival, FlightFrom* môže byť použitý ako jednoznačná identifikácia letu.

13. *Reservation* - Ak má záznam aj tento atribút, tak sa jedná o rezerváciu dopredu. Jedná sa o kód rezervácie. Ak ho dvaja pasažieri zdieľajú, letia títo pasažieri spolu (na jednu rezerváciu).

14. *HitType* - Atribút označujúci jednotlivé hrozby. Ak je hodnota tohto atribútu 1 tak sa jedná o normálneho pasažiera, inak nie. Jedná sa o označenie na základe porovnania s databázou už známych nebezpečných ľudí. Nie všetky záznamy obsahujú tento atribút.

2.1.1 Nekonzistencie

Po podrobnom preskúmaní som narazil na isté problémy v týchto dátových súboroch.

2.1.1.1 Formáty súborov

Prvým problémom boli rozdielne formáty súborov, v ktorých sú dáta pasažierov uložené. Jedným z nich je formát .xlsx, ktorý je štandardným formátom programu Microsoft Excel. Druhým formátom je .csv (comma-separated values). Vrámci .csv súborov však tiež dochádza k nekonzistenciám a to v spôsobe oddelenia jednotlivých záznamov. Prvým je oddelenie záznamov pomocou bodkočiarky, druhým pomocou čiarky. Taktiež v niektorých .csv súboroch, v ktorých sú záznamy oddelené pomocou bodkočiarky sa nachádzajú záznamy obsahujúce čiarky (nie ako oddeľovače, ale ako hodnoty atribútov), čo znemožňuje jednoduchú konverziu medzi týmito formátmi.

2.1.1.2 Dátumy

Ďalším problémom boli dátumy. Ako pre atribút *ScheduledArrival*, tak aj pre *BirthDate* sa dátumy vyskytovali v 7 rôznych formátoch.

- d.m.Y H:M
- Y-m-d H:M:S
- Y-m-d H:M
- Y-m-d
- Y/m/d H:M:S
- Y/m/d H:M
- Y/m/d

2.1.1.3 Atribúty HeadGUID a BodyUID

Niektoré z letov osahujú ešte pred atribútom *FlightNumber* atribúty *HeadGUID* a *BodyUID*.

- *HeadGUID* - jedná sa o alfanumerický atribút (jednou z hodnôt, ktoré nadobúda je napríklad *74cf9b88-dcc3-40f2-9960-44cc88c76a54*)
- *BodyUID* - jedná sa o numerický atribút.

Ani po konzultácii s poskytovateľmi dát nie je jasný význam týchto dvoch atribútov, preto ich považujeme za nekonzistenciu.

2.1.1.4 Identifikácia pasažiera

Pri identifikácii pasažiera dochádza k viacerým nekonzistenciám.

Atribút Nationality Prvým problémom je, že pri niektorých záznamoch chýba atribút reprezentujúci národnosť pasažiera.

Atribút Names Ďalším problémom pri jednoznačnej identifikácii pasažiera je pri jeho menách (okrem priezviska). Pri väčšine záznamov sú jednotlivé mená oddelené medzerou, čo však nie je pravdou pri všetkých záznamoch. Pri niektorých nie sú tieto mená oddelené vôbec, čo môže znemožniť identifikáciu.

Atribút Sex Atribút sex hovorí o pohlaví pasažiera. Pri niektorých záznamoch však chýba.

Atribút DocumentType Tento atribút má pojednávať o type dokumentu, ktorým sa pasažier preukazuje. Malo by sa jednať o občiansky preukaz, alebo pas. Avšak, nie je to tak, keďže tento atribút pri každom zázname nadobúda len jednej hodnoty a to hodnoty P (pas). Kvôli praktickým dôvodom sa teda nemôžeme spoliehať na informačnú hodnotu tohoto atribútu. Pri niektorých záznamoch sa tento atribút zase nenachádza.

Atribút DocumentIssued Tento atribút hovorí o tom, v akom štáte je tento identifikačný dokument vydaný. Malo by sa teda jednať konkrétne o trojpísmenovú skratku tohoto štátu. Sú však aj záznamy, ktoré tento atribút nemajú, ale väčšinou majú určený typ dokumentu ako pas.

Atribút DocumentNumber Taktiež číslo dokumentu je niekedy nekonzistentné. Malo by sa jednať o alfanumerickú hodnotu, ale pri niektorých záznamoch tento atribút nadobúda hodnôt desatinných čísel. Naviac, pri niektorých záznamoch tento atribút zase chýba. Toto teda tiež považujeme za nekonzistenciu v dátach.

2.1.2 Atribút Reservation

Atribút Reservation je atribút, ktorý chýba pri najväčšom množstve záznamov. Toto však nie je chybou. Jedná sa o informáciu, že daný pasažier nemal rezerváciu. Ďalším problémom s týmto atribútom je, že pri niektorých letoch namiesto toho aby pasažierom ponechali chýbajúci atribút, nadobúda tento Reservation kladných celých čísel (vždy rôzna hodnota). V niektorých prípadoch zase nadobúda hodnôt desatinných čísel (vo formáte 1,10693E+11).

2.1.3 Atribút HitType

Aj keď sa môže javiť, že sa jedná o smerodajný atribút pri identifikácii potenciálnych hrozieb, nie je to tak. Tento atribút sa nevyskytuje pri mnohých záznamoch a keď sa vyskytuje, nemôžeme sa spoliehať na jeho pravdivosť. Napríklad vieme, že ak tento atribút nadobúda hodnoty 1, malo by sa jednať o bežného a bezpečného pasažiera. Ak však nenadobúda 1, mal by nadobúdať hodnotu 2 alebo 3 (tak sú označené známe hrozby). V poskytnutých dátach sa však vyskytujú celé lety, čo majú nastavený *HitType* na hodnotu mimo tejto množiny známych označení.

Pre tieto dôvody neprítomnosť atribútu a nadobúdanie neznámych hodnôt považujem za nekonzistenciu v dátach.

2.2 Požadované výstupy

V tejto sekcii analyzujem ciele tejto práce.

2.2.1 Spracovanie dát

V predchádzajúcej kapitole som opisoval mimo iné aj nekonzistencie v dátach. Tieto nekonzistencie spôsobujú, že dáta nie sú vhodné na automatické spracovanie a tým pádom ani vhodné na strojové učenie a tiež na detekciu anomálií. Prvou úlohou je teda analyzovať spôsoby, akými je možné zbaviť sa opísaných nekonzistencií.

Ďalej je potrebné dáta dostať do najvhodnejšej formy na automatické spracovanie. Keďže pôvodne dáta boli rozdelené do štruktúry podľa dátumu príletu, už len prístup k jednotlivým letom a teda aj k pasažierom je problematický.

Ďalším problémom je formát súborov, v ktorých sa dáta nachádzajú. Je potrebné zvoliť jednotný formát.

2.2.2 Detekcia anomálií

Ďalším cieľom je preskúmať v týchto dátach možnosti detekcie anomálií, analyzovať vhodnosť jednotlivých techník a prípadne demonštrovať tieto techniky na dátach.

2.2.3 Analytické otázky

Posledným bodom je odpovedať na analytické otázky zadané políciou Českej republiky. Týmito otázkami sú:

1. Je možné na základe poskytnutých dát zachytiť atribúty nebezpečného pasažiera? (označiť všetkých nebezpečných pasažierov)
   - Ak áno, s akou presnosťou vieme určiť týchto pasažierov?
   - Dokážeme vymodelovať „bezpečného pasažiera“?
2. Dajú sa určiť na základe týchto dát celé lety (alebo letiská), ktoré majú oproti ostatným vyššiu pravdepodobnosť, že v nich budú nebezpeční pasažieri?
3. Existujú ľudia, čo stále cestujú spolu v lietadle, ale nikdy nie na jednu rezerváciu?

Na tieto otázky bude možné odpovedať po vykonaní zvyšných analýz. Odpoveď bude podložená analýzou a experimentami v jednom z data miningových nástrojov.

Kapitola 3

Teoretický základ

Na základe analýzy vstupov a odpovedajúcich výstupov z predchádzajúcej kapitoly je potrebné mať istý teoretický základ aby sme dáta vedeli správnym spôsobom spracovať a výsledky interpretovať.

3.1 Predspracovanie dát

Predspracovanie údajov je dôležitým krokom v procese data miningu. Fráza „garbage in, garbage out“ je obzvlášť uplatniteľná na data mining a strojové učenie. Metódy získavania údajov sú často slabo kontrolované, čo vedie k získaniu hodnôt mimo validný rozsah (napr. Plat: -100), nemožných kombinácií atribútov (napr. Pohlavie: Muž, Tehotný: Áno), chýbajúcich hodnotôt atď. Analýza dát, ktoré neboli podrobne skontrolované, či obsahujú spomenuté problémy môže priniesť zavádzajúce výsledky. Z toho vyplýva, že preskúmanie a zaistenie kvality dát je potrebné uskutočniť pred analýzou [2].

3.1.1 Nahrádzanie chýbajúceho atribútu

V bežných dátach sa často môže stať, že niektoré záznamy neobsahujú všetky z atribútov. Toto môže nastať z rôznych príčin (porucha jedného zo senzorov na sonde, chybujúci ľudský faktor, atp.). Tieto defekty však musia byť odhalené a vrámci predspracovania dát by mala byť zvolená jedna z možností ako sa s nekonzistenciami vysporiadať. Tieto techniky zohľadňujú dôležitosť informácie, že atribút chýba.

1. **Nespraviť nič** - Prvou možnosťou je ponechať atribút chýbajúci. Zachováme tak informáciu, že niečo pri tomto zázname nebolo v poriadku. Nevýhodou tohoto prístupu je, že mnohé techniky učenia sa nevedia vysporiadať s chýbajúcim atribútom.

2. **Vynechať záznam**- Druhou možnosťou je celý záznam zmazať. Takto prídeme nielen o informáciu, že atribút chýbal, ale aj o ostatné (nechýbajúce atribúty). Tento prístup je vhodný, ak máme veľké množstvo záznamov a len nebatateľné percento z nich má chýbajúci nejaký z atribútov. Nevýhodou je, že môžeme prichádzať o cenné informácie.

3. **Nahradenie priemerom** - Ďalšou možnosťou je chýbajúci atribút nahradiť priemerom, mediánom alebo inou blízkou hodnotou (ak daný atribút poskytuje možnosť priemerovania - numerické atribúty), alebo hodnotou, ktorú atribút najčastejšie nadobúda pri záznamoch, kde nechýba. Takto sa síce zbavíme nekonzistencie, ale zase prídeme o informáciu, že atribút chýbal a navyše zo záznamov, ktoré boli do veľkej miery odlišné od ostatných sa môžu stať záznamy, ktoré nie sú odlišné badateľným spôsobom. Táto technika je vhodná ak môžeme o dátach predpokladať, že sa vyskytujú v zhlukoch a takéto vyhladenie nespôsobí žiadny problém.

4. **Nahradenie hodnotou atribútu najbližšieho suseda** - Možnosťou nahradenia chýbajúcej hodnoty je tiež nájdenie si najbližšieho suseda (najpodobnejší záznam tomu, ktorému atribút chýba) alebo k najbližších susedov a nahradenia chýbajúceho atribútu hodnotou, ktorá sa medzi týmito k susedmi vyskytuje najviac. Takto zase prichádzame o informáciu, že atribút chýbal.

5. **Nahradenie význačnou hodnotou** - Táto metóda spočíva v nahradení atribútu istou hodnotou, ktorú tento atribút nenadobúda v žiadnom inom prípade (napríklad pre počty je vhodné zvoliť -1, keďže počet nadobúda hodnoty prirodzených čísel). Takto nestratíme ani záznam, ani informáciu o tom, že atribút chýbal a ani nemôže nastať vyhladenie v dátach. Potrebujeme však isté znalosti o dátach, ktoré majú aby sme zvolili význačnú hodnotu správne.

3.1.2 Normalizácia dát

Normalizácia dát je proces predspracovania dát. Pomocou tejto normalizácie upravujeme (štandardizujeme) rozsah premenných alebo vlastností dát.

Keďže rozsah hodnôt nespracovaných údajov sa môže značne líšiť, v niektorých algoritmoch strojového učenia funkcie nemusia fungovať správne bez normalizácie. Napríklad väčšina klasifikátorov vypočíta vzdialenosť medzi dvoma bodmi podľa istej miery vzdialenosti (mnohokrát euklidovská). Ak niektorý z atribútov má veľký rozptyl hodnôt, vzdialenosť bude značne ovplyvnená práve týmto atribútom. Rozptyl všetkých atribútov by sa mal normalizovať tak, aby každý z nich prispel ku konečnej vzdialenosti rovnako.

Techniky normalizácie dát:

- Min-Max normalizácia. Tento druh normalizácie spočíva v naškálovaní atribútu do istého intervalu (min - max). Štandardným intervalom je interval $[0, 1]$, kde normalizovanú hodnotu atribútu získame ako

$$x' = \frac{x - min(x)}{max(x) - min(x)}.$$

  Tento spôsob je jednoducho rozšíriteľný na akýkoľvek interval $[a, b]$ a to spôsobom:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \cdot (b - a) + a$$

- Desatinné škálovanie (decimal scaling). Jedná sa o normalizáciu takým spôsobom, že každá hodnota daného atribútu sa vynásobí rovnakou celočíselnou mocninou 10.
- Štandardizácia každú hodnotu atribútu upraví spôsobom: $x' = \frac{x - \bar{x}}{\sigma}$, kde $\bar{x}$ je stredná hodnota atribútu a σ jeho štandardná odchýlka.
- Eliminácia odľahlých hodnôt. Táto technika spočíva v nájdení odľahlých hodnôt a následnom vymazaní alebo nahradení týchto hodnôt.

3.1.3 Anonymizácia dát

Anonymizácia dát je úprava dát za účelom ochrany súkromia. Je to proces, pri ktorom sú z data setov zašifrované alebo odstránené informácie, ktoré vedú k jednoznačnej identifikácii človeka.

Anonymizácia údajov bola definovaná ako „technológia, ktorá nezvratne tranformuje čisté textové údaje na nečitateľnú podobu použitím hashovacích techník neumožňujúcich spätné zrekonštruovanie pôvodných dát (jednosmerné hashe) a šifrovacích techník, ktorých dešifrovací kľúč bol zahodený.“[3] Anonymizácia údajov umožňuje prenos informácií, napríklad medzi dvoma oddeleniami v rámci jednej spoločnosti alebo medzi dvoma spoločnosťami, pričom sa znižuje riziko neúmyselného zverejnenia. V určitých prostrediach sa anonymizujú dáta spôsobom, ktorý umožňuje následné hodnotenie a analýzu pôvodných dát.

V súvislosti s lekárskymi záznamami sú anonymizované údaje také údaje, z ktorých pacient nemôže byť identifikovaný príjemcom informácií. Meno, adresa a iné musia byť odstránené spolu s akoukoľvek inou informáciou, ktorá v spojení s ostatnými údajmi uchovávanými alebo poskytnutými príjemcovi môže identifikovať pacienta.[4]

De-anonymizácia je opačný proces, pri ktorom sú anonymné údaje prepojené s inými zdrojmi údajov pre opätovné identifikovanie anonymného zdroja údajov.[5]

3.1.3.1 Jenkinsove hashovacie funkcie

Príkladom anonymizačných hashovacích funkcií sú Jenkinsove hashovacie funkcie. Je súbor nekryptografických hashovacích funkcií pre viacbajtové kľúče navrhnutých Bobom Jenkinsom. Prvá z nich bola publikovaná v roku 1997.

1. one-at-a-time. Táto hashovacia funkcia má nasledovný zdrojový kód[6]:

```
uint32_t one_at_a_time_hash(const uint8_t* key, size_t length)
{
  size_t i = 0;
  uint32_t hash = 0;
  while (i != length) {
    hash += key[i++];
    hash += hash << 10;
    hash ^= hash >> 6;
  }
  hash += hash << 3;
  hash ^= hash >> 11;
  hash += hash << 15;
  return hash;
}
```

2. lookup2. Hashovacia funkcia lookup2 bola nasledovníkom one-at-a-time (vychádza z jeho zdrojového kódu) a v súčasnej dobe je nevyužívaná kvôli existencii ďalších nasledovníkov.

3. lookup3. Táto hashovacia funkcia spracováva vstup po 12 bajtových kusoch. Jej využitie je výhodné ak sa chceme sústrediť na rýchlosť a nie na jednoduchosť hashovania. Využíva sa pri hashovaní veľkých kľúčov.[7]

4. SpookyHash. Publikovaná v roku 2011, jedná sa o novú 128-bitovú hashovaciu funkciu, ktorá je ešte rýchlejšia ako lookup3.[8]

3.2 Detekcia anomálií

Detekcia anomálií predstavuje problém nájdenia vzorov v dátach, ktoré nedosahujú očakávané správanie. Tieto nevyhovujúce vzory sú často označované ako anomálie alebo odľahlé hodnoty. Detekcia anomálií nachádza rozsiahle uplatnenie v širokej škále aplikácií, ako je detekcia chýb v bezpečnostných systémoch, vojenský dohľad nad nepriateľskými aktivitami alebo tiež detekcia anomálií medzi leteckými pasažiermi.

3.2.1 Čo sú to anomálie?

Podľa [1], anomálie sú vzory v dátach, ktoré nezodpovedajú normálnemu chovaniu. Možno ich spôsobiť v dátach rôznymi spôsobmi, ako je škodlivá činnosť, napríklad podvody s kreditnými kartami alebo porucha systému. Všetky tieto

neštandardné vzory majú istú hodnotu a to „zaujímavosť“ alebo význam v reálnom živote, čo je hlavným rysom detekcie anomálií.

3.2.2 Problematickosť domény

Na abstraktnej úrovni, anomália je definovaná ako vzor, ktorý nie je v súlade s normálnym chovaním. Jednoduchým prístupom pre detekciu anomálií je preto vymedziť rozsah reprezentujúci normálne správanie a každé pozorovanie/záznam, ktoré nepatrí do tohto rozsahu označiť ako anomáliu. Avšak podľa [1], niekoľko faktorov spôsobuje, že tento zdanlivo jednoduchý prístup sa stáva náročným:

- Definovanie tejto oblasti, ktorá zahŕňa všetko možné normálne správanie je veľmi ťažké. Taktiež hranica medzi normálnym a abnormálnym chovaním často nie je presná.

- Keď sú anomálie výsledkom škodlivých akcií, útočníci sa snažia javiť ako bežní užívatelia, preto aj ich akcie sú často veľmi podobné akciám bežných užívateľov, čím sa zase sťažuje detekcia týchto útokov.

- V mnohých doménach sa toto normálne správanie zase časom vyvíja a čo bolo normálnym správaním v minulosti, už v budúcnosti normálnym správaním byť nemusí.

- Presný pojem anomálie sa líši v rôznych aplikačných oblastiach. Napríklad, v medicínskej oblasti už malá odchýlka od normálu (napríklad kolísanie telesnej teploty) môže byť anomália, zatiaľ čo podobná odchylka na burze cenných papierov (napríklad výkyvy v hodnote akcie) by mohla byť považovaná za normálnu. Vyvinutie jednej stratégie pre detekciu anomálií teda nemusí byť aplikovateľná na inú doménu.

- Dostupnosť označených dát pre učenie a validáciu modelov je tiež často problémom.

- Dáta často obsahujú šum, ktorý má tendenciu byť podobný reálnym anomáliám a preto je ťažké ich rozlíšiť a odstrániť.

Vzhľadom k vyššie uvedeným problémom, je problém detekcie anomálií vo svojej najvšeobecnejšej forme obtiažne vyriešiť. V skutočnosti väčšina súčasných techník detekcie anomálií rieši jednu konkrétnu formuláciu problému. Formulácia je vyvolaná rôznymi faktormi, ako je povaha dát, dostupnosť označených dát, typu anomálie, ktorú sa snažíme detekovať, atď. Tieto faktory sú určené doménou v ktorej anomálie hľadáme. Pri riešení tohto problému sa využívajú poznatky z rozmanitých odborov, ako je štatistika, strojové učenie a data mining.

3.2.3 Rôzne aspekty problému detekcie anomálií

Ako som už spomenul, konkrétna formulácia problému je daná niekoľkými faktormi, ako je povaha vstupných dát a dostupnosť (či nedostupnosť) značených dát. [1]

3.2.3.1 Povaha vstupných dát

Kľúčovým aspektom akejkoľvek techniky detekcie anomálií je povaha vstupných dát. Vstup je obvykle kolekcia inštancií dát. Každá inštancia dát je označená sadou atribútov/dimenzií. Atribúty môžu byť rôznych druhov (numerické, binárne, atď). Povaha atribútov určuje použiteľnost techník na detekciu anomálií. Napríklad pre techniky založené na metóde najbližšieho suseda potrebujeme atribúty pre ktoré vieme určiť vzdialenosť medzi dvoma inštanciami/záznamami.

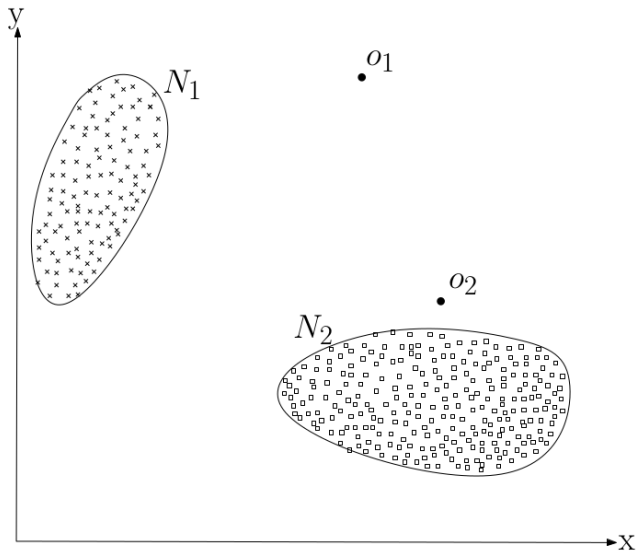
Vstupné dáta môžu tiež byť klasifikované na základe vzťahu medzi nimi. Väčšina existujúcich techník na detekciu anomálií funguje na základe obdržaných alebo nameraných dát (alebo bodových údajov), v ktorých sa nepredpokladá žiadny vzťah medzi inštanciami dát. Všeobecne však platí, že inštancie dát môžu byť vo vzájomnom vzťahu. Niektoré príklady sú dáta sekvencií, priestorové údaje a grafové dáta. V sekvenčných dátach sú jednotlivé inštancie zoradené, napríklad na základe času (časové postupnosti), sekvencie genómov a iné. V priestorových dátach, každá inštancia dát sa vzťahuje k jej susedným inštanciách. Keď priestorové dáta majú aj časovú (sekvenčnú) zložku sú označované ako časopriestorové dáta, napríklad dáta o klíme, alebo letecké dáta. V grafových dátach sú inštancie reprezentované ako vrcholy v grafe a sú prepojené s ďalšími vrcholmi hranami.

3.2.3.2 Druhy anomálií

Dôležitým aspektom techniky detekcie anomálií je povaha požadovanej anomálie. Anomálie možno zaradiť do troch kategórií:

Bodové anomálie Ak jednotlivé inštancie dát môžu byť považované za anomálne vzhľadom ku zvyšku dát, potom je táto inštancia bodovou anomáliou. Jedná sa o najjednoduchší typ anomálie. Ako príklad z reálneho života zoberme detekciu podvodov s kreditnými kartami. Súbor dát obsahuje transakcie kreditnou kartou. Predpokladajme, že dáta sú definované použitím iba jedného atribútu: zaplatená suma. Transakcie, pre ktoré je táto suma veľmi vysoká v porovnaní s ostatnými výdavkami bude klasifikovaná ako bodová anomália.

Kontextové anomálie Ak je inštancia dát anomálnou v špecifickom kontexte (inak nie), potom sa nazýva kontextuálna anomália (tiež označovaný ako podmienené anomálie). Kontext je tvorený štruktúrou v súbore dát a musí



Obr. 3.1: Bodové anomálie[1]

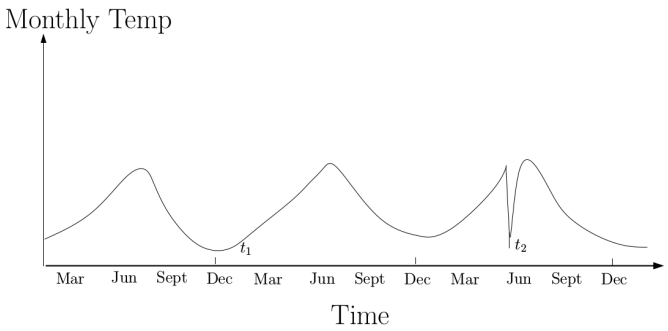
byť zadaný ako súčasť formulácie problému. Každá inštancia dát je definovaná dvoma typmi atribútov:

1. Kontextové atribúty určujú kontext (alebo susednosť) pre túto inštanciu. Napríklad v priestorových dátach, zemepisná dážka a šírka sú kontextové atribúty. V dátach časových postupností, čas je kontextový atribút, ktorý určuje pozíciu jednej inštancie vrámci sekvencie.

2. Behaviorálne atribúty definujú nekontextuálne charakteristiky inštancie. Napríklad v priestorových dátach priemerných zrážok z celého sveta, je množstvo zrážok v akomkoľvek mieste behaviorálny atribút.

Kontextuálne atribúty môžu byť:

- Priestorové - máme polohu a tým pádom aj priestorové okolie [9].
- Grafové - máme hrany, ktoré spájajú jednotlivé uzly (inštancie), čím sa zase určuje okolie.
- Sekvenčné - atribúty, ktoré určujú pozíciu v postupnosti. Jedná sa napríklad o časové rady [10][11].
- Profilové - sú to atribúty, ktoré zaraďujú inštancie do skupín (profilovanie), vrámci ktorých sa potom testuje anomálnosť.



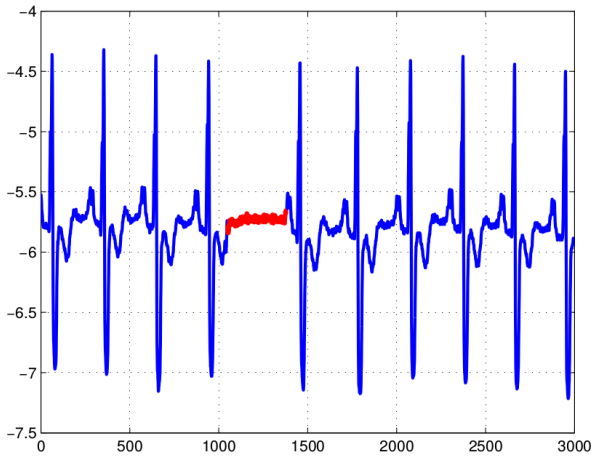
Obr. 3.2: Kontextová anomália[1]

Anomálne správanie je stanovené použitím hodnôt pre behaviorálne atribúty v určitom kontexte. Inštancia dát môže byť kontextuálnou anomáliou v danom kontexte, ale identická inštancia dát (pokiaľ ide o behaviorálne atribúty) by mohla byť považovaná za normálnu v inom kontexte. Táto vlastnosť slúži k identifikácii kontextových a behaviorálnych atribútov pre techniku detekcie kontextových anomálií.

Kontextové anomálie sú najčastejšie skúmané v časových a priestorových dátach. Voľba použitia techniky detekcie kontextových anomálií závisí od zmysluplnosti kontextuálnych anomálií v doméne cieľovej aplikácie. ďalším kľúčovým faktorom je dostupnosť kontextových atribútov. V niekoľkých prípadoch je definícia kontextu jednoduchá a teda aj použitie metód pre detekciu kontextových anomálií dáva zmysel. V iných prípadoch môže byť definovanie kontextu obtiažne, čo znemožňuje použitie mnohých metód.

Kolektívne anomálie Ak kolekcia súvisiacich inštancií je anomálna vzhľadom na celý súbor dát, nazýva sa kolektívna anomália. Individuálne inštancie dát v kolektívnej anomálii nemusia byť anomálie samy o sebe, ale ich výskyt spolu ako kolekcia je abnormálny.

Je potrebné poznamenať, že zatiaľ čo bodová anomália sa môže objaviť v každom súbore dát, ku kolektívnym anomáliám môže dôjsť iba v dátach, ktorých inštancie spolu súvisia. Na druhú stranu, výskyt kontextových anomálií závisí od dostupnosti kontextových atribútov v dátach. Bodová alebo kolektívna anomália môže byť tiež kontextovou ak je analyzovaná vzhľadom na kontext. Problém detekcie bodových alebo kolektívnych anomálií môže byť transformovaný na detekciu kontextových anomálií začlenením kontextových atribútov.



Obr. 3.3: Kolektívna anomália[1]

3.2.3.3 Označenie dát

Označenie dát hovorí o tom, či inštancia je normálna alebo anomália. Avšak získanie značených dát, ktoré sú presnou reprezentáciou všetkých typov chovania je často nemožné. Značenie sa často vykonáva ručne a preto sa vyžaduje značná snaha na získanie značených dát pre trénovanie. Zvyčajne je obtiažnejšie získať značené dáta pre všetky typy anomálií ako pre normálne chovanie.

Navyše anomálne správanie má často dynamický charakter, môžu vzniknúť nové typy anomálií, pre ktoré nie sú k dispozícii žiadne značené trénovacie dáta. V niektorých prípadoch, ako je napríklad bezpečnosť letovej prevádzky, anomálne prípady by mohli ústiť do katastrofických udalostí a preto budú veľmi vzácne. Na základe rozsahu, v akom sú k dispozícii značené dáta, môže detekcia anomálií prebiehať nasledovnými spôsobmi:

Supervised detekcia anomálií Techniky natrénované s učiteľom predpokladajú dostupnosť trénovacích dát, ktoré obsahujú inštancie pre bežné ako aj anomálne triedy. Typický prístup v takýchto prípadoch je vybudovanie prediktívneho modelu pre klasifikáciu normálnych vs. anomálnych inštancií. Akúkoľvek inštanciu dát je potom možné pomocou tohto modelu klasifikovať. Existujú dva hlavné problémy ktoré vznikajú v supervised detekcii anomálií. Po prvé, anomálne prípady sú ďaleko menej frekventované v porovnaní s bežnými prípadmi v dátach pre trénovanie modelu [12].

Po druhé, získanie presných a reprezentatívnych označení, najmä pre triedu anomálií je zvyčajne náročné. Niekoľko techník bolo navrhnutých tak, aby vkladali umelé anomálie medzi normálne dáta pre získanie obsiahlejšieho trénovacieho setu dát [13][14].

Semi-supervised detekcia anomálií Semi-supervised detekcia anomálií znamená detekovať anomálie, ak máme označené len normálne inštancie. Keďže tieto techniky nepotrebujú označenie anomálnej triedy, sú všeobecne viac uplatniteľné. Typickým prístupom týchto metód je vytvoriť model reprezentujúci normálne dáta a tento model následne použiť na identifikáciu anomálií (ktoré tomuto modelu neodpovedajú). Naopak modely natrénované len na anomálnych inštanciách sú neobvyklé, keďže je obtiažne zachytiť každý možný druh anomálie.

Unsupervised detekcia anomálií Tieto techniky nevyužívajú trénovacie dáta pre žiadnu z tried a teda sú použiteľné najviac. Metódy v tejto kategórií predpokladajú, že normálne inštancie sú ďaleko viac frekventované ako anomálne v testovacích dátach (inak by anomálne inštancie mohli byť považované za druh normálneho chovania a teda detekcia by neprebiehala správne).

3.2.3.4 Výstup detekcie anomálií

Dôležitým aspektom detekcie je tiež požadovaný výstup, ktorým sú anomálie identifikované. Typicky sa jedná o dva typy výstupov:

Skóre Tieto techniky priraďujú každej inštancii z testovacieho data setu isté skóre, ktoré určuje mieru anomálnosti. Výstupom je teda ohodnotený zoznam anomálií. Za anomálie môžeme označiť zvolené množstvo inštancií s najvyšším anomálnym skóre, alebo zvoliť istú hranicu skóre a označíť za anomálne všetky inštancie, ktoré dosiahli vyššie skóre.

Označovanie Techniky využívajúce označovanie (za normálnu alebo anomálnu inštanciu) priradzujú každej inštancii „nálepku“.

Techniky využívajúce skóre umožňujú analytikovi priamo ovplyvňovať citlivosť detekcie anomálií. Na druhú stranu označovacie metódy neposkytujú možnosť túto citlivosť ovplyvniť priamo, ale cez nastavovanie jednotlivých parametrov vrámci týchto metód.

3.2.4 Detekcia anomálií založená na klasifikácii

Používa sa vo dvoch krokoch a to naučenie modelu na označených dátach (trénovanie) a následnej klasifikácii inštancií, o ktorých chceme zistiť či sú anomáliou alebo nie (testovanie)[13]. Pri tomto prístupe predpokladáme, že model dokážeme naučiť na základe zadaného priestoru.

Na základe počtu „nálepiek“ rozdeľujeme techniky na *one-class* a *multi-class* detekcie anomálií. Ako pri *one-class* detekcii máme len jednu triedu pre normálne dáta, tak v *multi-class* máme viac druhov normálneho správania a preto vieme rozoznávať medzi nimi. V tomto prípade je inštancia anomálna, ak ju ani jeden z klasifikátorov pre normálne triedy neklasifikuje ako normálnu. Niektoré techniky tiež využívajú mieru istoty klasifikátora so svojím rozhodnutím. Ak žiadny z klasifikátorov nemá túto mieru vysokú pri tom, ako inštanciu klasifikuje ako normálnu, rozhodneme, že táto inštancia je anomálna.

Neurónové siete Neurónové siete sa využívajú pri *multi-class* detekcii anomálií. Základnou myšlienkou je natrénovať neurónovú sieť na normálnych dátach (naučiť ju rozpoznávať rôzne normálne triedy) a následne v testovacej fázi použiť inštanciu, ktorú chceme klasifikovať ako vstup do neurónovej siete. Ak ju prijme, jedná sa o normálnu inštanciu, ak nie o anomáliu[15].

Bayesovské siete Bayesovské siete sa využívajú pri *multi-class* detekcii anomálií. Tento spôsob je založený na určení posteriornej pravdepodobnosti, že inštancia patrí do danej triedy. Keďže máme viac tried, zvolíme ako výslednú triedu tejto inštancie tú s najväčšou pravdepodobnosťou [16]. Závislosti medzi jednotlivými atribútmi a výslednou triedou sú získané z trénovacej množiny. Táto technika predpokladá nezávislosť medzi atribútmi. Niektoré techniky tiež zachytávajú závislosti medzi rôznymi atribútmi využívajúc komplexné Bayesovské siete [17].

Support vector machine Support vector machine (SVM) sa využíva pri *one-class* detekcii anomálií [18]. Táto technika pracuje tak, že sa snaží zachytiť normálne správanie oblasťou, ktorá zachytáva trénovacie dáta. Pre komplexné normálne oblasti sa využívajú rôzne jadrové funkcie (napríklad radial basis function - RBF[19]). Klasifikácia následne prebieha pozorovaním, či testovaná inštancia spadá do naučeného regiónu a je normálnou alebo nespadá a je anomáliou.

Techniky založené na pravidlách Ako všetky klasifikátory, aj tento spôsob sa snaží zachytiť normálne chovanie dát. Ak inštancia, ktorú testujeme nie je zachytená žiadnym pravidlom, predpokladáme, že sa jedná o anomáliu. Tieto metódy sa používajú ako na *multi-class* tak aj *one-class* detekciu [14].

Prvou fázou je tiež trénovanie na základe trénovacej množiny, kde sa objavujú pravidlá v dátach. Typickými reprezentantmi týchto metód sú napríklad klasifikačné pravidlá. Každé získané pravidlo má priradenú takzvanú confidence hodnotu, ktorá je podielom počtu inštancií ktoré spĺňajú toto pravidlo a všetkých inštancií, ktoré sú zahrnuté týmto pravidlom. Druhým krokom je samotná detekcia anomálií. Keďže by sme už mali mať zachytené normálne

chovanie pravidlami, čo sme vytvorili, pre testovanú inštanciu zvolíme to najviac vyhovujúce pravidlo. Anomálnym skóre budeme nazývať prevrátenú hodnotu confidence tohto najviac vyhovujúceho pravidla.

Dolovanie asociačných pravidiel je taktiež využívané na *one-class* detekciu anomálií a to generovaním pravidiel bez učiteľa (unsupervised) [21].

3.2.4.1 Výhody a nevýhody

Výhody:

1. Pri presnom a dostatočne obsiahlom trénovacom data sete vieme zachytiť rôzne triedy normálneho správania a tým veľmi presne detekovať anomálie.

2. Testovanie prebieha rýchlo, keďže len využívame už natrénovaný model.

Nevýhody:

1. Multi-class detekcia sa spolieha na dostupnosť presne označených dát, čo v mnohých prípadoch vôbec nie je reálne.

3.2.5 Detekcia anomálií založená na vzdialenosti záznamov

Prístupy založené na vzdialenosti záznamov predpokladajú, že normálne dáta sa sú v zhlukoch zatiaľ čo anomálie sa v nich nevyskytujú (nachádzajú sa ďaleko od svojho najbližšieho suseda [22]), alebo sa vyskytujú v malých zhlukoch. Všetky tieto techniky tiež vyžadujú mieru, podľa ktorej môžeme jednotlivé inštancie porovnávať a tým získať istú mieru podobnosti alebo vzdialenosť medzi nimi. Pre spojité atribúty je klasickou voľbou Euklidovská vzdialenosť, pre iné je často potrebné použiť nejakú komplexnejšiu mieru. Ak inštancia obsahuje rôzne druhy atribútov je táto vzdialenosť obyčajne spočítaná pre jednotlivé atribúty zvlášť a následne skombinovaná.

Techniky založené na vzdialenosti záznamov sa všeobecne delia na dve kategórie

1. Techniky využívajúce vzdialenosť ku k-temu susedovi ako anomálne skóre

2. Techniky počítajúce relatívnu hustotu susedov pre každú inštanciu

Techniky využívajúce vzdialenosť ku k-temu susedovi Pri tomto prístupe je anomálne skóre inštancií počítané ako vzdialenosť ku k-temu susedovi. Citlivosť detekcie môžeme ovplyvňovať parametrom k, ale aj zvolením istej hranice anomálneho skóre alebo namiesto tejto hranice zvoliť n inštancií s najvyšším anomálnym skóre a prehlásiť ich za anomálie [23].

Iným spôsobom, ako vypočítať anomálne skóre je spočítať susedov (n), ktorí nie sú ďalej ako d. Jedná sa o určovanie globálnej hustoty, keďže počítame susedov v hyperguli o polomere d so stredom v danej inštancii [24][25][26].

Avšak anomálne skóre by malo stúpať ak predpokladáme s vyššou pravdepodobnosťou že sa jedná o anomáliu. Preto sú dva rôzne prístupy:

1. Stanoviť fixné d a anomálne skóre zvoliť ako $1/n$
2. Stanoviť fixné n a anomálne skóre zvoliť ako $1/d$

Keďže výpočetná zložitosť pri týchto metódach je $O(n^2)$, kde n je počet inštancií (rátame vzájomné vzdialenosti medzi všetkými inštanciami), mnohé prístupy sa snažia vylúčiť inštancie, ktoré nemôžu byť anomálne. Medzi tieto prístupy patrí napríklad technika, kde sa najskôr dáta rozdelia do zhlukov (clustering), v ktorých sa vypočíta spodná a horná hranica pre vzdialenosť od k-teho najbližšieho suseda. Táto informácia je následne použitá na identifikáciu partícií, v ktorých sa nemôže nachádzať k inštancií s najvyšším anomálnym skóre a ďalej ich neberieme v úvahu (anomálie hľadáme vo zvyšných partíciách). Ďalším prístupom ako zefektívniť túto metódu je hľadať najbližšieho suseda vrámci malej vzorky z data setu, čím sa zníži zložitosť na $O(mn)$, kde m je počet inštancií vo zvolenej vzorke.

Techniky počítajúce relatívnu hustotu susedov Tieto techniky počítajú relatívnu hustotu susedov pre každú inštanciu. Inštancie, ktoré ležia v hustom susedstve označujeme za normálne a naopak tie, ktoré v riedkom označujeme za anomálne. Pre zadanú inštanciu, vzdialenosť k jej k-temu susedovi odpovedá polomeru hypergule so stredom v tejto inštancii zahŕňajúcej k najbližších susedov našej inštancie. Z toho plynie, že táto vzdialenosť môže byť považovaná za inverziu k hustote a teda základná technika využívajúca vzdialenosť ku k-temu susedovi môže byť tiež technikou počítajúcou s relatívnou hustotou susedov.

Metódy rátajúce s touto hustotou nemusia pracovať správne nad dátami, kde sú oblasti s rôznymi hustotami výskytu inštancií. Aby sa tomuto predišlo, zaviedli sa metódy, ktoré zohľadňujú relatívnu hustotu svojich susedov. Jedným z riešení je napríklad *Local Outlier Factor* (LOF)[27]. Pre danú inštanciu dát, LOF skóre je pomer priemernej hustoty k najbližších susedov a lokálnej hustoty tejto inštancie. Pre vypočítanie tejto lokálnej hustoty najskôr nájdeme polomer najmenšej hypergule, ktorá obsahuje k najbližších susedov a následne vydeleniu k jej objemom. Pre normálne inštancie bude lokálna hustota podobná ako hustota ich susedov, pričom anomálne inštancie budú mať túto hustotu menšiu (jej LOF skóre bude vyššie).

Výpočetná zložitosť je pri LOF zase $O(n^2)$, kde n je počet inštancií a preto existujú rôzne modifikácie:

- Connectivity-based Outlier Factor (COF)[28]. Funguje inkrementálne, do okolia sa pridáva vždy inštancia, ktorá je najbližšie k súčasnému okoliu (najmenšia vzdialenosť od akejkoľvek inštancie v okolí) až kým nedosiahneme veľkosť okolia k. Následne sa anomálne skóre spočíta rovnako ako pri LOF.

- Outlier Detection using In-Degree Number (ODIN)[22]. Pre každú inštanciu spočítame počet k najbližších inštancií, pre ktoré sa zadaná inštancia nachádza v ich k najbližšom okolí. Prevrátená hodnota tohto počtu je anomálne skóre inštancie.

- Multi-granularity Deviation Factor (MDEF)[29]. Pre danú inštanciu spočítame štandardnú odchýlku lokálnych hustôt najbližších susedov (aj samotnej inštancie). Prevrátenou hodnotou tejto odchýlky je anomálne skóre inštancie.

Techniky založené na algoritmoch zhlukovania Zhlukovanie (clustering)[30] sa používa na organizovanie podobných dát do zhlukov. Zhlukovanie funguje zvyčajne bez učiteľa, ale existujú aj semi-supervised prípady. Aj keď sa môže zdať, že zhlukovanie a detekcia anomálií sú dve odlišné veci, existujú metódy detekcie anomálií založené na zhlukovaní.

Tieto metódy sa delia do troch kategórií podľa predpokladov o dátach:

1. Normálne inštancie patria do zhluku, pričom anomálie nepatria do žiadneho.

2. Normálne inštancie ležia blízko centroidu najbližieho zhluku, zatiaľ čo anomálie ležia ďaleko.

3. Normálne inštancie patria do veľkých a hustých zhlukov, pričom anomálie patria do malých alebo riedkych zhlukov.

Techniky založené na prvom tvrdení označujú všetky inštancie, ktoré sme nezaradili do zhluku za anomálne (príkladom je algoritmus ROCK [31]). Nevýhodou týchto techník však je, že nie sú optimalizované na nachádzanie anomálií, ale ich cieľom je nájsť zhluky.

Metódy založené na druhom tvrdení pozostávajú z dvoch krokov. V prvom kroku sa dáta zhlukujú pomocou nejakého zhlukovacieho algoritmu. V druhom pre každú inštanciu vyrátame vzdialenosť od centroidu najbližšieho zhluku, čo následne berieme ako anomálne skóre. Bežne používanými algoritmami pre zhlukovanie sú napríklad zhlukovanie *K-means* a *Self-Organizing Maps* (SOM)[32]. Tieto techniky však neodhalia anomálie, ak budú tvoriť vlastný zhluk [33].

Tretia kategória metód označuje za anomálne také inštancie, ktoré patria do zhlukov, ktorých veľkosť alebo hustota je pod zvolenou hranicou[34]. Jednou z techník ako takéto anomálne skóre zvoliť je *Cluster-Based Local Outlier Factor* (CBOLF), ktorý je prakticky zhlukovou variantou *Local Outlier Factor*. Zahŕňa ako veľkosť zhluku, tak aj vzdialenosť od centroidu zhluku do ktorého patrí.

Výpočetná zložitosť týchto metód závisí od zvoleného algoritmu. Ak je potrebné vypočítať vzdialenosti medzi dvojicami inštancií, je zložitosť obyčajne kvadratická, ale na druhú stranu ak sú použité algoritmy založené na heuristike, môže byť zložitosť lineárna. Testovacia fáza je obyčajne rýchla, keďže porovnávame inštancie s obmedzeným množstvom zhlukov.

3.2.5.1 Výhody a nevýhody

Výhody:

1. Jedná sa o unsupervised metódy detekcie anomálií a nepredpokladáme žiadne tvrdenia ohľadom distribúcie dát. Sú čisto založené na dátach.

2. Prispôsobovanie týchto metód na rôzne dáta je priamočiare, jediné čo je pri tom potrebné je mať mieru podobnosti pre inštancie.

Nevýhody:

1. Ak majú anomálie dostatok blízkych susedov a tiež naopak ak normálne inštancie majú málo blízkych susedov sa môže stať, že detekcia neprebehne správne.

2. Výpočetná zložitosť pri týchto metódach je vysoká (bežne $O(n^2)$), keďže musíme rátať vzdialenosti medzi všetkými inštanciami, alebo inštanciami patriacimi do nejakého okolia inštancie.

3. Spoľahlivosť detekcie anomálií sa spolieha na zvolenú mieru podobnosti inštancií. Zvoliť mieru môže byť nadmieru obtiažna úloha, ak sa jedná o komplexné dáta (nespojité atribúty, postupnosti a iné).

3.2.6 Štatistická detekcia anomálií

Tieto metódy sa zakladajú na myšlienke, že anomáliou je taká inštancia, ktorá neodpovedá predpokladanému stochastickému modelu. Spolieha sa pritom na tvrdenie, že normálne dáta sa vyskytujú vo vysoko pravdepodobných oblastiach stochastického modelu, pričom anomálie naopak v oblastiach s nízkou pravdepodobnosťou [35].

Štatistické techniky detekcie fitujú štatistický model na dané dáta a následne sledujú či ďalšie inštancie patria do tohto modelu alebo nie. Inštancie, čo majú nízku pravdepodobnosť, že sú generované týmto modelom (na základe aplikovanej testovacej štatistiky) prehlásime za anomálie. Sú využívané ako parametrické, tak aj neparametrické techniky.

3.2.6.1 Techniky založené na modeli

Predpokladáme, že dáta sú generované parametrickou distribúciou s parametrami θ a s hustotou pravdepodobnosti $f(x, \theta)$, kde x je pozorovanie. Anomálne skóre testovanej inštancie vypočítame ako prevrátenú hodnotu $f(x, \theta)$. Parametre θ určujeme na základe daných dát.

Alternatívnou možnosťou detekcie anomálií v tomto modeli je tiež testovanie hypotéz. Zvolíme nulovú hypotézu H_0 tak, že inštancia x bola generovaná predpokladanou distribúciou (s parametrami θ). Ak štatistický test zamietne hypotézu H_0, prehlásime x za anomáliu. Testovanie hypotéz je spojené s testovacou štatistikou, ktorá môže byť použitá na získanie pravdepodobnostného anomálneho skóre pre inštanciu x.

Gaussovský model Tieto techniky predpokladajú, že dáta boli generované Gaussovským rozdelením. Parametre sú určené pomocou metódy *Maximum Likelihood Estimates* (MLE). Vzdialenosť inštancie od priemeru je potom braná ako anomálne skóre. Pre označenie anomálií sa volí hranica a inštancie nad túto hranicu sú označené za anomálie. Rôzne techniky rátajú túto vzdialenosť od priemeru rôznym spôsobom.

Jednou z najjednoduchších detekcií odľahlých inštancií je označiť všetky inštancie, ktoré sú od priemeru μ vzdialené viac ako 3σ, kde σ je smerodatná odchylka rozdelenia. Oblasť $\mu \pm 3\sigma$ zahŕňa 99.7% inštancií.

Ďalšou jednoduchou metódou je využitie box plot rule. *Box-plot* graficky znázorňuje najmenšie neanomálne pozorovanie, dolný kvartil (Q_1), medián, horný kvartil (Q_3) a najväčšie neanomálne pozorovanie. $Q_3 - Q_1$ sa nazýva Inter Quartile Range (IQR). Box plot tiež indikuje, kedy pozorovanie pokladať za anomáliu. Inštancia dát, ktorá leží viac ako $1.5 \cdot IQR$ pod Q_1, alebo $1.5 \cdot IQR$ nad Q_3, je označovaná za anomáliu. Oblasť $Q_1 - 1.5 \cdot IQR$ až $Q_3 + 1.5 \cdot IQR$ obsahuje 99.3% pozorovaní pri normálnom rozdelení a teda voľba $1.5 \cdot IQR$ ako hranice anomálnosti je takmer ekvivalentná 3σ technike.

Grubbov test zase využíva výpočet z skóre pre každú inštanciu (predpokladáme jednorozmerné dáta) x: $z = \frac{|x-\bar{x}|}{s}$, kde $\bar{x}$ je priemer a s je štandardná odchylka vzorky dát. Inštancia je potom anomálna ak

$$z > \frac{(N-1)}{\sqrt{N}} \cdot \sqrt{\frac{t^2_{\alpha/2N,N-2}}{N-2+t^2_{\alpha/2N,N-2}}}$$

kde N je počet inštancií, $t^2_{\alpha/2N,N-2}$ je hranica určujúca, či je inštancia anomálna (hodnota t-rozdelenia na hladine významnosti $\alpha/2N$) [36].

Varianta Grubbovho testu pre viacrozmerné dáta počíta s Mahalanobisovou vzdialenosťou inštancie od priemeru na redukovanie viacrozmerného priestoru do jednorozmerného skaláru.

$$y^2 = (x - \bar{x})' S^{-1} (x - \bar{x})$$

Následne je na y uplatnený Grubbov test podobne ako pri jednorozmerných dátach.

Jednou z ďalších variant detekcie je použitie χ^2 štatistiky. Predpokladáme, že máme viacrozmerné dáta s normálnym rozdelením. Potom je hodnota χ^2 štatistiky definovaná ako:

$$\chi^2 = \sum_{i=1}^{n} \frac{(X_i - E_i)^2}{E_i}$$

kde X_i je hodnota i-teho atribútu, E_i je priemerná hodnota i-teho atribútu (získaná z trénovacieho data setu) a n je počet atribútov. Veľká hodnota χ^2 značí, že sa v pozorovanej vzorke nachádzajú anomálie.

Kombinácia parametrických rozdelení Táto kategória je rozdelená na dva smery:

1. Modelovanie normálnych a anomálnych inštancií odlišnými parametrickými rozdeleniami. Testovanie prebieha sledovaním, do ktorého rozdelenia patrí daná inštancia.

2. Modelovanie normálnych inštancií ako kombináciu parametrických rozdelení. Testovanie prebieha skúmaním, či daná inštancia patrí do nejakého naučeného rozdelenia. Ak nie, je prehlásená za anomáliu.

3.2.6.2 Histogramy

Pre jednorozmerné dáta je základnou myšlienkou vytvoriť histogram nad týmito dátami a následne sledovať či testovaná inštancia spadá do niektorého z binov. Ak áno, je inštancia prehlásená za normálnu, ak nie, za anomálnu. Ak zvolíme príliš malé biny, môže sa stať, že aj normálne inštancie budú spadať do prázdnych oblastí a tým pádom budú nesprávne detekované. Naopak

pri príliš veľkých binoch môžu zase byť anomálie klasifikované ako normálne inštancie [37].

Pre viacrozmerné dáta histogramová metóda pracuje s atribútmi oddelene a vždy sleduje veľkosť binu, do ktorého hodnota atribútu spadá a následne tieto veľkosti sčítame. Anomálne skóre získavame ako prevrátenú hodnotu týchto veľkostí [38].

3.2.6.3 Výhody a nevýhody

Výhody:

1. Ak sú splnené predpoklady, štatistické metódy poskytujú štatisticky dokázateľné riešenie pre detekciu anomálií.

2. Anomálne skóre je spojené s konfidenčným intervalom, čo môže byť použité pri voľbe hranice.

Nevýhody:

1. Štatistické metódy sa spoliehajú na predpoklady o dátach. Tieto predpoklady častokrát nie sú splnené [36]).

2. Aj keď sú predpoklady splnené, testovanie hypotéz je obtiažnou úlohou (napríklad už zostaviť testovaciu hypotézu pre dáta s vysokou dimenziou je netriviálne).

3. Histogramové metódy sú síce jednoduché na implementáciu, ale nie sú schopné zachytiť závislosti medzi jednotlivými atribútmi (anomália môže mať hodnoty atribútov normálne, ale ich kombinácia môže byť nezvyčajná).

3.2.7 Spektrálne techniky

Spektrálne techniky sa snažia o aproximáciu dát použitím kombinácie atribútov zachytávajúcich rozptyl v dátach. Predpokladáme, že dáta môžu byť transformované do priestoru s nižšou dimenziou, kde sa normálne a anomálne inštancie javia značne odlišné.

Niektoré z týchto techník využívajú Principal Component Analysis (PCA) pre projekciu dát do nového priestoru [39]. Jednou z nich je napríklad analýza projekcie každej inštancie do hlavných komponent s nízkym rozptylom. Normálna inštancia, ktorá odpovedá korelačnej štruktúre má nízku hodnotu projekcie zatiaľ čo anomália vysokú.

Spektrálnou technikou na hľadanie anomálií v časových radách grafov je napríklad reprezentovať graf ako maticu susednosti pre daný časový okamih. Pre každú časovú inštanciu bude zvolený vektor aktivity (zmeny) ako hlavná

komponenta. Časová rada týchto vektorov je braná ako matica a z nej získavame hlavný ľavý singulárny vektor (principal left singular vector) pre zachytenie normálnych závislostí v dátach vzhľadom na čas. Pre nový záznam (graf) získavame jeho anomálne skóre ako uhol medzi týmto vektorom a vektorom aktivity nového záznamu [40].

3.2.7.1 Výhody a nevýhody

Výhody:

1. Tieto techniky sú vhodné na analýzu vysokodimenzionálneho priestoru, keďže ho redukujú. Tiež môžu byť použité ako predspracovanie pre iné techniky.

2. Vieme ich aplikovať v prostredí bez učiteľa.

Nevýhody:

1. Sú použiteľné len ak sú anomálie a normálne inštancie separabilné v priestore s nižšou dimenziou.

2. Vysoká výpočetná zložitosť.

3.2.8 Detekcia kontextových anomálií

Predchádzajúce techniky boli primárne zamerané na identifikáciu bodových anomálií. Detekcia kontextových anomálií vyžaduje, aby dáta mali kontextuálne a behaviorálne atribúty.

Techniky zaoberajúce sa kontextovými anomáliami môžeme deliť na dve kategórie:

Redukcia problému na bodovú detekciu anomálií Keďže kontextové anomálie sú inštancie, ktoré sú anomálne len vzhľadom na kontext, jedným z prístupov je aplikovať bodovú detekciu anomálií v tomto kontexte.

Táto redukcia najskôr určí kontext pre každú z inštancií využívajúc kontextuálne atribúty a následne vypočíta anomálne skóre pomocou niektorej z techník bodovej detekcie anomálií.

Využitie štruktúry dát V niektorých prípadoch nie je rozdelenie na kontexty priamočiare (typicky pre časové rady). Základnou myšlienkou tohto prístupu je naučenie modelu na trénovacích dátach, tak aby vedel určovať behaviorálne atribúty na základe kontextu. Ak je očakávané chovanie iné, predpokladáme anomáliu.

3.2.8.1 Výhody a nevýhody

Výhody:

1. Sú schopné detekovať anomálie, ktoré by nemuseli byť odhalené bodovými detekciami.

Nevýhody:

1. Sú aplikovateľné len keď môže byť kontext jasne definovaný.

3.2.9 Detekcia kolektívnych anomálií

Primárnym predpokladom pre detekciu kolektívnych anomálií sú závislosti medzi inštanciami dát.

3.2.9.1 Detekcia sekvenčných anomálií

Tieto anomálie môžu byť rozdelené do troch kategórií:

Detekcia anomálnej sekvencie v množine sekvencií Tieto techniky pracujú semi-supervised, alebo unsupervised. Najväčšími problémami v tejto oblasti sú rozdielne dĺžky sekvencií a tiež rozdielne zarovnanie.

1. Prvým prístupom ako tieto anomálie detekovať je zase redukcia na bodovú detekciu anomálií. Snažíme sa teda jednotlivé sekvencie previesť do konečného priestoru a v ňom aplikujeme jednu z metód bodovej detekcie.

2. Druhým je modelovanie sekvencií. Najčastejšou metódou na toto modelovanie je pomocou Markovských modelov.

Detekcia anomálnej subsekvencie v sekvencii Jedná sa o detekciu anomálneho vzoru vrámci sekvencie udalostí alebo časovej rady [41]. Táto detekcia pracuje zvyčajne v unsupervised móde a teda predpokladá, že sa časová rada odpovedá definovanému vzoru. Táto detekcia naráža opäť na problémy. Jedným z najzávažnejších je fakt, že vo všeobecnosti nepoznáme dĺžku anomálnej sekvencie [10][42].

Detekcia, či frekvencia vzoru v sekvencii nie je anomálna Detekovať tento typ anomálií znamená nájsť vzory, ktorých frekvencia výskytu v inštancii sa líši od frekvencie v normálnom data sete[43]. Bežne sa využíva metóda pohyblivého okna[44].

3.2.9.2 Detekcia priestorových anomálií

Kolektívna detekcia anomálií v priestorových dátach zahŕňa nachádzanie podgrafov alebo subkomponent v dátach, ktoré sú anomálne. Môžeme ju rozdeliť na hľadanie anomálií v statických a dynamických grafoch.[45]

Statická detekcia anomálií v grafoch Prvá kategória zahŕňa detekciu anomálií v statických grafoch, t.j. grafoch, ktoré sa v čase nemenia. Detekujeme teda anomálne hrany, uzly alebo dokonca podgrafy v celom zadanom (alebo nami vytvorenon) grafe.

Túto detekciu ďalej rozdeľujeme podľa toho, či sa jedná o jednoduchý graf, alebo je tento graf nejakým spôsobom označený (uzly alebo hrany).

Jednoduché grafy Keďže nemáme k dispozíci žiadne údaje okrem štruktúry grafu, musíme sa teda sústrediť na ňu - nájsť pravideľnosti a následne identifikovať prvky, ktoré túto pravideľnosť porušujú. Spomínané pravideľnosti tiež môžu byť dvoch druhov a to štruktúrne vzory a komunitné vzory.

Prvou kategóriou sú teda štruktúrne vzory. Pri týchto vzoroch sa sústredíme na extrahovanie číselných atribútov reprezentujúcich štruktúru grafu ako napríklad stupne uzlov a tiež vzdialenosti v grafe. Príkladmi týchto metód sú napríklad ODDBALL [46], ktorý pre každý uzol v grafe zoberie jeho bezprostredných susedov (ich vzdialenosť od daného uzlu je 1) a hrany medzi nimi a následne z týchto podgrafov extrahuje vlastnosti, na ktoré je použitá jedna z techník detekcií bodových anomálií. Druhým príkladom týchto metód je metóda PageRank [47]. Jej princíp zase spočíva v ohodnotení každého uzlu (stupeň významnosti/dôležitosti) na základe náhodnej prechádzky grafom.

Druhou kategóriou sú zase komunitné vzory. Ich myšlienkou je skupinkovať (vytvárať komunity) uzly, ktoré sú husto prepojené a skúmať, ktoré z uzlov/hrán majú prepojenie mimo vlastnú komunitu. Príkladom využívania týchto vzorov je napríklad AUTOPART [48], ktorý vytvára spomínané komunity na základe pôsobnosti susedov a hrany, ktoré spájajú dve komunity sú považované za anomálne. Taktiež uzly, ktoré majú veľa spojení naprieč rôznymi komunitami sú považované za anomálie.

Označené grafy Ak máme isté označenie grafov a hrán, môžeme metódy opísané pre jednoduché grafy rozšíriť o túto informáciu.

Prvou kategóriou vzorov sú zase štruktúrne vzory. Hlavnou myšlienkou je hľadať také vzory, ktoré sú anomálne nielen konektivitou, ale aj atribútmi (označením). Jedným z prístupov je napríklad opísaný Noblesom a Cookom [49] a je založený na hľadaní opakov takzvaných najlepších podgrafov. Najlepšie podgrafy sú také podgrafy, ktoré sa vyskytujú v grafe často a teda ho dokážu dobre komprimovať.

Druhou kategóriou sú teda komunitné vzory. Pri ich nachádzaní môžeme postupovať buď tak, že anomálie detekujeme priamo pri vytváraní komunít, alebo detekcia anomálií prebieha až po atribútovanom zhlukovaní v grafe.

Ďalšou kategóriou, ktorú prináša zavedenie atribútov, sú vzory založené vzťahoch medzi jednotlivými záznamami a hodnotami ich atribútov. Jedná sa o klasifikáciu.

Dynamická detekcia anomálií v grafoch Na druhú stranu, keď sa jedná o dynamickú detekciu anomálií, predpokladáme sekvenciu grafov (či už jednoduchých alebo atribuovaných). Jej cieľom je identifikovať vrámci tejto sekvencie anomálne zmeny (udalosti). Podľa [45], ako pri statickej detekcii anomálií, aj tu sa môžeme na tento problém pozerať z rôznych hľadísk.

- **Techniky založené na vlastnostiach grafu** Tieto techniky využívajú extrakciu istých vlastností grafu a ich následnom porovnávaní pri dvoch po sebe idúcich grafoch v sekvencii. Ak je vzdialenosť medzi dvoma grafmi príliš veľká, jedná sa o anomáliu. Voľba týchto vlastností však nie je jednoznačná a líši sa od problému k problému. Najčastejšie vlastnosti a miery podobnosti sú napríklad: Vzdialenosť *Maximum Common Subgraph* (MSC) alebo *Graph Edit Distance* (GED).
- **Komunitné techniky** Princíp týchto techník spočíva v zhlukovaní jednotlivých uzlov do komunít a sledovaní štrukturálnych alebo kontextuálnych zmien vrámci komunít namiesto v celom grafe.
- **Sledovanie zmien** Poslednou kategóriou dynamickej detekcie anomálií v grafoch je sledovanie zmien v posledných grafoch (vytvoriť akýsi vektor zmien) a sledovaní, či nasledujúci graf je oproti predchádzajúcemu iný v akceptovateľnom zmysle, t.j. zmena odpovedá do istej miery vektoru zmien v predchádzajúcich grafoch.

Kapitola 4

Realizácia

Na základe teoretickej časti vieme akým spôsobom musíme aplikovať nadobudnuté znalosti na vstupné dáta aby sme získali požadované výstupy.

4.1 Použité nástroje

4.1.1 Gnumeric

Gnumeric je tabuľkový procesor [50], počítačový program vytvorený GNOME projectom, ktorý slúži na manipuláciu a analýzu číselných dát. Gnumeric pomáha sledovať informácie v zoznamoch, organizovať číselné hodnoty do stĺpcov a riadkov, vykonávať a aktualizovať zložité výpočty tým, že definujeme jednotlivé kroky výpočtu a následne ich modifikujeme. Umožňuje tiež vytvárať a zobraziť alebo vytlačiť rôzne typy grafov a vykonávať zložité optimalizačné modelovanie alebo vykonávať mnoho ďalších úloh, zahŕňajúcich čísla, dátumy, časy, mená alebo iné dáta. *Ssconvert* je nástroj príkazového riadka obsiahnutý v balíčku Gnumeric pre konverziu tabuľkových súborov na rôzne formáty. Jeho syntax:

ssconvert [*OPTIONS*] *infile outfile*

4.1.2 scikit-learn

Scikit-learn je jednoduchý a zároveň efektívny framework na data mining a dátovú analýzu. Poskytuje rôzne možnosti trénovania modelov, ich následnej aplikácie, detekcie anomálií a iných techník strojového učenia ako aj možnosti grafického znázornenia dát alebo výsledkov. Využíva moduly *NumPy, SciPy* a *matplotlib*.

4.2 Spracovanie dát

Na základe požadovaných výstupov práce je samozrejmé, že dáta musia byť najskôr transformované do takej podoby, aby boli analyzované čo najjednoduchším spôsobom a taktiež automaticky (t.j. skriptom, alebo programom). Snažíme sa teda o to, aby v dátach neboli chýbajúce atribúty a tiež aby neobsahovali nezmyselé hodnoty, ktoré by túto analýzu znemožňovali.

4.2.1 Formáty súborov

Prvým problémom, ktorý je potrebné vyriešiť, je zjednotenie formátov súborov, v ktorých sa dáta nachádzajú. Keďže je .xlsx príliš špecifickým formátom a tým pádom aj ťažko spracovateľným automaticky, je vhodné zvoliť iný formát. Na druhú stranu, .csv je štandardizovaný formát, s ktorým vieme dostatočne jednoducho pracovať (ako s textovým súborom) a preto som sa rozhodol použiť tento formát. Avšak, keď už sme sa rozhodli použiť formát .csv, je potrebné tiež zvoliť oddeľovač. Aj keď niektoré z poskytnutých súborov obsahujú desatinné čísla, ktoré používajú ako desatinnú čiarku znak ',' namiesto bežného '.', záznamy, ktoré obsahujú desatinné číslo nie sú správne (žiadny atribút by nemal nadobúdať hodnoty desatinných čísel). S týmito nesprávnymi záznamami je potrebné sa vysporiadať skôr ako sa dáta budú automaticky spracovávať alebo zlučovať a preto nie je žiaden dôvod nepoužiť ako oddeľovač znak ','.

Napriek zdanlivej problematickosti tejto úlohy táto úprava bola vykonaná pomocou niekoľkých jednoduchých príkazov *sed s*, ktorým som najskôr vymenil všetky čiarky za bodky v súboroch, kde je ako oddeľovač použitý znak ';'. Následne som vymenil všetky znaky ';' za čiarky, čím som zaručil, že všetky súbory sú vo formáte .csv s jednotným oddeľovačom.

4.2.2 Atribúty

Dátumy Keďže sa vyskytujú v mnohých rôznych formátoch, je potrebné zjednotiť.

Rozhodol som sa pre formát **Y-m-d H:M:S**, ktorý je všeobecne akceptovaný rôznymi dataminingovými nástrojmi ktorý som sa preto rozhodol použiť ako pre *BirthDate*, tak aj pre *ScheduledArrival*. Pri *BirthDate* budú hodiny, minúty a sekundy nastavené na 0 a pri spracovaní ignorované. Rovnaký formát som zvolil kvôli jednoduchému odčítavaniu dátumov na získanie veku pasažiera.

Na túto úlohu som využil modul *datetime* v jazyku Python s jeho funkciami *strptime(format)* pre načítanie dátumu z textového reťazca v zadanom formáte a *strftime(format)* pre konverziu dátumu do formy textového reťazca v zadanom formáte. Takto vieme pomocou blokov *try-except* v pythone identifikovať jednotlivé formáty a následne ich konvertovať do jednotného.

Atribúty HeadGUID a BodyUID Keďže sa tieto atribúty vyskytujú len vo výnimočných prípadoch (samozrejme ich majú celé lety, ale len niektoré) a tiež ich význam nie je žiadnym spôsobom smerodajný, rozhodol som sa tieto atribúty vynechať.

Atribút Nationality Neprítomnosť atribútu pasažiera, ktorý určuje národnosť pasažiera je rozhodne zvláštnosťou. Keďže však o záznamy, ktoré tento atribút nemajú nechceme prísť, zvolíme nahradenie tohto atribútu nejakou špecifickou hodnotou. Jeho nahradenie priemerom by nebolo rozumné, kvôli tomu, že by sme stratili informáciu o tom, že tento pasažier daný atribút nemal (táto informácia by však mohla viesť k nejakej kľúčovej závislosti medzi podozrivými pasažiermi).

Atribút Names Keďže nemáme ako určiť, kde by mala byť medzera medzi menami a spojenie všetkých mien do reťazca bez medzier by mohlo viesť k zjednoteniu rôznych pasažierov, preto necháme tento atribút v pôvodnej forme.

Atribút Sex Keďže nechceme prísť o informáciu, že daný atribút pri zázname chýbal, volíme doplnenie špeciálnej hodnoty.

Atribút DocumentType Prvou nekonzistenciou tohoto atribútu je jeho neprítomnosť pri niektorých záznamoch. Nechceme prísť o informáciu, že daný atribút pri zázname chýbal, takže zase volíme doplnenie špeciálnej hodnoty.

Druhou nekonzistenciou je, že pri takmer každom zázname je uvedený typ dokumentu ako pas (aj keď z formátu čísla dokumentu je jasne vidno, že sa jedná o občiansky preukaz). Týmto atribút stráca na svojej informačnej hodnote a zároveň nevieme tieto chybné hodnoty upraviť tak aby reálne odpovedali typu dokumentu, takže zvyšné hodnoty nechávame v pôvodnej podobe.

Atribút DocumentIssued Neprítomnosť atribútu DocumentIssued je zase pozoruhodnou informáciou, takže tieto neprítomnosti nahrádzame špeciálnou hodnotou.

Atribút DocumentNumber Informáciu o neprítomnosti tohoto atribútu nechceme stratiť, preto chýbajúce hodnoty nahrádzame špeciálnou hodnotou.

Číslo dokumentu, ktorým sa pasažier preukazuje by tiež nemalo nadobúdať desatinných hodnôt. Tieto hodnoty považujeme za nesprávne a teda ich tiež nahradíme za špeciálnu hodnotu (inú ako pre chýbajúce dáta, keďže tieto prípady chceme rozlišovať).

Atribút Reservation Prvým problémom pri atribúte Reservation je neprítomnosť atribútu. Ako aj pri ostatných chýbajúcich atribútoch, aj tu použijeme nahradenie špeciálnou hodnotou, lebo informácia, že atribút chýbal je pre nás cenná.

Druhým problémom bolo, že atribút nadobúdal hodnôt kladných celých čísel aj keď sa nejednalo o rezerváciu. Tieto hodnoty však nemáme ako odlíšiť od reálnych rezervácií a preto ich ponecháme v pôvodnom stave.

Ďalším problémom bolo, že tento atribút nadobúdal hodnôt desatinných čísel. Keďže pasažieri, čo cestujú na jednu rezerváciu majú túto hodnotu rovnakú, tieto hodnoty ponecháme tiež v pôvodnom stave.

Atribút HitType Sú dva typy nekonzistencií, čo sa týkajú atribútu *HitType*. Prvou je neprítomnosť atribútu. Keďže aj neexistencia atribútu je istou informáciou, o ktorú nechceme prísť, doplníme do záznamov, kde atribút chýba špeciálnu hodnotu.

Druhým druhom nekonzistencie je, že tento atribút nadobúda hodnôt, ktorých význam nie je známy. Kedže však *HitType* bezpečného pasažiera by mal nadobúdať vždy hodnotu 1, pasažieri, čo majú inú hodnotu tohoto atribútu sú istým spôsobom zaujímaví. Preto tieto hodnoty môžeme nechať v pôvodnom stave (prípadne pri spracovaní niektorým data miningovým nástrojom hodnoty označiť ako 1 - bezpeční a 0 - všetci ostatní, kde teda spadajú aj pasažieri, čo majú inú hodnotu *HitType* ako 1, aj tí, pri ktorých tento atribút chýbal).

4.2.3 Zjednotenie dát

Predpokladajme, že predchádzajúce kroky prebehli bez problémov a máme všetky dáta v jednotnej forme a to v súboroch .csv, kde každý záznam má presne 14 atribútov. Pre jednoduchšiu manipuláciu s týmito dátami by bolo lepšie ich mať v jednom súbore. Rozčlenenie do jednotlivých adresárov síce zlepšuje prehľadnosť pre užívateľa, ale predpokladáme, že užívateľ do týchto dát bude zasahovať (bude ich skúmať manuálne) v čo najmenšej miere.

Keďže jednoznačne identifikovať let vieme už z jednotlivých záznamov v týchto súboroch (kombinácia atribútov *FlightNumber, ScheduledArrival, FlightFrom, FlightTo*), nie je potrebné túto adresárovú štruktúru udržiavať. Preto som sa rozhodol spojiť všetky .csv súbory do jedného.

4.2.4 Anonymizácia dát

Pre zverejenie práce a tiež prípadné naväzovanie na ňu je potrebná anonymizácia dát. Keďže chceme zachovať všetky informácie a zároveň chceme, aby daný pasažier nebol dohľadateľný na základe anonymizovaných záznamov, musíme niektoré z údajov vynechať alebo zakódovať. Aby sme vedeli identifikovať, keď

sa v dvoch záznamoch jedná o rovnakú osobu, rozhodol som sa dané atribúty nevynechať, ale zahashovať jednosmerným hashom. Zvoleným hashom je SpookyHash - jedna z Jenkinsových hashovacích funkcií.

Atribúty, ktoré hashujem do jednoznačnej identifikácie budú: *Nationality, Surname, Names, BirthDate, Sex, DocumentType, DocumentIssued, DocumentNumber*. Hashovanie nebude samozrejme prebiehať po zložkách, ale všetky tieto atribúty zreťazím do jedného textového reťazca, ktorý následne použijem ako vstup do spomínanej hashovacej funkcie. Jej výstupom bude nový atribút PassengerID. Pôvodné atribúty, ktoré umožňujú jednoznačnú identifikáciu do anonymizovaného datasetu nezahŕňam. Atribúty, ktoré ďalej používam na učenie sú:

1. *PassengerID*
2. *FlightNumber*
3. *ScheduledArrival*
4. *Nationality*
5. *BirthDate*
6. *Sex*
7. *DocumentType*
8. *DocumentIssued*
9. *FlightFrom*
10. *FlightTo*
11. *Reservation*
12. *HitType*

Pre určenie kontextu nechávame všetky atribúty identifikujúce let. Keďže zoznamy pasažierov nie sú verejné, nejedná sa o citlivú informáciu.

Ako vidíme, niektoré atribúty sú použité v *PassengerID* aj ponechané v anonymizovanom datasete. Jedná sa o atribúty, ktorých ani kombinácia jednoznačne neidentifikuje človeka a pritom pomocou nej môžeme skúmať rôzne závislosti a trendy. Preto túto anonymizáciu považujem za primeranú.

Anonymizáciu dát realizujem pomocou programovacieho jazyka Python, ktorý ponúka implementáciu funkcie SpookyHash a tiež elegantnú prácu s .csv súbormi.

4.2.5 Transformácia dát pre scikit-learn

Keďže práca s kategorickými atribútmi (a tým pádom aj nenormalizovanými) znemožňuje použitie viacerých kľúčových metód zo *scikit-learn*u, dáta najskôr transformujem, aby atribúty boli spojité a normalizované. Táto transformácia prebieha pythonovským skriptom za použitia knižníc *scikit-learn* 4.1.2 a datetime.

1. Konverzia dátumov na timestamp.

2. Transformácia kategorických atribútov na numerické. Táto konverzia prebieha pomocou triedy *LabelEncoder*, ktorá každej nadobúdanej hodnote atribútov priradí unikátne celé číslo. Z tejto konverzie vynechávame dátumy, pretože ich už máme vo formáte celého čísla. Nationality a DocumentIssued zase nadobúdajú rovnakých hodnôt (jedná sa o národnosti - respektíve trojpísmenové skratky štátov) a preto hodnoty týchto dvoch atribútov sú zakódované rovnako.

3. Normalizácia dát. Keďže na základe predchádzajúceho kroku už máme všetky atribúty transformované na numerické, uplatňujem *MinMax* normalizáciu (transformáciu hodnot jednotlivých atribútov do intervalu $[0, 1]$).

Celá transformácia je samozrejme časovo náročná, pretože máme viac ako dva a pol milióna záznamov. Aby som ju nemusel vykonávať pre každý nový skript, používam knižnicu *pickle* pre jednoduchú a elegantnú serializáciu a deserializáciu dát.

Táto transformácia je vratná, čo nám umožňuje aby boli výsledky následne interpretovateľné.

4.3 Detekcia anomálií

Po vykonaní predchádzajúcej transformácie dát môžeme začať pracovať na automatickej analýze dát. Prvým požadovaným výstupom je detekcia anomálií.

Dáta z *OBZOR*u ponúkajú mnohé možnosti definície kontextu - od detekcie anomálií v dátach ako v celku až po detekovanie anomálií v sekvencii letov jedného pasažiera (profil pasažiera). Pre demonštrovanie rôznych techník som sa rozhodol použiť práve tieto dve definície kontextu.

4.3.1 Dáta ako celok

Keď definujeme kontext týmto spôsobom, môžeme samotnú detekciu anomálií vykonať rôznymi spôsobmi.

Prvou otázkou je, či budeme záznamy, ktorých atribút *HitType* nadobúda hodnotu inú ako 1 brať ako anomálne:

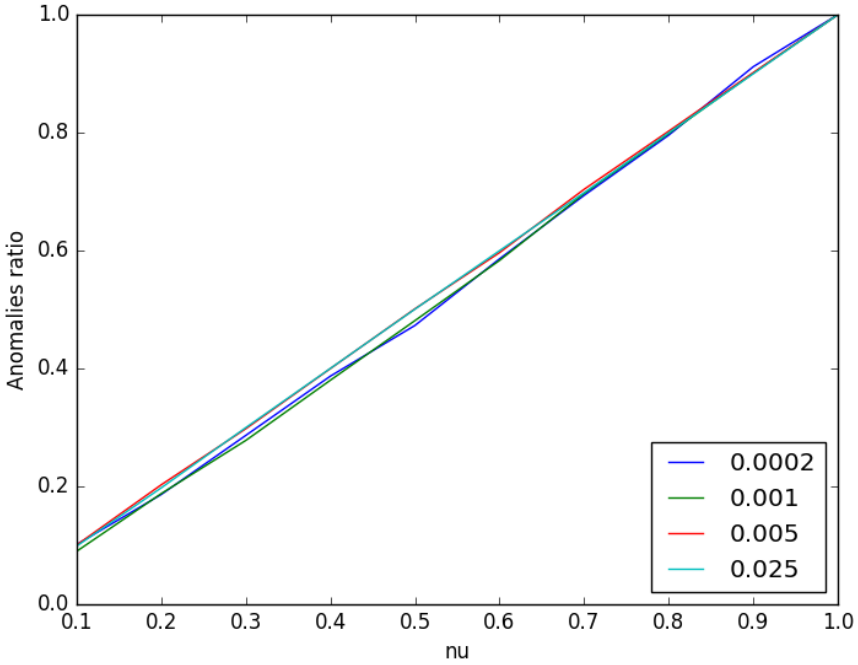
- **HitType určuje anomálnosť** Ak *HitType* určuje anomálnosť, môžeme využívať supervised techniky detekcie anomálií. Pre zjednodušenie rozdelíme záznamy na normálne - tie, ktoré majú *HitType* 1 a tie, ktoré majú inú hodnotu *HitType* - anomálne záznamy. Problém detekcie anomálií sme takto zredukovali na problém binárnej klasifikácie záznamov. Aplikovať teda budeme techniky opísané v 3.2.4. Keďže tento problém je analogický s klasifikáciu „nebezpečných“ pasažierov, ktorá je zase výstupom odpovede na analytické otázky, tento problém je podrobne analyzovaný v sekcii 4.4.1.

- **HitType neurčuje anomálnosť** V prípade, že *HitType* neberieme ako označenie anomálnosti záznamu, prichádzajú v úvahu unsupervised techniky detekcie anomálií ako techniky najbližšieho suseda 3.2.5, techniky založené na zhlukovaní 3.2.5, alebo iné.

V tejto sekcii sa teda budeme sústrediť na unsupervised detekciu anomálií. *Scikit-learn* obsahuje tiež rôzne možnosti unsupervised detekcie anomálií, preto som sa rozhodol pri každej ponúkanej možnosti preskúmať do akej miery ovplyvňujú parametre výstup detekcie anomálií. Všetky tieto experimenty vedú k znalosti, či je daný nástroj detekcie použiteľný, alebo je výstup diametrálne odlišný už pri malých zmenách v parametroch a teda označujeme zakaždým inú anomálnu množinu alebo nevidíme vo výstupe žiadnu závislosť na skúmanom parametri.

OneClassSVM Prvou možnosťou detekcie anomálií je využitie triedy *OneClassSVM*, ktorá má základy v štatistickej detekcii anomálií a keďže je unsupervised, nebudeme používať *HitType*. Jej parametrom je *nu*, ktorý určuje hornú hranicu trénovacích chýb (training errors) a zároveň spodnú hranicu support vektorov. Princíp detekcie anomálií spočíva v nafitovaní na trénovacie dáta - natrénovanie normálneho správania, kde určujeme pomocou parametra *nu* to, koľko očakávame v trénovacích dátach anomálií, alebo abnormálnych vzorkov.

Keďže sa jedná o trénovanie SVM, ktorého zložitosť je kvadratická s počtom trénovacích vzoriek, najskôr som sa sústredil na vplyv počtu trénovacích vzoriek na to, koľko bude v testovacej časti označených vzorkov z testovaných vzorkov. Využíval som stratifikovaný výber vzorkov s rôznymi pomermi trénovacej a testovacej časti (testovacia časť je vždy doplnok k trénovacej). Techniku aplikujeme na anonymizované a transformované dáta (4.2.5).



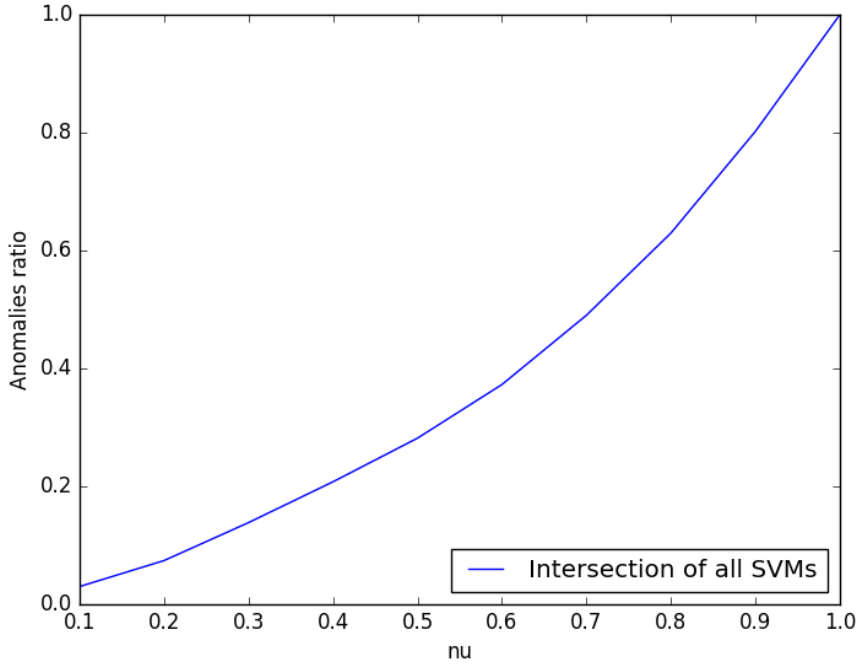
Obr. 4.1: *OneClassSvm* - Podiel veľkosti trénovacej množiny

Ako vidíme na grafe 4.1, kde jednotlivé krivky odpovedajú pomeru trénovacích dát k celkovému počtu vzorkov, aj relatívne malá časť dát určená na trénovanie je schopná zachytiť normálne chovanie v celom datasete. Preto naďalej budem *OneClassSvm* trénovať na trénovacej množine o veľkosti 1% z celkového počtu záznamov využívajúc stratifikovaný výber vzoriek.

Vidíme tiež, že parametrom *nu* do detekcie anomálií prinášame ovplyvňovanie citlivosti. Keďže SVM vieme natrénovať s rôznymi kernelmi, rozhodol som sa preskúmať do akej miery sa výstupy týchto detekcií anomálií prekrývajú (a ak všetky tieto SVM natrénované s rôznymi kernelmi označia záznam za anomálny je vysoká pravdepodobnosť, že anomálny reálne bude).

Skúmam teda prienik štyroch *OneClassSvm* natrénovaných na kerneloch:

1. Radial basis function kernel
2. Lineárny kernel
3. Polynomiálny kernel s maximálnym stupňom 3
4. Sigmoidný kernel



Obr. 4.2: *OneClassSvm* - Podiel anomálnych záznamov

Z grafu 4.2 vidíme, že Prienik medzi jednotlivými SVM naozaj existuje a je do istej miery podobný. Medzi týmito anomáliami sú aj zaujímavé záznamy, napríklad:
['F', 'P ', 'DXB', 'PRG', 'D5BJ2J', '-1', 'EGY', '2014-08-24 13:25:00', 31]
Tu sa jedná o človeka, u ktorého národnosť bola Egyptská, ale dokument ktorým sa preukazoval nemal uvedený štát vydania (ak sa vôbec preukazoval).

['M', 'P ', 'AUH', 'PRG', 'Z2867', 'AFG', 'AFG', '2012-08-17 14:30:00', 21]
Tu sa jednalo o človeka z Afganistanu cestujúceho z Abu Dhabi.

Medzi ďalšími záznamami boli napríklad nadpriemerne starí ľudia (80 rokov) alebo tiež malé deti.

O *OneClassSvm* vieme povedať, že aj s rôznymi kernelmi označujeme za anomálne množiny, ktoré obsahujú vysoký podiel spoločných prvkov. Tento nástroj detekcie anomálií teda prehlásime za vhodný ďalšieho skúmania a tiež validácie Políciou ČR.

EllipticEnvelope Ďalšou možnosťou unsupervised detekcie anomálií je použitie triedy *EllipticEnvelope*, ktorá počas trénovania „obalí“ vzorky do akejsi obálky, kde sa nachádza väčšina trénovacích vzorkov. Keď následne chceme zis-

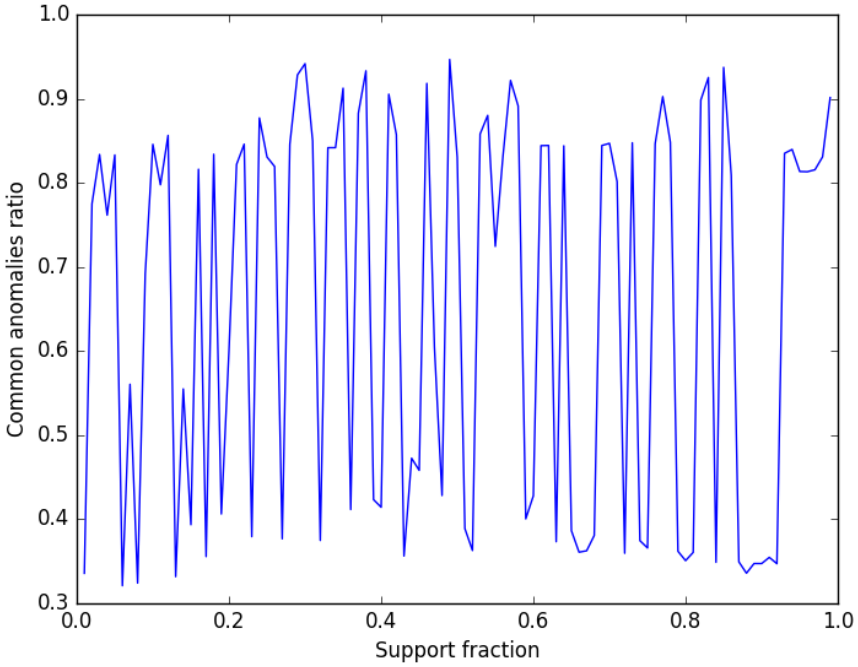
tiť, či je záznam anomálnym, skúmame, či sa nachádza v obálke alebo mimo nej.

Podobne ako pri *OneClassSvm*, aj tu som sa rozhodol použiť na trénovanie množinu vzorkov výrazne menšiu ako je celý dataset. Využívam stratifikovaný výber vzorkov, ktorým na trénovanie vyberiem 5% vzorkov z datasetu. Kontaminácia (predpokladaný podiel anomálnych záznamov) datasetu bola nastavená na 10%.

Tento výber následne používam na skúmanie vplyvu parametru *support_fraction* na to, ktoré inštancie z testovacieho datasetu (doplnok z celého datasetu k trénovacej množine) sú označené za anomálne. Tento experiment som sa rozhodol uskutočniť tak, že ako referenčné označenie anomálnych záznamov zoberiem tie inštancie, ktoré označila *EllipticEnvelope* s prednastavenou hodnotou parametra

$$support\_fraction \text{ a to } (n\_sample + n\_features + 1)/2,$$

čo je v našom prípade hodnota približne polovica trénovacieho datasetu. Následne potom skúmam pomer tých inštancií, ktoré označili za anomálne obe *EllipticEnvelope* ku všetkým, ktoré označila referenčná *EllipticEnvelope*.



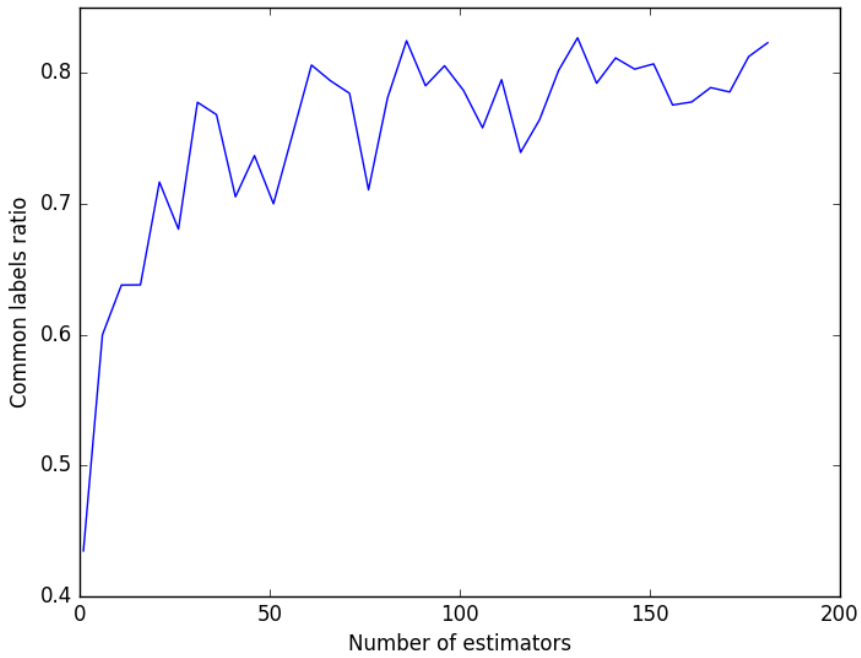
Obr. 4.3: *EllipticEnvelope* - Vplyv support_fraction

Ako vidíme na obrázku 4.3, označenie voči referenčnému riešeniu rapídne kolíše. Taktiež nevidíme žiadnu závislosť medzi *support_fraction* a pomerom rovnako označených inštancií.

Preto na základe predpokladu, že zmysluplná detekcia anomálií buď nie je začne ovplyvnená malou zmenou parametru, alebo vidíme závislosť medzi touto zmenou parametru a výstupom detekcie anomálií, prehlasujem *EllipticEnvelope* za nevhodný nástroj.

IsolationForest Predposledným spôsobom detekcie anomálií ponúkanou knižnicou scikit je *IsolationForest*. Dataset delím rovnako ako pri *EllipticEnvelope* a aj experimenty s *IsolationForest* sú vedené v podobnom duchu. Pri každom z nich skúmam vplyv zmeny hodnoty parametra na podiel anomálií označených referenčným lesom aj skúmaným lesom ku všetkým anomáliám označeným referenčným lesom (vždy prednastavená hodnota parametru). Kontaminácia datasetu bola nastavená na 10%.

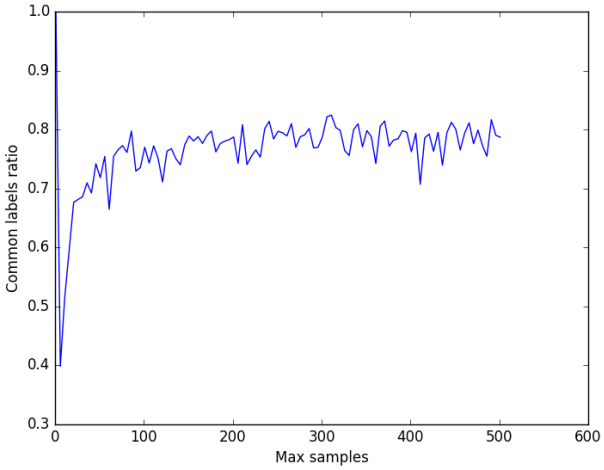
Prvým parametrom bol počet stromov v lese. Prednastavenou hodnotou tohoto parametra bolo 100 stromov. Ako vidíme na grafe 4.4, do istej hranice (približne 60 stromov) sa výstup líši od referenčného výstupu. Rozdiely pomerov s počtom stromov viac ako 60 sa už dajú pokladať za odchýlku počas trénovania (rozdielne množiny použité na trénovanie jednotlivých stromov).



Obr. 4.4: *IsolationForest* - Vplyv number_of_estimators

Ďalším parametrom je max_samples, ktorý hovorí koľko prvkov sa má použiť na trénovanie jednotlivých stromov. Prednastavenou hodnotou je

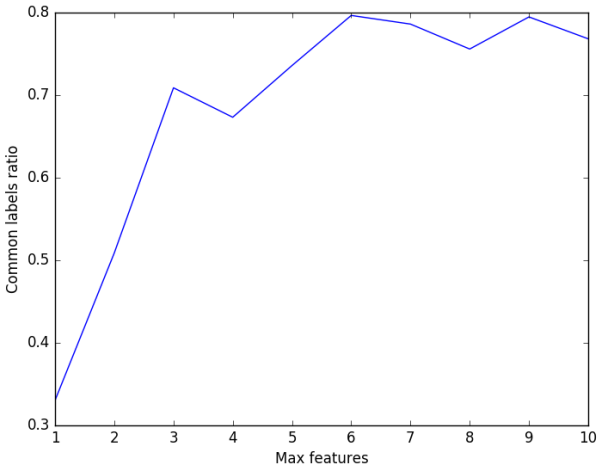
$$max\_samples = min(256, n\_samples), \text{ čiže } 256.$$



Obr. 4.5: *IsolationForest* - Vplyv max_samples

Z grafu 4.5 vidíme, že podobne ako pri počte stromov v lese už pri 100 prvkoch na trénovanie jedného stromu je pomer rovnako označených inštancií vysoký a môžeme teda hovoriť zase o odchýlke počas trénovania.

Tretím parametrom, s ktorým som experimentoval bol maximálny počet atribútov. Prednastavenou hodnotou bolo využitie všetkých atribútov.



Obr. 4.6: *IsolationForest* - Vplyv max_features

Ako sme mohli očakávať, graf 4.6 je rastúci a rozhodne chceme použiť všetky atribúty pre detekciu anomálií, aby nám neušli žiadne.

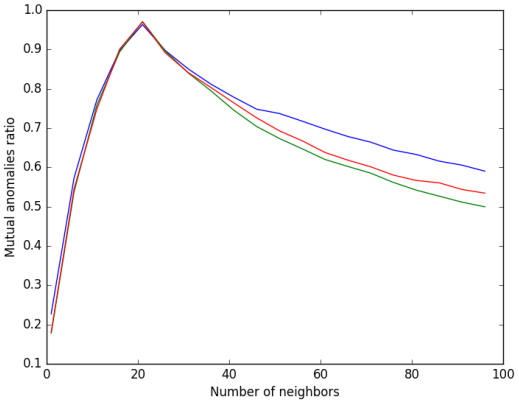
Posledným parametrom bolo využitie bootstrapingu. Podobnosť výsledkov bez použitia a s použitím bootstrapingu je 79.09%. Bootstraping teda detekciu ovplyvňuje, ale len do malej miery.

Z výsledkov *IsolationForest*u vidíme, že aj dva lesy natrénované rovnakou trénovacou množinou s rovnakou konfiguráciou označujú za anomálne rôzne množiny prvkov z testovacieho datasetu, ktoré zdieľajú 80% prvkov. Preto označujem *IsolationForest* za nevhodný pre ďalšie skúmanie.

Local Outlier Factor Poslednou triedou, ktorú ponúka scikit na detekciu anomálií je Local Outlier Factor, ktorý pracuje so vzdialenosťami medzi záznamami a ich hustotou. Parametrami, ktoré môžu byť ladené pri tomto spôsobe je miera vzdialenosti a tiež počet susedov, na základe ktorých budeme vyslovovať závery o anomálnosti záznamu. Tu dáta nedelíme na trénovacie a testovacie keďže sa jedná o detekciu anomálií „in place“. Zložitosť spomínanej detekcie je $O(n^2)$, kde n je počet inštancií a preto som sa rozhodol LOF testovať na menšej množine (stratifikovaný náhodný výber o veľkosti 5% z pôvodného datasetu). Kontaminácia datasetu bola nastavená na 10%.

Ako pri predchádzajúcich experimentoch, aj tu testujem pomer anomálií označených ako testovaným LOF, tak aj referenčným LOF voči počtu anomálií označených referenčným LOF, ktoré je v tomto prípade Local Outlier Factor natrénovaný s parametrom $n\_neighbors = 20$. Testovanými mierami vzdialenosti sú:

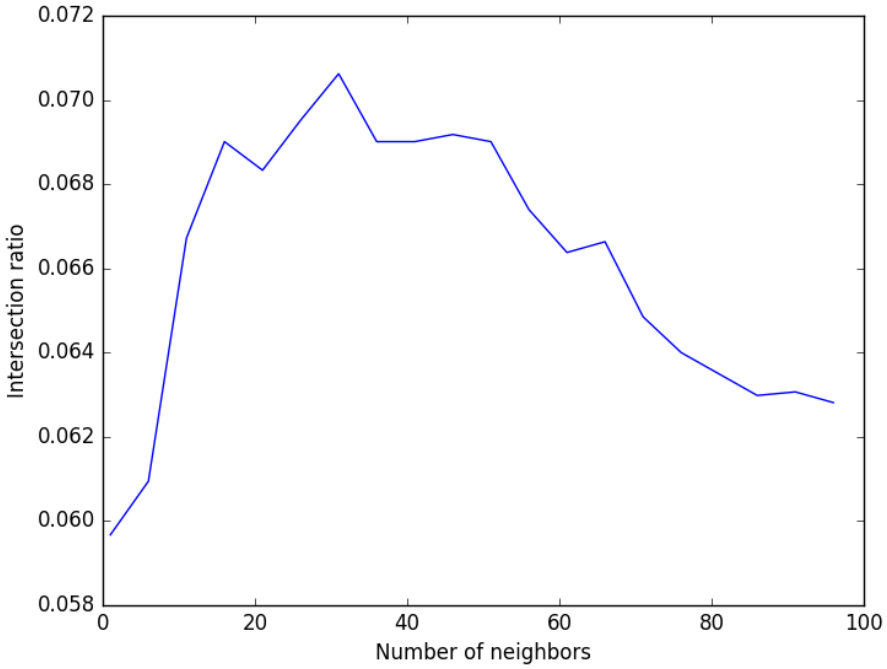
- manhattanská vzdialenosť
- kosínová vzdialenosť
- euklidovská vzdialenosť



Obr. 4.7: LOF - Vplyv number_of_neighbors

Ako vidíme z obrázku 4.7, trend pomeru rovnako klasifikovaných je rovnaký pri každej miere vzdialenosti, líšia sa akurát v prudkosti rastu/klesania.

Pre zaujímavosť som sa rozhodol preskúmať aj prienik týchto LOF s rôznymi mierami podobnosti prvkov a zistiť, akú časť testovacieho datasetu označia za anomálnu všetky tri LOF (prienik).



Obr. 4.8: LOF - Vplyv number_of_neighbors na prienik

Z grafu 4.8 vidíme, že prienik jednotlivých LOF je značný. Každý z nich mal kontamináciu datasetu nastavenú na 10% a ich prienik dosahuje až 7%, čo je oproti *IsolationForest* značne vyšší prekryv. *LOF* teda kvôli malým rozdielom vo výstupe detekcie anomálií s malou zmenou parametru a tiež kvôli viditeľným závislostiam medzi parametrami a výstupom tejto detekcie označujem za vhodný pre ďalšie skúmanie a prezentáciu Polícii ČR.

Kombinácia rôznych metód Už som ukázal ako sa prekrývajú výsledky vrámci jednotlivých tried na detekciu anomálií, ale stále som neukázal ako veľmi sa prekrývajú výsledky medzi triedami. Preto som sa rozhodol uskutočniť experiment, kde na rovnakých dátach natrénujem rôzne unsupervised techniky detekcie anomálií a následne budem skúmať aký veľký je ich prienik na testovacej množine. Na výber inštancií využívam zase stratifikovaný náhodný výber s 5% na trénovanie a 95% na testovanie. Skúmanými triedami sú *EllipticEnvelope*, *OneClassSvm* a *IsolationForest*. Local Outlier Factor nebol

použiteľný, keďže pri ňom sa dáta nedelia na trénovacie a testovacie, ale prebieha na jednej množine a jeho čas je kvadratický s počtom inštancií.

Spomenuté metódy detekcie anomálií majú nasledovné nastavenia:
EllipticEnvelope

- $support\_fraction = (n\_sample + n\_features + 1)/2$

OneClassSVM

- kernel - RBF kernel
- $nu = 0.1$

IsolationForest

- $n\_estimators = 100$
- $max\_samples = 256$
- $max\_features$ - všetky
- bootstrap - false

Všetkým metódam bol nastavený koeficient kontaminácie na 10%

Prienik výsledkov týchto metód označil za anomálne 4.567% testovacích inštancií. Pre ilustráciu, medzi týmito označenými pasažiermi boli:

['F', 'P ', 'YUL', 'PRG', 'CAN', 'CAN', '2014-06-25 11:30:00', 1]
['M', 'P ', 'EVN', 'PRG', 'NLD', 'XXA', '2012-08-31 06:15:00', 76]
['M', '-1', 'EVN', 'PRG', '-1', 'ESP', '2014-08-14 06:20:00', 12]
['F', 'P ', 'DXB', 'PRG', 'VNM', 'VNM', '2014-01-13 12:35:00', 0]

Vidíme, že medzi označených zase patria vekové extrémy, alebo chýbajúci dokument, ktorým sa pasažier preukazoval. Taktiež vidíme, že takmer polovica anomálií označených akýmkoľvek nástrojom pre detekciu anomálií poskytnutého *scikit-learn*om bola označená aj všetkými ostatnými nástrojmi, čo znamená, že tento prienik by mal byť zase prezentovaný Polícii ČR.

4.3.2 Profily pasažierov

Jednou z možných volieb kontextu je vytvoriť každému pasažierovi profil, v ktorom budú zaznamenané všetky jeho lety. Vzniká tak akýsi multigraf, kde sú všetky hrany orientované smerom k letiskám v Českej republike (keďže máme dáta letov začínajúcich mimo Schengenský priestor a končiacich v Českej republike). Týmto spôsobom vieme identifikovať anomálie v letovom profile jednotlivca a skúmať nepravidelnosti v miestach odletu.

Detekcia anomálií založená na profile pasažiera prebieha spôsobom odlišným od klasických detekcií anomálií. Je pri nej vhodné sledovať spolu s miestom odletu aj čas odletu, aby sme boli schopní nové záznamy zohľadňovať viac

ako záznamy, ktoré sú staršie. Keďže je tento spôsob definície kontextu nadmieru špecifický pre daný problém rozhodol som sa inšpirovať spektrálnymi technikami detekcie anomálií podrobnejšie opísané v 3.2.7 a vyvinul vlastný spôsob detekcie anomálií.

Na tom, na aké české letisko let prilieta nám nezáleží, takže profil bude obsahovať vektor, kde každá zložka odpovedá jednému letisku. Keďže chceme viac zohľadňovať nové záznamy v profile, budeme uplatňovať na celý vektor faktor rozpadu (decay rate) $d \in (0, 1)$, ktorým za každý určený časový úsek vynásobíme tento vektor.

Samotná detekcia anomálií bude prebiehať tak, že pri každom novom zázname do profilu porovnáme posledný záznam (bez uplatneného faktoru rozpadu) s novým záznamom, kde najskôr uplatníme faktor rozpadu a tiež pripočítame 1 k zložke vektoru odpovedajúcej regiónu, do ktorého patrí letisko (FlightFrom) tohoto nového záznamu. Toto porovnávanie môže prebiehať rôznymi spôsobmi:

1. Rozdiel vektorov - Pri rozdiele vektorov (ani normalizovanom) však nevieme presne zachytiť anomálie ak sa jedná o letisko (alebo región), z ktorého letí pasažier prvýkrát, ale inak lieta veľa (rozdiel bude malý aj po uplatnení faktor rozpadu).

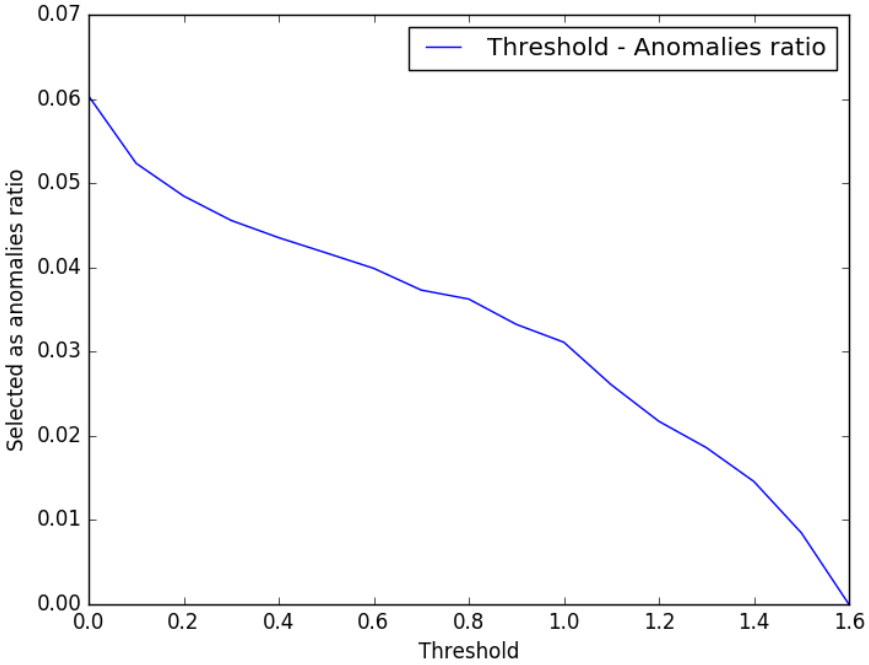
2. Porovnávanie uhlu dvoch vektorov - Takto určená vzdialenosť vektorov odhalí aj anomálie, ktoré rozdiel vektorov nie je schopný zachytiť. Ak sa jedná o letisko, z ktorého pasažier bežne nelieta, bude uhol vektorov veľký. Taktiež, ak pasažier dlhšiu dobu nelieta z letiska, z ktorého lietal bežne a často, faktor rozpadu spôsobí, že vysoké hodnoty klesajú rýchlejšie a tým pádom bude uhol medzi vektormi vyšší.

Porovnávaním uhlov vieme teda detekovať rôzne typy anomálií a teda ho zvolíme pre porovnanie vektorov.

Máme teda dva parametre, ktoré tu môžeme ladiť a to faktor rozpadu a hranica, ktorú keď prekročí anomálne skóre pre nový záznam budeme ho považovať za anomálny. Takto vieme upravovať citlivosť detekcie anomálií.

Avšak pre overenie úplnej správnosti tejto detekcie anomálií a nezostáva nič iné, len dané parametre nastavovať manuálne a skúmať, či výstup je zmysluplný. Pri nastavení faktoru rozpadu na 0.8 (ak človek lieta približne raz za rok z jedného letiska), tak hodnota zostáva približne rovnaká a teda uhol medzi pôvodným vektorom a novým vektorom bude veľmi malý.

Naopak, ak daný pasažier neletel z daného letiska ešte nikdy, získava vektor nový rozmer a tým pádom je uhol medzi novým a pôvodným veľký a teda získavame anomáliu. Taktiež ak pasažier dlho nelieta z letiska, z ktorého zvykol lietavať bežne (vysoká hodnota vo vektore odpovedajúca danému letisku), faktor rozpadu zmenší túto hodnotu viac v porovnaní s ostatnými hodnotami, ktoré odpovedajú menej frekventovaným letiskám.



Obr. 4.9: Profily pasažierov - Vplyv hranice anomálnosti

Na grafe 4.9 vidíme, ako klesá podiel detekovaných anomálií na základe voľby hranice anomálnosti. Samozrejme môžeme označovať za anomálnych pasažierov, ktorí letia prvýkrát (ktorých je drvivá väčšina), ale na tomto grafe som chcel ukázať, ako veľmi je podiel vybraných pasažierov ovplyvnený týmto parametrom.

4.4 Analytické otázky

Ďalším požadovaným výstupom sú odpovede na dohodnuté analytické otázky.

4.4.1 Klasifikácia nebezpečných pasažierov

Prvá otázka a všetky jej podotázky sa zaoberajú jedným - trénovaním modelu, následnou klasifikaciou jednotlivých pasažierov a hodnotením úspešnosti modelu. Na natrénovanie aj testovanie budeme využívať anonymizované dáta (kde PassengerId budeme využívať ako identifikáciu pasažiera) a ako label, na základe ktorého budeme učiť a testovať využijeme atribút *HitType*. Aby sa jednalo o binárnu klasifikáciu, *HitType* transformujeme na binárny label. Túto transformáciu som sa rozhodol vykonať tak, že pasažierov, ktorých *HitType* nie je (bezpeční pasažieri) označím 0 a naopak nebezpečných 1. Nastáva

tu však otázka, čo urobiť s pasažiermi, ktorým *HitType* chýba. Ja som sa rozhodol týchto pasažierov označiť ako nebezpečných a na základe toho boli uskutočnené experimenty.

Atribúty budú transformované tak, ako bolo opísané v sekcii 4.2.5. Takto transformovanými dátami prebieha trénovanie a testovanie rôznych prediktívnych modelov pre ich následné zhodnotenie na základe analytických otázok a tiež porovnanie medzi sebou.

Aby sme však rozumeli, ako zhodnotiť jednotlivé modely, potrebujeme rozumieť, na ktoré z metrík úspešnosti modelu musíme pozerať:

- Je možné na základe poskytnutých dát zachytiť atribúty nebezpečného pasažiera? (označiť všetkých nebezpečných pasažierov)
  Na túto otázku odpovieme tak, že nájdeme prediktívny model, ktorý má pri klasifikácii nebezpečných pasažierov lepšiu hodnotu recall (podiel správne označených nebezpečných pasažierov ku všetkým nebezpečným pasažierom) ako náhodný klasifikátor (t.j. viac ako 50%).

- S akou presnosťou vieme určiť týchto pasažierov?
  Zase sa sústredíme na klasifikáciu nebezpečných pasažierov, ale naopak na hodnotu metriky precision (podiel správne klasifikovaných nebezpečných pasažierov ku všetkým klasifikovaným ako nebezpeční pasažieri).

- Dokážeme vymodelovať „bezpečného pasažiera“? Táto otázka sa sústredí na nájdenie modelu, ktorý má vysokú hodnotu precision pri klasifikovaní bezpečných pasažierov. Pri takomto modeli teda vieme povedať, že ak už nejakého pasažiera klasifikuje ako bezpečného tak bezpečný v skutočnosti (s vysokou pravdepodobnosťou) aj je.

Pre podrobné preskúmanie každého prediktívneho modelu, vykonávam experimenty, pri ktorých skúmam vplyv jednotlivých parametrov na úspešnosť klasifikácie a tým tiež hľadám optimálne nastavenie spomínaných parametrov.

Náhodné lesy Prvým zo skúmaných prediktívnych modelov je náhodný les. Keďže *scikit-learn* už poskytuje jeho implementáciu, rozhodol som sa ju použiť. Keďže trénovanie aj testovanie prebiehalo nadmieru rýchlo oproti ostatným klasifikátorom, uskutočnil som na nich rôzne experimenty.

Ako som už spomenul, v transformácii som dátumy konvertoval na timestamp. Keďže je ale vek pasažiera v dátum letu (rozdiel timestampov týchto dátumov) viac vypovedajúcou informáciou o tomto pasažierovi, rozhodol som sa dátum narodenia nahradiť vekom a porovnať úspešnosť klasifikácie.

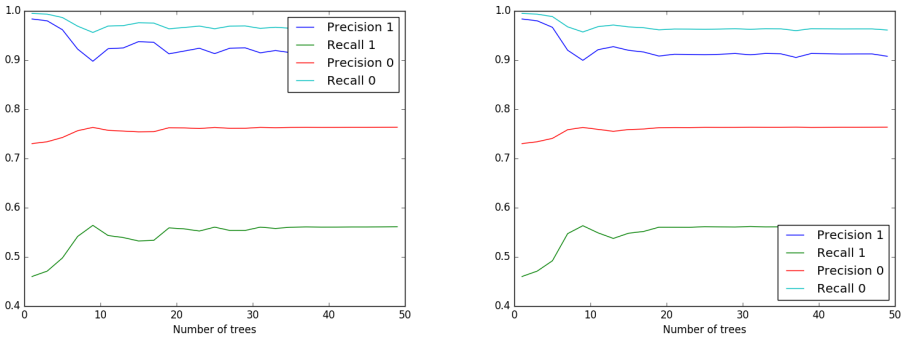
Náhodný les ako model má tiež rôzne parametre, ktoré môžeme ladiť. Skúmanými parametrami sú:

- Počet stromov v náhodnom lese
- Maximálna hĺbka stromu v lese
- Váha jednotlivých tried

Prvým skúmaným parametrom je teda počet stromov. Testovanou konfiguráciou náhodného lesa je:

- Maximálna hĺbka stromu v lese - 5
- Kritérium na štiepenie listu - Giny impurity
- Váha jednotlivých tried - balanced

Trénovanie prebieha s bootstrapingom na celom datasete. Výsledky sú taktiež testované na celom datasete.

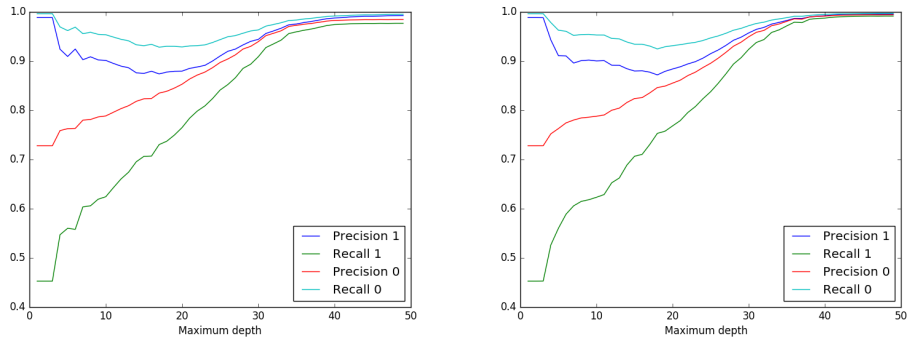


(a) Timestamp narodenia (b) Vek

Obr. 4.10: RandomForest - Počet stromov

Ako vidíme na obrázkoch 4.10, počet stromov výrazne ovplyvňuje klasifikáciu do hodnoty 20 a následne sa ustáli. Povedzme teda, že počet stromov nám bude stačiť 20. Naopak rozdiel medzi použitím veku a dátumom narodenia je minimálny.

Druhým testovaným parametrom je maximálna hĺbka stromu v lese. Počet stromov nastavíme na hodnotu 20 na základe predchádzajúceho experimentu.



(a) Timestamp narodenia (b) Vek

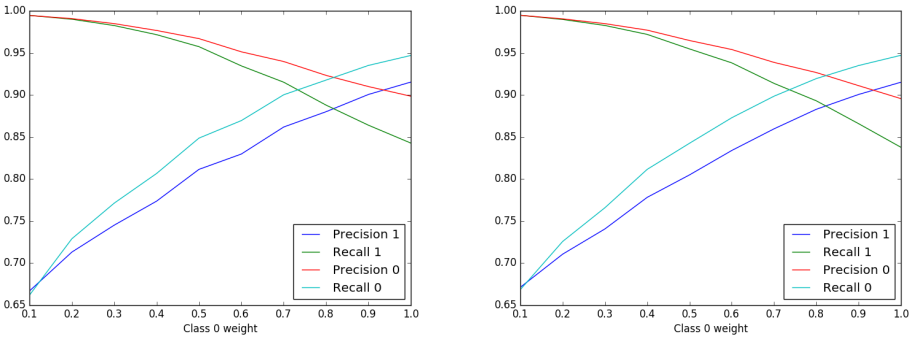
Obr. 4.11: RandomForest - Maximálna hĺbka stromu

Ako vidíme na obrázkoch 4.11, maximálna hĺbka stromov ovplyvňuje úspešnosť klasifikácie v značnej miere. Čo je však zaujímavejšie, dostávame sa k veľmi vysokým hodnotám precision aj recall pri oboch triedach. Aj keď je rozdiel medzi použitím veku namiesto dátumu narodenia malý, je badateľný a pri vyšších hĺbkach poskytuje skvelé výsledky.

Posledným ladeným parametrom bola váha tried. Keďže máme dve triedy a môžeme predpokladať, že nás vždy bude viac (alebo rovnako) zaujímať trieda nebezpečných pasažierov. Váha triedy nebezpečných pasažierov je fixovaná na hodnotu 1.

Testovanou konfiguráciou náhodného lesa je:

- Počet stromov v lese - 20
- Maximálna hĺbka stromu v lese - 25
- Kritérium na štiepenie listu - Giny impurity



(a) Timestamp narodenia (b) Vek

Obr. 4.12: RandomForest - Váha triedy bezpečných pasažierov

Ako vidíme na obrázkoch 4.12, váha tried len posúva úspešnosť klasifikácie jednej triedy na úkor druhej. Na základe toho, na ktorú z otázok chceme odpovedať (ktorú triedu chceme presne klasifikovať), môžeme váhu tried nastaviť tak, aby sme dosiahli čo najlepších výsledkov. Experiment nahradenia timestampu dátumu narodenia vekom však ovplyvnil klasifikáciu minimálne. S ohľadom na tieto experimenty teda odpovedáme na otázky:

1. Áno, je možné zachytiť atribúty nebezpečného pasažiera a označiť tak všetkých nebezpečných. Pri počte stromov 20 a maximálnej hĺbke stromu 45 a vyváženej váhe tried máme recall triedy nebezpečných pasažierov $\sim 99\%$.

2. S odvolaním na model spomenutý v predchádzajúcom bode, vieme týchto pasažierov určiť tiež s presnosťou $\sim 99\%$.

3. Vymodelovať bezpečného pasažiera teda vieme tiež. Pravdepodobnosť, s ktorou vieme povedať, že nami označený bezpečný pasažier je naozaj bezpečný je pri spomínanom modeli $\sim 99\%$.

Experiment nahradenia timestampu dátumu narodenia vekom ovplyvnil klasifikáciu so skúmanými parametrami a to tak, že model sa bol v niektorých prípadoch lepšie naučiť. Preto v ďalších skúmaniach nahrádzam timestamp vekom.

Neurónové siete Druhým zo skúmaných prediktívnych modelov sú neurónové siete. Empiricky som zistil, že trénovanie prebieha veľmi zdĺhavo na vysokom počte trénovacích inštancií, rozhodol som sa túto metriku zahrnúť v experimentoch.

Ako aj náhodné lesy, aj neurónové siete sú zahrnuté ako súčasť balíčka scikit. Keďže trénovanie na celom datasete neprichádza v úvahu (spôsobuje MemoryError), využívam na určenie úspešnosti modelu operátor krížovej validácie. Presnejšie sa jedná o trojnásobnú krížovú validáciu so stratifikovaným k-fold výberom.
Pri neurónových sieťach skúmam nasledujúce parametre:

- Počet skrytých vrstiev
- Počet neurónov v skrytých vrstvách
- Solver pre optimalizáciu váh

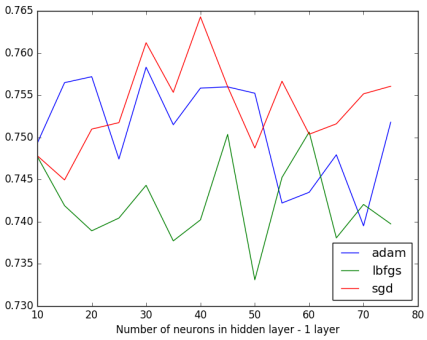
Testovanou konfiguráciou neurónovej siete je:

- Aktivačná funkcia - relu ($f(x) = max(0, x)$)
- Learning rate - constant
- Learning rate init - 0.001

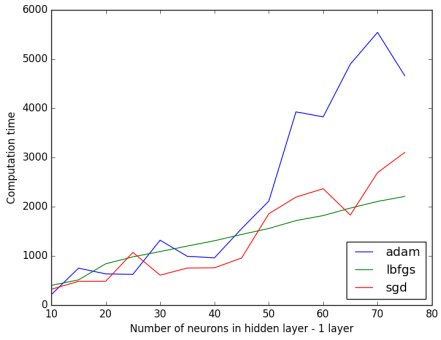
Na základe tejto konfigurácie som uskutočnil merania rozdelené podľa počtu vrstiev a jednotlivé krivky odpovedajú typom solveru pre optimalizáciu váh.

- *adam* - Metóda založená na stochastic gradient descent podľa Kingma, Diederik, a Jimmy Ba [51]
- *lbfgs* - Kvazi-Newtonovská metóda optimalizácie
- *sgd* - Stochastic gradient descent

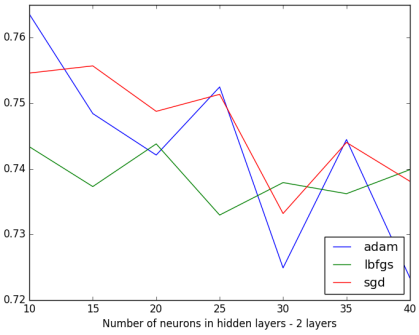
Každé z meraní odpovedá neurónovým sieťam s odpovedajúcim počtom skrytých vrstiev (v každej skrytej vrstve sa nachádza rovnaký počet neurónov, ktorý je parametrom merania - nachádza sa na osi x).



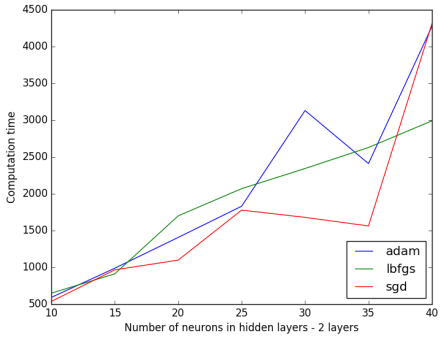
(a) CV score 1 vrstva



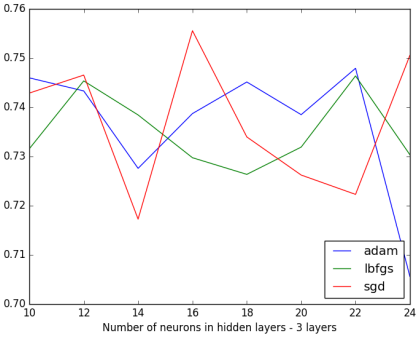
(b) Čas behu 1 vrstva



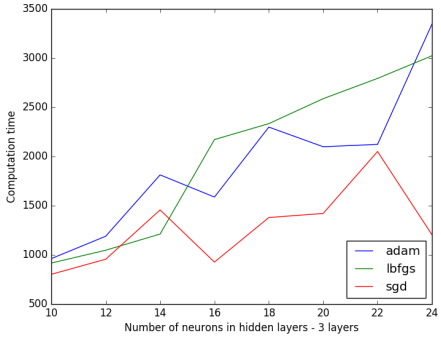
(c) CV score 2 vrstvy



(d) Čas behu 2 vrstvy



(e) CV score 3 vrstvy



(f) Čas behu 3 vrstvy

Obr. 4.13: MLPClassifier - Skryté vrstvy

Ako vidíme z obrázkov 4.13, cross validation score klesá s každou pridanou vrstvou pre všetky typy solverov. Avšak, s rastúcim počtom skrytých vrstiev (aj keď celkový počet neurónov ostáva približne rovnaký) pri solveroch adam a sgd čas trénovania rapídne klesá, pričom čas trénovania lbfgs ostáva približne rovnaký.

Čo sa týka rastúceho počtu neurónov v jednotlivých vrstvách, nevidíme

rastúci trend v cross validation score. Samozrejme všetky krivky času učenia javia rastúci trend s pridanými neurónmi.

Posledným skúmaným je typ solveru. Rozdiely medzi jednotlivými cross validation score týchto solverov sú minimálne. Jediné v čom sa líšia je čas učenia vzhľadom na počet skrytých vrstiev - najprudšie klesanie vidíme u adama, následne u sgd a takmer žiadnu zmenu nevidíme pri lbfgs.

Pre celkové zhodnotenie neurónových sietí nemôžeme priamo použiť analytické otázky a ani porovnanie s náhodnými stromami. Cross validation score za sebou skrýva Accuracy, ktoré je podiel správne klasifikovaných (či bezpečných alebo nebezpečných pasažierov) ku všetkým pasažierom. Keďže priemernou accuracy neurónových sietí je $\sim 75\%$ môžeme povedať, že vieme správne určiť bezpečnosť daného pasažiera s pravdepodobnosťou $\sim 75\%$.

SVM Ďalším skúmaným prediktívnym modelom je Support Vector Machine. Svoje experimenty na SVM som zase vykonával pomocou *scikit-learn*. Ako už samotná dokumentácia scikitu napovedá, použitie SVM by sme mali obmedziť na trénovanie na maximálne desiatkach tisícov záznamov, keďže jeho čas učenia rastie kvadraticky s počtom inštancií. Preto som sa rozhodol pre vykonanie experimentu klasifikácie rozdelením datasetu (počet trénovacích inštancií:počet testovacích inštancií) v pomere 1 : 20. Tento výber prebieha stratifikovaným náhodným výberom. Na trénovanie teda využívame 13000 inštancií.

Skúmaným parametrom vrámci SVM je typ kernelu, ktorý nadobúda hodnôt

1. rbf
2. linear
3. polynomial - maximálny stupeň polynómu je nastavený na tri
4. sigmoid

Kernel	**rbf**	**linear**	**polynomial**	**sigmoid**
precision 1	0.819	0.696	0.795	0.563
recall 1	0.52	0.606	0.509	0.541
precision 0	0.739	0.753	0.731	0.696
recall 0	0.921	0.82	0.911	0.714

Tabuľka 4.1: Výsledky SVM

Podľa tabuľky 4.1, vieme odpovedať na analytické otázky:

1. Áno, je možné zachytiť atribúty nebezpečného pasažiera a označiť tak všetkých nebezpečných. Pri lineárnom kerneli máme recall pri nebezpečných pasažieroch $\sim 60\%$.

2. Čo sa týka precision pri nebezpečných pasažieroch, najlepšiu zase dosahuje rbf kernel, ktorý ju má $\sim 82\%$.

3. Vymodelovať bezpečného pasažiera teda vieme tiež. Pravdepodobnosť, s ktorou vieme povedať, že nami označený bezpečný pasažier je naozaj bezpečný je pri spomínanom modeli (rbf kernel) $\sim 74\%$.

Výsledky SVM môžu byť v dôsledku použitia malej časti datasetu na trénovanie ovplyvnené. Avšak už aj spomínaných 5% datasetu sa trénovalo niekoľko hodín a pridaním ďalších inštancií na trénovanie by sa ešte predĺžilo.

Výsledky Všetky tieto klasifikátory sme teda otestovali s rôznymi parametrami. Ako vidíme, najlepšie výsledky sme dostávali z náhodných lesov, pri počte stromov 20 a maximálnej hĺbke stromu 45. S váhou tried nie je potrebné manipulovať, pretože pri takejto konfigurácii je úspešnosť klasifikácie oboch tried $\sim 99\%$. Ak by sme sa však chceli sústrediť na niektorú triedu viac (aby sme na úkor úspešnosti klasifikácie druhej triedy zvýšili recall prvej triedy) môžeme váhu tejto triedy zvýšiť.

Na základe týchto pozorovaní by som za najlepší prediktívny model zvolil náhodný les so spomínanou konfiguráciou (počet stromov 20 a maximálna hĺbka stromu 45).

Analógia s detekciou anomálií Ako som spomenul v sekcii 4.3.1, ak *HitType* určuje anomálnosť, stáva sa z detekcie anomálií klasifikácia. V tejto kapitole som sa bližšie pozrel na klasifikáciu nebezpečných pasažierov, ktorí by teda mali byť označení ako anomálni (trieda 1). Zároveň som teda splnil úlohu supervised detekcie anomálií a tiež zhodnotil úspešnosť tejto detekcie.

4.4.2 Nebezpečné lety a letiská

Redukovaním záznamov na atribúty

1. *FlightFrom*

2. *HitType*

získavame základ pre identifikáciu nebezpečných letísk. *HitType* v takomto prípade berieme ako označenie a teda ho využívame na skúmanie, aké veľké percento pasažierov z daných letísk má túto hodnotu vyššiu ako 1 (záznamy, kde *HitType* chýbal vynechávame).

Druhou úlohou je analyzovať lety. Podobným spôsobom ako pre letiská zredukujeme atribúty

1. *FlightNumber*

2. *HitType*

a následne sledujeme, ktoré lety majú najvyššiu pravdepodobnosť, že obsahujú nebezpečného pasažiera.

Obe skúmania prebiehajú pythonovským skriptom po vybraní spomenutých atribútov, počítaním záznamov odpovedajúcim jednotlivým letiskám alebo letom, ktoré majú *HitType* rôzny od -1 (ktorým tento atribút nechýbal) a záznamov, ktoré majú *HitType* vyšší ako 1 (nebezpeční pasažieri).

Tieto hodnoty následne dáme do pomeru a sledujeme, ktoré lety alebo letiská majú najvyššiu pravdepodobnosť, že pasažier, ktorý týmito letmi alebo z týchto letísk letí je nebezpečný.

V tabuľke 4.2 stĺpec Airport odpovedá trojpísmenovej skratke jednotlivých letísk. Stĺpec Passengers odpovedá počtu pasažierov, ktorí odlietali z daného letiska do Českej republiky a zároveň mali určený *HitType*. Ratio odpovedá podielu nebezpečných pasažierov spomedzi všetkých pasažierov z daného letiska.

Airport	Passengers	Ratio	Airport	Passengers	Ratio
OVB	18679	0.21	SAW	6125	0.35
VKO	15470	0.24	MSQ	36694	0.36
DME	37334	0.25	UFA	16489	0.36
DWC	12	0.25	ODS	7863	0.38
GYD	13029	0.25	SVX	56625	0.38
SJJ	185	0.26	PEE	14409	0.41
CEK	1649	0.28	KUF	43594	0.44
BEG	10295	0.30	EVN	30747	0.45
DOK	1283	0.31	ROV	32069	0.46
IST	145298	0.31	ALA	15442	0.53
GOJ	17296	0.32	VOZ	161	0.54
KBP	95537	0.32	LED	134020	0.64
TBS	8875	0.32	ADB	117	0.66
SVO	545198	0.34	YUL	2818	0.71

Tabuľka 4.2: Nebezpečné letiská

Ako vidíme v tabuľke 4.2 isté letiská naozaj majú vyššiu pravdepodobnosť, že pasažier, ktorý z nich odlieta má *HitType* vyšší ako 1 a teda by bol zaradený ako nebezpečný pasažier (napríklad YUL - Montréal–Pierre Elliott Trudeau International Airport alebo LED - Pulkovo Airport). Tieto údaje nemusia viesť k presnému označeniu letísk, ktoré sú nebezpečné keďže vynechávame záznamy, ktoré *HitType* určený nemali.

Napriek tomu pri niektorých letiskách vidíme, že sú dostatočne frekventované, čo sa týka pasažierov letiacich do Českej republiky a tiež podiel nebezpečných pasažierov je vysoký.

Na druhú stranu rôznych letov je veľké množstvo. Spomedzi všetkých letov som v tabuľke 4.3 vybral len tie s najvyšším podielom nebezpečných pasažierov:

Flight	Passengers	Ratio	Flight	Passengers	Ratio
OK1893	589	0.31	OK913	21044	0.48
OK2897	175	0.31	OK947	6386	0.51
OK895	46283	0.31	OK181	15442	0.53
OK905	61575	0.31	OK863	2208	0.53
OK917	46641	0.31	7R5509	161	0.54
OK935	8875	0.32	8Q6017	247	0.57
OK893	66179	0.33	OK899	40293	0.59
U6701	20860	0.33	FV6221	19598	0.61
B2861	30107	0.34	OK887	46773	0.61
OK921	1147	0.34	FV221	36896	0.62
PC453	6019	0.34	OK889	7724	0.62
PS807	48763	0.34	QS2653	117	0.66
SU2012	69023	0.34	OK891	10979	0.67
OK255	11056	0.35	TS700	2563	0.69
OK865	4334	0.35	UN9797	53	0.70
OK251	16489	0.36	FV223	1346	0.77
SU2014	47416	0.36	TK3306	106	0.83
JU610	5192	0.37	UN461	10124	0.83
OK907	1441	0.37	FV6715	513	0.87
TK1767	75723	0.37	JA1002	54	0.89
OK923	7863	0.38	CAI755	150	0.91
OK257	13849	0.41	8Q781	94	0.95
SU2010	65089	0.42	6W2951	38	0.97
OK911	40585	0.45	FV5929	67	0.99
OK931	30747	0.45	TS798	70	0.99
OK915	32069	0.46	TS690	159	1.00
UN8361	324	0.46	U6500	3	1.00

Tabuľka 4.3: Nebezpečné lety

Stĺpce sú v tabuľke 4.3 označené podobne ako pri letiskách (tabuľka 4.2), Flight označuje číslo letu, ktorého podiel nebezpečných pasažierov skúmame. Zase musím podotknúť, že vynechávame záznamy pasažierov, ktorí nemali pôvodne atribút *HitType*.

Ako vidíme v tabuľke 4.3, niektoré lety sú príliš malé - nie sú pravidelné, nesú málo pasažierov alebo sme vyfiltrovali veľké množstvo záznamov tým, že sme ignorovali záznamy s chýbajúcim atribútom *HitType*.

Zase sa však vyskytujú isté lety, ktoré sú dostatočne veľké a ich podiel nebezpečných pasažierov je vysoký - napríklad UN461. Čo je však ešte zaují-

mavejšie, let UN461 je pravidelný let medzi letiskami LED - Pulkovo Airport a letiskom v Pardubiciach. Ako sme si mohli všimnúť, letisko LED - Pulkovo Airport sa vyskytlo aj medzi nebezpečnými letiskami.

4.4.3 Neznámi spolucestujúci

Poslednou analytickou otázkou bolo, či existujú pasažieri, čo stále cestujú spolu, ale nikdy nie na jednu rezerváciu.

Jedná sa o jednoduché vyhľadávanie v dátach.

Neznámych spolucestujúcich vieme identifikovať pythonovským skriptom. Najskôr si vytvoríme databázu, kde kľúčom je jednoznačná identifikácia pasažiera a hodnotou je množina jednoznačných identifikátorov letov. Každý z týchto záznamov záznamov porovnám proti ostatným záznamom v databázi a urobíme prienik jednotlivých letov, ktoré absolvovali. Tento prienik však nerobíme ak aspoň jeden z pasažierov absolvoval len jeden let (neodhalíme tým nič zaujímavé) alebo letia na jednu rezerváciu (rodina, kolegovia a podobne - ľudia, čo nemajú problém s tým, že je o nich známe, že cestujú spolu).

Výstupom je zoznam dvojíc pasažierov a letov, na ktorých boli obaja títo pasažieri a zároveň necestovali na jednu rezerváciu. Tento výstup je formátovaný do .json-ovej databázy, ktorej obsah môže byť ďalej spracovaný podľa počtu spoločných letov bez spoločnej rezervácie.

Flights	Pairs
2	101226
3	5300
4	1197
5	410
6	140
7	73
8	31
9	23
10	12
11	12
12	6
13	3
14	3
16	2
18	2
21	1
24	1

Tabuľka 4.4: Neznámi spolucestujúci

V tabuľke 4.4 Flights odpovedá počtu spoločných letov bez spoločnej rezer-

vácie a Pairs odpovedá počtu dvojíc pasažierov, ktoré tento počet spoločných letov má.

Ako vidíme v tabuľke 4.4, pasažierov, ktorí majú spoločné 2 lety a zároveň neletia na jednu rezerváciu je obrovské množstvo (tých, čo majú spoločný len jeden let by bolo ešte viac, ale tých sme nezahrnuli). Zaujímavejším faktom je však to, že sa vyskytujú pasažieri, ktorí spolu lietajú (nie však na jednu rezerváciu) pravidelne. Títo pasažieri by mohli byť označení na podrobnejšie preskúmanie, môže sa však jednať o obchodných cestujúcich so zdieľaným štátom/mestom obchodných ciest. Pre porovnanie dvojice, ktorá dosiahla 24 spoločných letov, jeden pasažier má celkovo 69 a druhý 66 letov a z nich drvivá väčšina začína na rovnakom letisku a teda fakt, že sa vyskytli v 24 spoločných letoch nie je až tak prekvapivý.

Kapitola 5

Budúce práce

Ako som už načrtol v teoretickej časti svojej práce, je mnoho rôznych spôsobov náhľadu na problém detekcie anomálií v dátach, ktoré nám boli poskytnuté. Z mnohých prístupov som sa v tejto práci sústredil na niekoľké a preto v náväznosti na túto prácu by sa mohli spracovať aj ďalšie prístupy.

5.1 Voľba kontextu

Prvou možnosťou pokračovania je zvoliť iný spôsob definície kontextu v dátach. Pri rôzne vymedzených kontextoch sme schopní skúmať iné závislosti a tým pádom detekovať rôzne anomálie. Definícia kontextu nie je jednoznačnou úlohou. Vyžaduje analýzu a preskúmanie rôznych možností, akými môže byť kontext definovaný. V tejto sekcii priblížim spôsoby definície od najviac všeobecného až po špecializované kontexty.

Rozčlenenie na základe národnosti Prvou možnosťou je zoskupenie záznamov s rovn akou národnosťou. Keďže rôznych skupín pri jednom kontexte pre každú krajinu by bolo veľké množstvo a zároveň, črty istých národností sú nadmieru podobné, môžeme zaviesť kontext ako istú skupinu národností. Kontext teda bude obsahovať záznamy, ktoré majú rovnakú národnosť a anomálie budeme skúmať vrámci neho. Takto môžeme nájsť záznamy, ktoré majú vzhľadom na danú národnosť podivuhodné miesto odletu, miesto vydania dokumentu, alebo iné.

Rozčlenenie na základe veku pasažiera Ďalšou možnosťou je rozdeliť dáta na základe veku pasažiera. Vek pasažiera získame jednoduchým odčítaním dátumu narodenia od plánovaného času príletu. Pre jednoduchosť nám bude stačiť vek v rokoch. Vek následne znormalizujeme do intervalu $[0, 1]$ pomocou min-max normalizácie, alebo určíme vlastné skupiny intervalov. Takto definovaný kontext bude pozostávať zo záznamov, ktorých vek v čas príletu

patrí do jedného intervalu. Počet intervalov, na ktoré bude vek pasažiera rozdelený bude parametrom, ktorý budeme sledovať.

Rozčlenenie na základe miesta odletu Jednou z možností je kontext určiť na základe miesta odletu. Jednotlivé miesta odletu môžu tvoriť kontext. Je však zase rozumné zvoliť všeobecnejší kontext. Letiská vrámci jedného štátu, alebo aj letiská vrámci štátov, ktoré majú rovnaké črty (napríklad krajiny a letiská stredného východu) môžu tvoriť ďalšiu možnosť kontextu. Do jedného kontextu teda budú patriť také záznamy, ktoré majú rovnaké miesto odletu, alebo tieto letiská patria do jednej skupiny.

Rozčlenenie na základe čísla letu Na druhú stranu, vieme kontext zvoliť aj viac špecificky. Každé číslo letu by znamenalo nový kontext. Takto by sme vedeli skúmať anomálie v pravidelných charterových letoch na konkrétnej linke.

Rozčlenenie na základe času príletu Taktiež vieme kontext určiť na základe času príletu. Takto zvolený kontext bude zjednocovať lety, ktoré majú plánovaný čas príletu v istom intervale. Veľkosť týchto časových intervalov je zase parametrom.

5.2 Voľba techniky detekcie anomálií

Ďalším spôsobom akým analyzovať a spracovávať dáta by mohlo byť skúmať ďalšie supervised aj unsupervised techniky detekcie anomálií. Predpokladám, že pri oboch spôsoboch by bolo možné vytvoriť ďalší zaujímavý prístup k detekcii anomálií.

5.3 Skúmanie regiónov

Skúmanie politickej a kultúrnej podobnosti jednotlivých národností alebo regiónov, do ktorých patria dané letiská a tým aj definovanie vzdialenosti (dvoch hodnôt atribútu, nie geografickej vzdialenosti) je aj v spolupráci s Políciou ČR obtiažnou úlohou. Napriek tomu je to veľmi zaujímavá úloha, ktorá by mohla viesť k zdokonaleniu detekcie anomálií ako aj konštrukcie prediktívneho modelu pre určovanie nebezpečnosti pasažiera.

Definovanie vzdialenosti medzi dvoma záznamami má kľúčový význam pre unsupervised techniky detekcie anomálií, keďže sa pri týchto metódach skúmajú vzdialenosti medzi susedmi a tiež hustota.

Záver

Vstupom tejto práce boli dáta z policajného systému *OBZOR*. Tieto dáta boli obsiahnuté v .xlsx, .csv súboroch a nachádzali sa v nich rôzne defekty (od nadbytočných a chýbajúcich atribútov až po nezmyselné hodnoty atribútov). Požadovanými výstupmi tejto práce boli predspracovať spomínané dáta do formy vhodnej na detekciu anomálií a následne vykonať detekciu anomálií. Keďže som sa rozhodol využívať knižnicu *scikit-learn* pre vykonávanie tejto úlohy, musel som výstup predspracovania uspôsobiť tak, aby bol vhodný ako vstup do metód, ktoré ponúka táto knižnica. Toto vyžadovalo vysporiadať sa s chýbajúcimi atribútmi a keďže sme chceli, aby detekcia anomálií objavovala anomálie, ktoré reálne anomáliami sú, som nezmyselné hodnoty atribútov nahradil príznačnými hodnotami.

Na takto upravených dátach som demonštroval techniky detekcie anomálií, ktoré ponúka scikit (*OneClassSVM, EllipticEnvelope, IsolationForest, LocalOutlierFactor*) - skúmal vplyv jednotlivých parametrov na tieto metódy a ukázal, ako sa mení výstup detekcie anomálií a na základe toho som zhodnotil, ktoré z týchto metód sú vhodné pre ďalšie použitie a ktoré nie. Pre porovnanie týchto techník som tiež skúmal, aký veľký je prienik ich výstupov. Taktiež som sa inšpiroval spektrálnymi technikami detekcie anomálií a prišiel s vlastnou technikou, ktorá spočíva vo vytvorení profilov pasažierov a sledovaní zmien s každým novým príchodzím letom daného pasažiera.

Ďalším požadovaným výstupom bolo odpovedať na analytické otázky, ktoré boli predmetom dohody s políciou ČR. Tieto otázky boli rozdelené na tri kategórie. Prvou kategóriou boli otázky zamerané na klasifikáciu a hodnotenie modelu:

- Je možné na základe poskytnutých dát zachytiť atribúty nebezpečného pasažiera? (označiť všetkých nebezpečných pasažierov)
- S akou presnosťou vieme určiť týchto pasažierov?
- Dokážeme vymodelovať „bezpečného pasažiera“?

Využitím výstupu predspracovania dát som učil a testoval rôzne modely, ktoré zase ponúkal *scikit-learn*. Medzi testovanými modelmi boli *RandomForest, MLPClassifier, SVM*. Podobne ako pri detekcii anomálií som skúmal úspešnosť týchto modelov v závislosti na ich parametroch a pri každom zhodnotil, ako dobre dokáže odpovedať na jednotlivé otázky. Za najlepší prediktívny model pre tento problém som zvolil *RandomForest* s počtom stromov 20 a maximálnou hĺbkou stromu 45, ktorého úspešnosť v každom ohľade dosahovala 99% (precision aj recall metriky pri oboch triedach). Touto klasifikáciou som tiež preskúmal supervised detekciu anomálií, ak *HitType* určuje anomálnosť pasažiera.

Druhou kategóriou analytických otázok bolo skúmanie nezpečných letísk a letov.

- Dajú sa určiť na základe týchto dát celé lety (alebo letiská), ktoré majú oproti ostatným vyššiu pravdepodobnosť, že v nich budú nebezpeční pasažieri?

Pre odpoveď na tieto otázky som dataset zredukoval na atribúty určujúce bezpečnosť pasažiera (*HitType*) a definujúce letisko (prípadne let) a skúmal pomer bezpečných a nebezpečných pasažierov. Výstupom tejto úlohy boli tabuľky ukazujúce pomer nebezpečných pasažierov v jednotlivých letoch alebo letiskách.

Zaujímavými výstupmi tejto analýzy boli napríklad letisko v Petrohrade, ktorého pomer nebezpečných pasažierov dosahoval 64% alebo charterový let UN461, kde pravdepodobnosť, že sa jedná o nebezpečného pasažiera je približne 83%.

Treťou a poslednou kategóriou analytických otázok, na ktoré som v tejto práci odpovedal je:

- Existujú ľudia, čo stále cestujú spolu v lietadle, ale nikdy nie na jednu rezerváciu?

Na túto otázku som odpovedal vytvorením zoznamov letov jednotlivých pasažierov a skúmal ich prieniky. V dátach sa vyskytovali aj ľudia, ktorí mali spoločných aj viac ako 20 letov, ale zase sa jednalo o pravidelné lety z Petrohradu a každý z týchto podozrivých dvojíc mal z daného letiska nalietaných aspon 60 letov.

Keďže dáta *OBZOR*u ponúkajú bohaté možnosti ďalšej analýzy a aplikovania rôznych techník, či detekcie anomálií, alebo iného skúmania, bol by som rád, keby na túto prácu naväzovali ďalšie výskumy. Preto som v sekcii 5 navrhol ďalšie postupy, ktoré by mohli byť v týchto výskumných prácach obsiahnuté.

Literatúra

[1] Chandola, V.; Banerjee, A.; Kumar, V.: Anomaly Detection: A Survey. *ACM Comput. Surv.*, ročník 41, č. 3, Júl 2009: s. 15:1–15:58, ISSN 0360-0300, doi:10.1145/1541880.1541882. Dostupné z: `http://doi.acm.org/10.1145/1541880.1541882`

[2] Pyle, D.: *Data Preparation for Data Mining*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., první vydání, 1999, ISBN 1558605290, 9781558605299.

[3] Privacy Technology Focus Group Report. `http://www.it.ojp.gov/documents/privacy_technology_focus_group_full_report.pdf`, 2006.

[4] Anonymized Data. `http://medical-dictionary.thefreedictionary.com/Anonymized+Data`, prístupné: 27.8.2017.

[5] De-anonymization. `http://whatis.techtarget.com/definition/de-anonymization-deanonymization`, prístupné: 27.8.2017.

[6] Jenkins, B.: Algorithm alley: Hash functions. *Dr. Dobb's Journal of Software Tools*, ročník 22, 1997. Dostupné z: `http://www.drdobbs.com/database/algorithm-alley/184410284`

[7] Jenkins, B.: lookup3.c. `http://www.burtleburtle.net/bob/c/lookup3.c`, 2006, prístupné: 27.8.2017.

[8] Jenkins, B.: SpookyHash: a 128-bit noncryptographic hash. `http://www.burtleburtle.net/bob/hash/spooky.html`, 2011, prístupné: 27.8.2017.

[9] Lu, C.-T.; Chen, D.; Kou, Y.: Algorithms for Spatial Outlier Detection. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, Washington, DC, USA: IEEE Computer Society, 2003,

ISBN 0-7695-1978-4. Dostupné z: http://dl.acm.org/citation.cfm?id=951949.952103

[10] Abraham, B.; Chuang, A.: Outlier Detection and Time Series Modeling. *Technometrics*, ročník 31, č. 2, Máj 1989, ISSN 0040-1706, doi:10.2307/1268821. Dostupné z: http://dx.doi.org/10.2307/1268821

[11] Basu, S.; Meckesheimer, M.: Automatic Outlier Detection for Time Series: An Application to Sensor Data. *Knowl. Inf. Syst.*, ročník 11, č. 2, Február 2007, ISSN 0219-1377, doi:10.1007/s10115-006-0026-6. Dostupné z: http://dx.doi.org/10.1007/s10115-006-0026-6

[12] Joshi, M. V.; Agarwal, R. C.; Kumar, V.: Mining Needle in a Haystack: Classifying Rare Classes via Two-phase Rule Induction. *SIGMOD Rec.*, ročník 30, č. 2, Máj 2001, ISSN 0163-5808, doi:10.1145/376284.375673. Dostupné z: http://doi.acm.org/10.1145/376284.375673

[13] Abe, N.; Zadrozny, B.; Langford, J.: Outlier Detection by Active Learning. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, New York, NY, USA: ACM, 2006, ISBN 1-59593-339-5, doi:10.1145/1150402.1150459. Dostupné z: http://doi.acm.org/10.1145/1150402.1150459

[14] Fan, W.; Miller, M.; Stolfo, S. J.; aj.: Using Artificial Anomalies to Detect Unknown and Known Network Intrusions. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, Washington, DC, USA: IEEE Computer Society, 2001, ISBN 0-7695-1119-8. Dostupné z: http://dl.acm.org/citation.cfm?id=645496.658057

[15] Vasconcelos, G. C.; Fairhurst, M. C.; Bisset, D. L.: Investigating Feedforward Neural Networks with Respect to the Rejection of Spurious Patterns. *Pattern Recogn. Lett.*, ročník 16, č. 2, Február 1995, ISSN 0167-8655, doi:10.1016/0167-8655(94)00092-H. Dostupné z: http://dx.doi.org/10.1016/0167-8655(94)00092-H

[16] Agarwal, D.: An Empirical Bayes Approach to Detect Anomalies in Dynamic Multidimensional Arrays. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM '05, Washington, DC, USA: IEEE Computer Society, 2005, ISBN 0-7695-2278-5, doi:10.1109/ICDM.2005.22. Dostupné z: http://dx.doi.org/10.1109/ICDM.2005.22

[17] Das, K.; Schneider, J.: Detecting Anomalous Records in Categorical Datasets. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-609-7, doi:10.1145/1281192.1281219. Dostupné z: http://doi.acm.org/10.1145/1281192.1281219

[18] Vapnik, V. N.: *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995, ISBN 0-387-94559-8.

[19] Li, Y.; Pont, M. J.; Jones, N. B.: Improving the Performance of Radial Basis Function Classifiers in Condition Monitoring and Fault Diagnosis Applications Where ‘Unknown’ Faults May Occur. *Pattern Recogn. Lett.*, ročník 23, č. 5, Marec 2002, ISSN 0167-8655, doi:10.1016/S0167-8655(01)00133-7. Dostupné z: `http://dx.doi.org/10.1016/S0167-8655(01)00133-7`

[20] Joachims, T.: Training Linear SVMs in Linear Time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, New York, NY, USA: ACM, 2006, ISBN 1-59593-339-5, doi:10.1145/1150402.1150429. Dostupné z: `http://doi.acm.org/10.1145/1150402.1150429`

[21] Mahoney, M. V.; Chan, P. K.: Learning Rules for Anomaly Detection of Hostile Network Traffic. In *Proceedings of the Third IEEE International Conference on Data Mining*, ICDM '03, Washington, DC, USA: IEEE Computer Society, 2003, ISBN 0-7695-1978-4. Dostupné z: `http://dl.acm.org/citation.cfm?id=951949.952127`

[22] Hautamaki, V.; Karkkainen, I.; Franti, P.: Outlier Detection Using k-Nearest Neighbour Graph. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3 - Volume 03*, ICPR '04, Washington, DC, USA: IEEE Computer Society, 2004, ISBN 0-7695-2128-2, doi:10.1109/ICPR.2004.671. Dostupné z: `http://dx.doi.org/10.1109/ICPR.2004.671`

[23] Ramaswamy, S.; Rastogi, R.; Shim, K.: Efficient Algorithms for Mining Outliers from Large Data Sets. *SIGMOD Rec.*, ročník 29, č. 2, Máj 2000, ISSN 0163-5808, doi:10.1145/335191.335437. Dostupné z: `http://doi.acm.org/10.1145/335191.335437`

[24] Knorr, E. M.; Ng, R. T.: A Unified Approach for Mining Outliers. In *Proceedings of the 1997 Conference of the Centre for Advanced Studies on Collaborative Research*, CASCON '97, IBM Press, 1997. Dostupné z: `http://dl.acm.org/citation.cfm?id=782010.782021`

[25] Knorr, E. M.; Ng, R. T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, ISBN 1-55860-566-5. Dostupné z: `http://dl.acm.org/citation.cfm?id=645924.671334`

[26] Knorr, E. M.; Ng, R. T.: Finding Intensional Knowledge of Distance-Based Outliers. In *Proceedings of the 25th International Conference on*

Very Large Data Bases, VLDB '99, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, ISBN 1-55860-615-7. Dostupné z: http://dl.acm.org/citation.cfm?id=645925.671529

[27] Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; aj.: LOF: Identifying Density-based Local Outliers. *SIGMOD Rec.*, ročník 29, č. 2, Máj 2000, ISSN 0163-5808, doi:10.1145/335191.335388. Dostupné z: http://doi.acm.org/10.1145/335191.335388

[28] Tang, J.; Chen, Z.; Fu, A. W.-C.; aj.: Enhancing Effectiveness of Outlier Detections for Low Density Patterns. In *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD '02, London, UK, UK: Springer-Verlag, 2002, ISBN 3-540-43704-5. Dostupné z: http://dl.acm.org/citation.cfm?id=646420.693665

[29] Papadimitriou, S.; Kitagawa, H.; Gibbons, P. B.; aj.: LOCI: Fast Outlier Detection Using the Local Correlation Integral. In *ICDE*, 2003.

[30] Jain, A. K.; Dubes, R. C.: *Algorithms for Clustering Data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1988, ISBN 0-13-022278-X.

[31] Guha, S.; Rastogi, R.; Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Inf. Syst.*, ročník 25, č. 5, Júl 2000, ISSN 0306-4379, doi:10.1016/S0306-4379(00)00022-3. Dostupné z: http://dx.doi.org/10.1016/S0306-4379(00)00022-3

[32] Kohonen, T. (editor): *Self-organizing Maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1997, ISBN 3-540-62017-6.

[33] He, Z.; Xu, X.; Deng, S.: Discovering Cluster-based Local Outliers. *Pattern Recogn. Lett.*, ročník 24, č. 9-10, Jún 2003, ISSN 0167-8655, doi:10.1016/S0167-8655(03)00003-5. Dostupné z: http://dx.doi.org/10.1016/S0167-8655(03)00003-5

[34] Jaing, M. F.; Tseng, S. S.; Su, C. M.: Two-phase Clustering Process for Outliers Detection. *Pattern Recogn. Lett.*, ročník 22, č. 6-7, Máj 2001, ISSN 0167-8655, doi:10.1016/S0167-8655(00)00131-8. Dostupné z: http://dx.doi.org/10.1016/S0167-8655(00)00131-8

[35] Soule, A.; Salamatian, K.; Taft, N.: Combining Filtering and Statistical Methods for Anomaly Detection. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, IMC '05, Berkeley, CA, USA: USENIX Association, 2005. Dostupné z: http://dl.acm.org/citation.cfm?id=1251086.1251117

[36] Aggarwal, C. C.; Yu, P. S.: Outlier Detection for High Dimensional Data. *SIGMOD Rec.*, ročník 30, č. 2, Máj 2001, ISSN 0163-5808,

doi:10.1145/376284.375668. Dostupné z: http://doi.acm.org/10.1145/376284.375668

[37] Eskin, E.: Anomaly Detection over Noisy Data Using Learned Probability Distributions. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, ISBN 1-55860-707-2. Dostupné z: http://dl.acm.org/citation.cfm?id=645529.658128

[38] Lane, T.; Brodley, C. E.: Temporal Sequence Learning and Data Reduction for Anomaly Detection. *ACM Trans. Inf. Syst. Secur.*, ročník 2, č. 3, August 1999, ISSN 1094-9224, doi:10.1145/322510.322526. Dostupné z: http://doi.acm.org/10.1145/322510.322526

[39] Günter, S.; Schraudolph, N. N.; Vishwanathan, S. V. N.: Fast Iterative Kernel Principal Component Analysis. *J. Mach. Learn. Res.*, ročník 8, December 2007, ISSN 1532-4435. Dostupné z: http://dl.acm.org/citation.cfm?id=1314498.1314562

[40] IDÉ, T.; KASHIMA, H.: Eigenspace-based Anomaly Detection in Computer Systems. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, New York, NY, USA: ACM, 2004, ISBN 1-58113-888-1, s. 440–449, doi:10.1145/1014052.1014102. Dostupné z: http://doi.acm.org/10.1145/1014052.1014102

[41] Keogh, E.; Lin, J.; Lee, S.-H.; aj.: Finding the Most Unusual Time Series Subsequence: Algorithms and Applications. *Knowl. Inf. Syst.*, ročník 11, č. 1, December 2006, ISSN 0219-1377, doi:10.1007/s10115-006-0034-6. Dostupné z: http://dx.doi.org/10.1007/s10115-006-0034-6

[42] Debar, H.; Dacier, M.; Nassehi, M.; aj.: Fixed vs. Variable-Length Patterns for Detecting Suspicious Process Behavior. In *Proceedings of the 5th European Symposium on Research in Computer Security*, ESORICS '98, London, UK, UK: Springer-Verlag, 1998, ISBN 3-540-65004-0. Dostupné z: http://dl.acm.org/citation.cfm?id=646647.699202

[43] Keogh, E.; Lonardi, S.; Chiu, B. Y.-c.: Finding Surprising Patterns in a Time Series Database in Linear Time and Space. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, New York, NY, USA: ACM, 2002, ISBN 1-58113-567-X, doi:10.1145/775047.775128. Dostupné z: http://doi.acm.org/10.1145/775047.775128

[44] Gwadera, R.; Atallah, M. J.; Szpankowski, W.: Reliable Detection of Episodes in Event Sequences. *Knowl. Inf. Syst.*, ročník 7, č. 4, Máj 2005,

ISSN 0219-1377, doi:10.1007/s10115-004-0174-5. Dostupné z: http://dx.doi.org/10.1007/s10115-004-0174-5

[45] Akoglu, L.; Tong, H.; Koutra, D.: Graph-based Anomaly Detection and Description: A Survey. *CoRR*, ročník abs/1404.4679, 2014, 1404.4679. Dostupné z: http://arxiv.org/abs/1404.4679

[46] Akoglu, L.; McGlohon, M.; Faloutsos, C.: *oddball: Spotting Anomalies in Weighted Graphs*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, ISBN 978-3-642-13672-6, s. 410–421, doi:10.1007/978-3-642-13672-6_40. Dostupné z: https://doi.org/10.1007/978-3-642-13672-6_40

[47] Brin, S.; Page, L.: The Anatomy of a Large-scale Hypertextual Web Search Engine. *Comput. Netw. ISDN Syst.*, ročník 30, č. 1-7, Apríl 1998: s. 107–117, ISSN 0169-7552, doi:10.1016/S0169-7552(98)00110-X. Dostupné z: http://dx.doi.org/10.1016/S0169-7552(98)00110-X

[48] Chakrabarti, D.: *AutoPart: Parameter-Free Graph Partitioning and Outlier Detection*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, ISBN 978-3-540-30116-5, s. 112–124, doi:10.1007/978-3-540-30116-5_13. Dostupné z: https://doi.org/10.1007/978-3-540-30116-5_13

[49] Noble, C. C.; Cook, D. J.: Graph-based Anomaly Detection. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, New York, NY, USA: ACM, 2003, ISBN 1-58113-737-0, s. 631–636, doi:10.1145/956750.956831. Dostupné z: http://doi.acm.org/10.1145/956750.956831

[50] Gnumeric Spreadsheet. http://www.gnumeric.org/, 12 2001.

[51] Kingma, D. P.; Ba, J.: Adam: A Method for Stochastic Optimization. *CoRR*, ročník abs/1412.6980, 2014, 1412.6980. Dostupné z: http://arxiv.org/abs/1412.6980

Dodatok **A**

Zoznam použitých skratiek

ŘSCP Ředitelství služby cizinecké policie

AGPL Affero General Public License

API Application programming interface

CSV Comma separated values

JSON JavaScript Object Notation

MCS Maximum Common Subgraph

GED Graph Edit Distance

NN Nearest Neighbour

LOF Local Outlier Factor

COF Connectivity-based Outlier Factor

ODIN Outlier Detection using In-Degree Number

MDEF Multi-granularity Deviation Factor

SOM Self-Organizing Maps

SVM Support Vector Machine

PCA Principal component analysis

RBF Radial Basis Function

SVM Support vector machine

Dodatok **B**

Obsah priloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `impl` ..... zdrojové kódy skriptov na spracovanie dát
  - `tables` ..... výsledkové tabuľky
  - `thesis` ..... zdrojová forma práce vo formáte LaTeX
- `text` ..... text práce
  - `tothmatu.pdf` ..... text práce vo formáte PDF